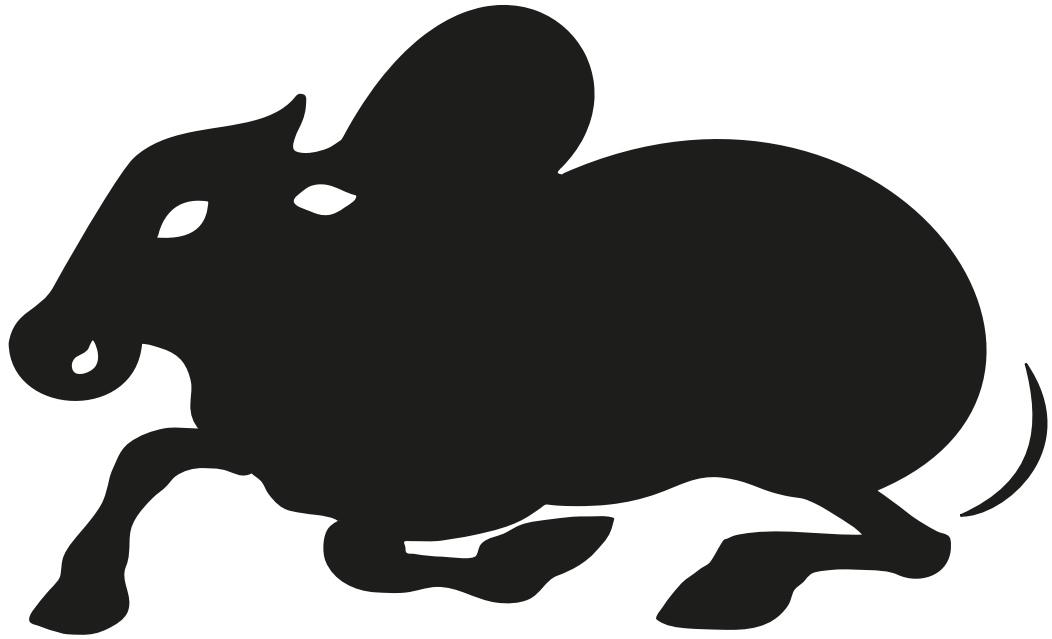
DHARMA Encoding Guide for Diplomatic Editions

Dániel Balogh, Arlo Griffiths



dharma

Encoding Guide

for Diplomatic Editions

Dániel Balogh & Arlo Griffiths

Release Version 1, 2020-07-05

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 809994).

Contents

1. Introduction 6

1.1. Version History ... 6
1.1.1. About this version 6
1.1.2. Fundamental changes since version 0.9 6
1.2. Introductory Remarks ... 7
1.2.1. Acknowledgements 7
1.2.2. Further reading 7
1.2.3. Software 7
1.2.4. Miscellaneous 7
1.3. Terms and Definitions ... 8
1.3.1. Abbreviations 8
1.3.2. Basic terminology 8
1.3.3. XML terms and concepts 9
1.3.4. Conceptual markup 12
1.4. The Structure of an EpiDoc Edition ... 13

2. Marking up Intrinsic Structure in the Edition 16

2.1. Block-level Containers for Intrinsic Structure ... 16
2.1.1. Overview 16
2.1.2. Container boundaries and text segmentation 16
2.2. Prose Containers ... 17
2.2.1. Paragraphs 17
2.2.2. Anonymous blocks 17
2.3. Verse Containers ... 18
2.3.1. Terminology and definitions 18
2.3.2. Overview 18
2.3.3. Numbering stanzas 19
2.3.4. Encoding metre for stanzas 20
2.3.5. Encoding metre for individual lines 20
2.3.6. Verse lines and text segmentation 21
2.3.7. Verse markup versus other markup 22
2.3.8. Marking up structure in lacunose verse 22
2.3.9. Markup examples for verse 22

3. Marking up Extrinsic Structure in the Edition 24

3.1. Overview ... 24
3.2. Physical Lines ... 24
3.2.1. Marking up line beginnings 25
3.2.2. Numbering lines 25
3.2.3. Placement of line beginnings 26
3.2.4. Line beginnings interrupting words 26
3.3. Not-quite Partitions ... 27
3.3.1. Stuff in margins 27
3.3.2. Sectioning with space 27
3.3.3. Spatially offset opening sections (incipits) 28
3.3.4. Spatially offset closing lines (colophons) 28
3.3.5. Pagination or foliation: “forme work” 29
3.4. Boxlike Partitions: Self-contained Zones ... 30
3.4.1. Overview 30
3.4.2. Encoding boxlike partitions 31
3.4.3. Textpart identification: subtype, number and headers 31
3.4.4. Numbered elements in textparts 33
3.4.5. Full markup example for boxlike partitions 34
3.5. Pagelike Partitions: Text Flows through Successive Zones ... 34
3.5.1. Overview 34
3.5.2. Genuine pages 35
3.5.3. Other pagelike zones 36
3.5.4. Zone identification: unit, number and label 36
3.5.5. Placement of page and zone beginnings 37
3.5.6. Numbered elements in pagelike partitions 38

3.5.7. Full markup example for pagelike partitions 39
3.6. Gridlike Partitions: Text Runs Across Contiguous Zones 40
3.6.1. Overview 40
3.6.2. Encoding gridlike partitions 40
3.6.3. Gridlike milestone identification: unit and number 40
3.6.4. Gridlike partitions interrupting words 41
3.6.5. When to encode gridlike partitions 41
3.6.6. Full markup examples for gridlike partitions 42
4. Encoding the Originally Inscribed Text 45
4.1. Alphabetic Characters 45
4.1.1. Tagging transliterated characters as one *akṣara* 45
4.1.2. Tagging parts of alphabetic characters 45
4.1.3. Unusual spatial arrangement in conjuncts 46
4.1.4. Complex characters split by an intervening feature 46
4.2. Non-alphabetic Characters 47
4.2.1. Overview 47
4.2.2. Numeral symbols other than decimal digits 48
4.2.3. Symbol tokens 49
4.2.4. Punctuation marks 50
4.2.5. Space filler signs 51
4.2.6. Miscellaneous symbols 51
4.2.7. Alphanumeric characters used for a different function 51
4.3. Space 52
4.3.1. Generic markup for original space 52
4.3.2. Space for semantic segmentation 52
4.3.3. Space left blank for subsequent filling 53
4.3.4. Space for visual layout 53
4.3.5. Spaces imposed by physical necessity 53
4.3.6. Binding holes in copper plates 54
4.3.7. Surface defects 54
4.3.8. Spaces imposed by other glyphs 55
4.4. Scribal Hands 55
4.5. Premodern Editorial Intervention 55
4.5.1. Premodern deletion 55
4.5.2. Premodern insertion 56
4.5.3. Premodern correction 56
5. Physical Condition and Legibility 58
5.1. Overview 58
5.2. Damage Not Affecting Legibility 59
5.3. Doubtful Readings 60
5.3.1. The EpiDoc element `<unclear>` 60
5.3.2. Tentative readings 60
5.3.3. Ambiguous characters 61
5.3.4. Reading difficulties below the *akṣara* level 61
5.4. Lacunae 63
5.4.1. The EpiDoc element `<gap/>` 63
5.4.2. The reason for a lacuna: illegible or lost 63
5.4.3. Inline lacunae 63
5.4.4. Lacunae with known metre 64
5.4.5. Lacunae below the *akṣara* level 65
5.4.6. Entire lines lost 66
5.4.7. Massive lacunae 67
5.4.8. Lost copper plates 70
5.4.9. Fractured inscriptions 71
5.5. Restoring Lacunae 72
5.5.1. Marking up restored text 72
5.5.2. The basis of restoration 73
6. Editorial Intervention 74
6.1. Correction and Normalisation 74
6.1.1. Correction versus normalisation 74
6.1.2. Markup methods for correction and normalisation 74
6.1.3. Good practice in editorial intervention 75
6.1.4. Correction and normalisation in verse 75
6.2. Encoding Correction 77

6.2.1. Flagging erroneous and uninterpretable text 77
6.2.2. Correcting erroneous text 77
6.2.3. Editorial deletion 77
6.2.4. Editorial addition 78
6.2.5. Distinguishing correction from deletion and addition 78
6.2.6. Good practice in correction 79
6.3. Encoding Normalisation 80
6.3.1. Flagging non-standard usage 80
6.3.2. Normalising non-standard usage 80
6.3.3. Nesting normalisation and correction 80
6.3.4. Good practice in normalisation 81
6.3.5. How non-standard is non-standard? 82
6.3.6. Supplying punctuation 83
6.3.7. Automated normalisation 83
7. Encoding Additional Information in the Edition 85
7.1. Numeral Values 85
7.1.1. Generic numeral markup 85
7.1.2. Difficulties in reading numbers 85
7.1.3. Editorial intervention and numerals 86
7.1.4. Numbers expressed in words 86
7.2. Tagging Language in the Edition 87
7.2.1. Inscriptions consisting of sections in different languages 87
7.2.2. Inscriptions containing foreign words or phrases 87
7.3. Abbreviations 88
7.4. Optional Encoding of Semantic Features 88
7.4.1. Personal names 88
7.4.2. Adding ranks and roles to names 89
7.4.3. Place names 90
7.4.4. Measurements 90
7.4.5. Tagged semantic features interacting with text or markup 90
7.5. Visual Features 91
7.5.1. The scope of visual features encoded in attributes 91
7.5.2. Alignment 92
7.5.3. Directionality and orientation 92
7.5.4. Script 93
7.5.5. Lettering 94
8. General Guidance for Tidy XML Code 95
8.1. Spaces and New Lines in the Code 95
8.1.1. White space 95
8.1.2. Editorial spaces and markup 96
8.1.3. Editorial hyphens and markup 98
8.2. Top to Bottom Hierarchy 98
8.2.1. Block-level elements representing XML structure and extrinsic structure 98
8.2.2. Block-level elements representing intrinsic structure 99
8.2.3. Empty elements representing extrinsic structure 99
8.2.4. Empty elements representing local features 99
8.2.5. Phrase-level elements 99
9. Additional Content Divisions 101
9.1. The Critical Apparatus 101
9.1.1. Overview 101
9.1.2. Indicating location 102
9.1.3. Specifying a precise spot by a lemma 103
9.1.4. Alternative readings, restorations and emendations 104
9.1.5. Identical lemmas, identical readings 104
9.1.6. XML tags in lemmas and readings 105
9.1.7. Freeform apparatus notes 105
9.1.8. Textpart divisions in the apparatus 105
9.2. The Translation 106
9.2.1. Overview 106
9.2.2. Front matter in a translation 107
9.2.3. Attaching multiple translations 107
9.2.4. Reproducing a published translation 108
9.2.5. Structural markup in translation 109
9.2.6. Indicating correspondence to the original 109

9.2.7. Phrase-level markup in translations 110
9.2.8. Foreign words 110
9.2.9. Additions to the translation 110
9.2.10. Indicating uncertainty 111
9.2.11. Indicating incorrect or unexpected text 112
9.2.12. Gaps in the translation 112
9.2.13. Blank space in the translation 113
9.2.14. Indicating bitextuality 113
9.3. The Commentary 113
9.3.1. Overview 113
9.3.2. Structure of the commentary and correspondence to the text 114
9.4. The Bibliography 114
9.4.1. Overview 114
9.4.2. The structured bibliography 115
9.4.3. Bibliographic sigla 115
9.4.4. The epigraphic lemma 115
9.4.5. Full markup example for the bibliography 116
10. Globally Available Markup Outside the Edition 117
10.1. Editorial Markup Outside the Edition 117
10.2. Formatting 117
10.2.1. Character formatting 117
10.2.2. Lists 118
10.3. Encoding Language 118
10.3.1. Tagging language with `@xml:lang` 118
10.3.2. Tagging language in pre-existing containers 118
10.3.3. Tagging foreign languages outside the edition 119
10.4. Notes, Quotations and References 119
10.4.1. Encoding notes 119
10.4.2. Encoding titles 120
10.4.3. Quotations without an encoded reference 120
10.4.4. Quoting published material 121
10.4.5. Bibliographic citations 121
10.4.6. Referring to inscriptions in the DHARMABase 123
10.5. Encoding Names 124
10.5.1. Tagging contemporary names 124
10.6. Attributes as Referencing Systems 124
10.6.1. Encoding authorship with `@resp` 124
10.6.2. Crediting publications with `@source` 125
10.6.3. Identifying persons and places with `@key` 125
10.6.4. Identifying elements with `@xml:id` 125
10.7. Punctuation and Style in Modern Languages 126
11. The TEI Header 127
11.1. Describing the XML Document 127
11.1.1. The title 127
11.1.2. The responsibility statement 127
11.1.3. The publication statement 127
11.2. Describing the Original Document 128
11.2.1. The hand description 128
11.3. Keeping Track of File History 129
Appendices 131
Appendix A. Converting CII/EI Markup Conventions to EpiDoc 131
Appendix B. Metre (Prosody) 132
Looking up Sanskrit metres 132
Syllable length 132
Prosodic code 133
Sanskrit syllabic metres 134
Notes on *anuṣṭubh* 136
Notes on the *upajāti* family 137
Notes on the *vaitālīya* family 137
Vedic trimeter 138
Sanskrit/Prakrit moraic metres 138
Tamil metres 139
Appendix C. “Case Studies” in Encoding Complex Layout 140
Case study 1: four-faced stele 140

Case study 2A: copperplate charter with seal and other goodies 142
Case study 2B: copperplate charter with a lost plate reconstructed 144
Case study 2C: copperplate charter with a lost plate not reconstructed 145
Appendix D. Language Codes 146
Appendix E. Titling Conventions..... 147
Appendix F. Normalisation Suggestions 148
References **150**

1. Introduction

1.1. Version History

Author(s)	Version	Changes	Date
Balogh, Griffiths	0.1	Redaction of the first draft	2019-07
Balogh, Griffiths	0.8	Expansion and revision for release	to 2019-12
Balogh, Griffiths	0.9	Redaction for release	to 2020-03-17
Balogh, Griffiths	1.0	Revision after feedback and discussion	2020-07-05

1.1.1. About this version

– this is the first definitive release version of this Guide
– in case of conflict with the working version in Google Docs,
  – the contents of this document override the Google doc version 1.0
  – but we may in future create a working version with a higher (fractional) version number in the process of working toward the next release, and the contents of such a document shall override the present one
– please at least skim through the guide cover to cover so that you have an idea of the topics addressed

1.1.2. Fundamental changes since version 0.9

– here follows a summary of major changes introduced in 2020
– **line numbering**:
  – unique line numbers are now mandatory (§3.2.2)
  – the line number 0 is no longer permitted; use 01 instead for specially placed initial lines (§3.3.3)
– **boxlike partitions** (textpart divs):
  – the use of this type of encoding is now limited to specific cases and strongly discouraged elsewhere; use a pagelike partition for all other scenarios unless that seems impossible (§3.4.1)
– **space**:
  – the use of `<space>` for blank copperplate pages has been discarded; encode only a `<pb/>` for the blank page (§3.5.2; §4.3.4)
  – the use of `<space>` for blank lines and in general for visual layout has been discarded (§4.3.4)
– **premodern editorial marks**:
  – all such marks are now encoded as `@rend="mark"` regardless of their location and number (§4.5.2)
– **clear characters uncertainly read because of their shape**:
  – instead of `<unclear reason="form">`, use the TEI-sanctioned `<unclear reason="eccentric_ductus">`
– **location references in apparatus**:
  – since unique line numbers are now mandatory, the prefixes “p” and “m” are no longer necessary (and no longer permitted) in `@loc` (§9.1.2)
– **editorial correction and normalisation**:
  – the relevant guidelines (§5.5) have been revised and slightly simplified
– **encoding of symbols**:
  – `@type="symbol"` is no longer used for any symbol (§4.2.6)
  – all punctuation marks (§4.2.4) and space fillers (§4.2.5) are explicitly represented in transliteration in addition to being encoded

1.2. Introductory Remarks

1.2.1. Acknowledgements

– many people in addition to the authors noted above have helped in the creation of this guide; the most significant contributions have been the following
  – the creation of §7.4 by Axelle Janiak and Emmanuel Francis
  – repeated draft review and suggestions by Annette Schmiedchen, Axelle Janiak and Emmanuel Francis

1.2.2. Further reading

– if you are entirely new to XML or the idea of markup, we recommend
  – "The Gentle Introduction to Mark-up for Epigraphers" (Roueché and Flanders, n.d.), available at http://www.stoa.org/epidoc/gl/latest/intro-eps.html
  – "What is XML and why should humanists care? An even gentler introduction to XML" (Birnbaum 2015), http://dh.obdurodon.org/what-is-xml.xhtml
  – for a more in-depth introduction, read the current version of the ur-text "A Gentle Introduction to XML" at https://www.tei-c.org/release/doc/tei-p5-doc/en/html/SG.html
– a good general introduction to EpiDoc can be found in Bodard 2010, available at http://www.stoa.org/wordpress/wp-content/uploads/2010/09/Chapter05_EpiDoc_Bodard.pdf
– for any specific details beyond what we summarise here, consult
  – the EpiDoc guidelines at http://www.stoa.org/epidoc/gl/latest/index.html
  – the TEI guidelines at https://tei-c.org/guidelines/
  – if you find any contradiction between this document and the above guidelines, please inform the authors of the Guide

1.2.3. Software

– the recommended XML editor is Oxygen
  – but you are free to use any editor to produce your marked-up texts
  – text editing software will usually be able to colour-code XML and may also be able to check the well-formedness of the markup or even to validate against a schema
– working in Oxygen, you will need to set a suitable font for the Editor at Options/Preferences/Appearance/Fonts
  – we find that a suitable font
    – can correctly display all the diacritical characters you work with
    – is easy on the eye
    – is preferably one in which the characters | (vertical bar), l (lowercase L) and I (uppercase i) are all easily distinguishable
    – is preferably not too wide, so that you can see plenty of text even when not working on a full screen
  – some fonts we have tested and liked include:
    – Google's free Noto Serif and Noto Sans
    – Microsoft's Cambria and `Consolas`

1.2.4. Miscellaneous

– this guide presupposes that you possess, and are at least superficially familiar with, the DHARMA Transliteration Guide
– the text fragments used for illustration are at present mostly Sanskrit from India
  – contributors working with other languages and regions are welcome to submit samples more relevant to their work, especially if these may require a way of treatment different from the methods described here
  – for the sake of brevity and simplicity, details irrelevant to the topic at hand may be silently normalised, restored, corrected or altered in illustrations drawn from actual inscriptions
– XML `<elements>` mentioned in discussion or used in illustrations are set apart from regular text by typeface, text colour and a shaded background

  - for the sake of brevity and simplicity, details irrelevant to the topic at hand (such as end-tags, attributes and text content) are often omitted in illustrations even though they may be mandatory in actual practice
- XML `@attributes`, when mentioned on their own, are prefixed with an @ sign and highlighted in the same way as elements

1.3. Terms and Definitions

1.3.1. Abbreviations

In addition to some straightforward abbreviations, this Guide uses:

EGD	the DHARMA Encoding Guide for Diplomatic Editions (the present document)
TG	the DHARMA Transliteration Guide[1]
ZG	the DHARMA Zotero Guide[2]

1.3.2. Basic terminology

Some technical terms related to encoding and epigraphy are explained as they are introduced throughout the text of this guide, while a few basic terms are gathered here for clarification.

- **markup** traditionally means annotation within a text to convey information about the presentation of the text, including among others
  - markings in a manuscript to instruct a typesetter, e.g. underline to indicate conversion to italics
  - various brackets and other signs used in philology and epigraphy, e.g. to indicate that certain parts of a text are tentatively read or supplied by the editor
- the TEI guidelines define **encoding** and **markup** as synonymous and applicable in a widely generalised sense to "any means of making explicit an interpretation of a text"[3] and including typographic devices, punctuation marks and even spaces
- in the more circumscribed usage of this guide,
  - **markup** may refer to editorial signs used in a printed edition or to XML encoding
  - **encoding** refers specifically to the method of encoding texts in XML
- a **markup language** is a set of markup conventions used together
- **XML** (eXtensible Markup Language) is a machine-readable markup language used for a wide variety of purposes and independent of hardware or software platform
- **TEI** (the Text Encoding Initiative) is a standard for the machine-readable encoding of texts (understood in a very broad sense) to facilitate text documentation, text representation, text analysis and interpretation
  - TEI has been developed and is maintained by the eponymous Text Encoding Initiative Consortium
  - TEI defines a versatile and massive set of XML methods to mark up texts
- **EpiDoc** is a subset of TEI-compliant markup rules specifically devised for marking up epigraphic documents
- the word **structure** is used in three distinct senses in this guide:
  - **intrinsic structure** refers here to the semantic and metrical structure of a text as abstracted from its physical medium, involving features such as
    - stanzas and other prosodic units
    - semantic units ("paragraphs" and "anonymous blocks") in prose, demarcated by changes in topic
  - **extrinsic structure** refers here to the physical structure of a particular manifestation of a text as a tangible creation, involving features such as

[1] Find the latest release version at https://halshs.archives-ouvertes.fr/halshs-02272407, and the latest internal (draft) version at https://github.com/erc-dharma/project-documentation/tree/master/guides/transliteration. The references to the TG in this document pertain to TG version 3, released simultaneously with EGD version 1.

[2] Find the latest internal version at https://github.com/erc-dharma/project-documentation/tree/master/guides/zotero.

[3] https://tei-c.org/release/doc/tei-p5-doc/en/html/SG.html.

- lines of a particular length that do not coincide with any intrinsic structural unit of the text as a rule, though they may do so
- various inscribed fields such as columns and object surfaces
- sides of copperplate inscriptions (which we call "pages")

– **XML structure** or **markup structure** refers to the way in which markup elements are structured

1.3.3. XML terms and concepts

– the conceptual model of XML is based on structural units technically known as **elements**, which may be
  – **empty**, containing neither text nor further elements; or
  – **non-empty**, containing
    – only text, or
    – only further (empty or non-empty) XML elements, or
    – mixed content, i.e. both text and further elements
– within an XML document, elements take the form of **tags**: words of code distinguished from the textual content by being always wrapped in angle brackets <>
  – most text editing software will use **syntax highlighting** to make tags visually pop out from the content by colouring them differently
– in addition to elements and text, XML documents may contain a few other items which need not concern you generally, except for one item type that you should be aware of: XML allows the use of **character entity references**
  – these are short code words preceded by an & (ampersand) and followed by a ; (semicolon)
  – the purpose of character entity references is to allow the typing, display and processing of characters which are
    – not necessarily supported on certain platforms (such as accented characters, but this case need not bother you)
    – reserved for a special function in XML (and this is what matters to us); thus,
      – should you need to use the < character (which an XML processing engine would interpret as the beginning of an XML tag), you must instead use the entity reference `<` (where "lt" stands for "less than")
      – should you need to use the & character (which an XML processing engine would interpret as the beginning of an entity reference), you must instead use the entity reference `&` (where "amp" stands for "ampersand")
    – so, if during validation in Oxygen you encounter unexpected errors, consider if you may have used the character & or < inadvertently
      – to correct the mistake, type the & character, whereupon Oxygen will automatically suggest a list of pre-defined entity references (starting with &) so all you need do is select and accept the suggestion for the character you need
– in addition to being enclosed in angle brackets, **every XML element must be closed** with the character / (slash)
  – **non-empty elements** must always consist of a pair of tags:
    – a start-tag which names the element, e.g. `<unclear>`
    – and an end-tag which includes the slash and repeats the element name, e.g. `</unclear>`
      – the text and/or other elements between these two tags are the content of such an element
      – as XML hierarchy is always nested, an end-tag always signifies the end of the most recently opened element
    – the tags for **empty elements** normally include this closer sign, e.g. `<lb/>`
      – but they may also be represented as a regular pair of tags with nothing between them: `<lb></lb>`
  – for our purposes, non-empty elements are distinguished into two basic types:
    – **phrase-level** elements, which must be entirely contained within a block-level element and cannot appear except within one
      – these serve to mark up local features of the text, for example uncertain readings, editorial alterations, segments in a different script and numerals

    - **block-level elements** or chunks, which must contain all text within an edition
        - these serve to encode the intrinsic structure of a text (§2)
- text structure is thus conceived of as **hierarchical**, consisting of "boxes within boxes within boxes" or more accurately an *ordered hierarchy of content objects*
    - as an illustration
        - the formatted text A**B***CDE***F**G can be encoded with XML tags marking the string BCDEF as bold and the string CDE within it as italic, since the italic string is nested within the bold one
        - but the text A**BC***DEF*G cannot be encoded with tags marking BCDE as bold and DEF as italic, since neither of these strings are fully nested within the other
            - instead, one would have to encode the formatting in one of the following ways:
                - BC as bold, DE as bold and italic, and F as italic
                - BCDE as bold, DE (within the former) as italic, and F (separately) as a italic
                - BC as bold, DEF as italic, and DE (within the former) as bold
- every XML document must be wrapped in a **root element** which serves as a container for the document as a whole
    - all other elements are **nested** (embedded) either directly within the root element, or at a lower level of embedding
- if an element is embedded directly within another element, then the former is referred to as a **child** of the latter, and the latter as the **parent** of the former
- if an element is embedded at any depth within another element, then the former is a **descendant** of the latter, and the latter is the **ancestor** of the former
    - thus, in Example 1.3.3.A,
        - B and E are children of A (the root element)
        - C and D are children of B
        - F is the child of E
        - B, C, D, E and F are all descendants of A

Example 1.3.3.A: XML hierarchy

```
<A>
  <B>
    <C></C>
    <D></D>
  </B>
  <E>
    <F></F>
  </E>
</A>
```

- while such a conceptual model is eminently suitable for representing the structure of texts in general, it faces a problem when it is desirable to encode further dimensions, i.e. additional (non-coterminous) structures within the same text, such as the extrinsic structure of an epigraphic document *as well as* the intrinsic structure of the text inscribed there
    - such situations are referred to as **overlapping hierarchies**: although either of these dimensions could be represented as an ordered hierarchy, the structures overlap (for instance, a stanza may begin in one inscribed line and end in the next)
        - other overlapping hierarchies relevant to textual studies include
            - syntactical structure
            - semantically distinguished segments (such as names or colophons)
            - the location of spots of damage in a physical support
            - lemmas to which apparatus entries or commentarial notes may need to be anchored
- since XML elements **must never overlap**, the primary structure of an XML document can represent no more than one hierarchy relevant to the encoded text
    - in our EpiDoc editions, the primary hierarchy is that of the text's intrinsic (as opposed to physical) structure
- any alternative hierarchies must be represented in XML using one of two basic methods:
    - by using dedicated empty elements (called "milestones") as pointlike markers of transitions in the alternative hierarchy, instead of non-empty elements as containers for items of the hierarchy
        - thus in our editions, transition points in physical structure are marked with empty elements instead of treating lines and other extrinsic units as elements of the primary hierarchy
    - by deploying linking mechanisms to establish a connection between items located in disparate points of the primary hierarchy

- in our editions this method is most prominently used in the critical apparatus, which is built using **standoff markup**, where the apparatus is located in a section of the XML document separate from the text edition
- alternative hierarchies may also be disregarded in XML editions; thus
  - we do not, at the present stage, use any markup to represent the syntactical structure of a text
  - where physical features of the inscription, such as spots of damage which render the text unclear or illegible, overlap with the intrinsic structure, we use separate XML elements to mark up stretches of the same physical feature divided between two separate elements of the primary hierarchy (see §8.2), and do not use any linking to indicate that the two are in fact a single continuous spot of damage (though such linking would be possible)

- **XML element names** (technically known as *generic identifiers*) are **case-sensitive**: e.g. `<unclear>` cannot be substituted with `<Unclear>` or `<UNCLEAR>`
- XML elements often have **attributes**, whose function is to record additional information about an element
  - attributes have a name (a code word) and a value, which are incorporated into the tag for an empty element, or into the start-tag of a non-empty element, e.g.
    - `<space quantity="3" unit="character"/>`
    - `<unclear cert="low">...</unclear>`
  - one element may have any number of attributes
  - attributes must be separated by spaces from each other and from the element name
  - attributes may appear in any order within an element
    - e.g. `<space unit="character" quantity="3"/>` is entirely identical in meaning to the above example with the same attributes in an inverted order
  - the attribute name is followed by an equal sign and the value in double quote marks[4]
    - note that these must be simple typewriter-style quote marks (i.e. " ")
      - when typing code in a word processor instead of a dedicated XML editor or generic text editor, you must be careful not to allow your "smart" software to change them into prettier printer's quote marks (i.e. “ ”)
  - attributes always qualify only the element to which they belong and have no influence on any other elements such as neighbouring ones or elements of the same type elsewhere in the XML structure
    - however, attributes may be inherited by elements further down in the hierarchy, so that if an attribute is used in an element that contains further elements to which that attribute can apply, then the attribute and value encoded in the ancestor element will also pertain to the descendant elements
  - when attributes are discussed in human-readable text without being cited as full XML tags, they are conventionally not highlighted in any way, but are prefixed with an @ ("at", implying "attribute") sign; thus, in the above examples
    - the element `<space>` has the attributes `@quantity` and `@character`, while the element `<unclear>` has the attribute `@cert`
- in addition to text and elements proper, XML documents may contain some other items, among which you only need to use one:
  - an **XML comment** is anything that is not considered to be part of the document and will be ignored by computers processing an XML file
    - an XML comment must begin with the characters `<!--` and end with the characters `-->`
    - comments may be added by editors as notes to other team members to explain their choice of code or to discuss problems in the edition, e.g. `<!--I'm not sure how to mark this up-->`

[4] The use of single quote marks (' apostrophe) is also permitted by the XML standard. However, for the sake of consistency, we shall always use double quote marks.

  - comments may also be used to "switch off" parts of an XML document without deleting them: any XML code placed within the comment opening and closing sequence will become invisible to computer processing[5]
- an XML document is said to be **well-formed** if it follows the above structural requirements, i.e.
  - the entire document is enclosed in a root element
  - there is no overlap between any elements
  - the start and end of each element is explicitly marked with a tag
- a well-formed XML document may use any arbitrary element names in any particular order and hierarchy: there is no universal and fixed list of possible XML element names and definitions (which is why this is an eXtensible Markup Language)
- the set of rules specifying how certain elements must or must not appear in structural relation to other elements is called an XML **schema** (thus, our editions follow the **EpiDoc schema**)
- an XML document is said to be **valid** if, in addition to being well-formed, it is structured in such a way as to meet the requirements of a particular schema

1.3.4. Conceptual markup

- one of the key points in the "philosophy" of XML is the use of conceptual markup in order to facilitate a separation of the concerns of content and appearance
- **conceptual markup** (also called descriptive markup and semantic markup) essentially means tagging content for *what it is*, as opposed to other types of markup (which are only mentioned here for contrast, but which you need not worry about), namely
  - presentational markup, which tags content for *what it should look like*, as in simple WYSIWYG word processing where you can apply bold, italic, font choice, colour, etc. to bits of text
  - procedural markup, which tags content for *what an algorithm should do with it*, as for instance in TeX (as opposed to LaTeX, which mostly uses conceptual markup)
- as an illustration
  - in a word-processor document you might use only presentational markup, such as
    - 16-point bold for primary headings, 14-point bold for secondary headings, and you might italicise foreign words and book titles,
  - whereas in an XML document you would tag these items as primary/secondary headings, foreign words and titles respectively[6]
    - for presentation, your XML would undergo a transformation (according to separately encoded instructions) and then be displayed as dictated by a stylesheet (also separately encoded) which would determine all details of appearance for each kind of tag in your code
  - in presentational markup, you would be prone to making mistakes, e.g. accidentally using 15-point text for a heading or forgetting to make a primary heading bold, which would at the least make your text look untidy
    - using conceptual markup greatly reduces the chance of such mistakes
  - in presentational markup, you would not have an easy way to manipulate your content selectively, for example to extract a table of contents, and you would have no way to selectively manipulate any kind of content that is not uniquely formatted: in the example above, you would have no means at all to extract a list of titles from your text, since titles are formatted in the same way as foreign words
    - with conceptual markup, all these things are easily done
  - in presentational markup, it would be difficult to change the appearance of items already formatted: to change all primary titles to a different font, you would have to search and replace a precise set formatting instructions with another, and to underline all titles, you would have to check every piece of italic text manually and underline only if it is a title

[5] In Oxygen, press CTRL + SHIFT + , (comma) to turn selected text into a comment or to uncomment text around the cursor.

[6] You are probably aware that advanced WYSIWIG word processing software usually also provides such a facility (the use of styles), which is indeed conceptual markup.

– since formatting is handled by a separate stylesheet in conceptual markup, changing details of global formatting is an easy matter

1.4. The Structure of an EpiDoc Edition

– this section presents an overview of the elements comprising a digital edition in EpiDoc
  – the code shown below is from the DHARMA EpiDoc template version 02 (as of May 2020), an adaptation of the generic EpiDoc template
    – the template file is available at https://github.com/erc-dharma/project-documentation/blob/master/templates/DHARMA_EncodingTemplateInscription_v02.xml
  – thus, you will not need to learn and produce this code, only to find your way around it and add contents
  – the template contains XML comments (§1.3.3) with instructions on filling out contents, but these have been removed from the sample below, and replaced with XML comments clarifying the nature of various parts of the code
  – should you find a discrepancy between the sample below and a later version of the template, the contents of the latest template shall prevail
– the explanatory comments in the sample below refer to sections of this Guide, and are not identical to the comments containing instructions in the actual template

```
<!--XML files begin with a declaration specifying the type of document, normally followed by some processing instructions. You will never need to touch this part of the code. -->
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="http://www.stoa.org/epidoc/schema/latest/tei-epidoc.rng"
schematypens="http://relaxng.org/ns/structure/1.0"?>
<?xml-model href="http://www.stoa.org/epidoc/schema/latest/tei-epidoc.rng"
schematypens="http://purl.oclc.org/dsdl/schematron"?>
<!--The element <TEI> must wrap all TEI-compliant content as a root tag. In our case this includes everything other than the XML declaration and processing instructions. -->
<TEI xmlns="http://www.tei-c.org/ns/1.0">
<!--A header section identifying the digital document and containing additional descriptive information metadata about the encoded text is a mandatory component of every TEI document. The contents of the header are grouped into sections called statements and descriptions, discussed in §11. -->
<teiHeader xml:lang="eng">
<fileDesc>
<titleStmt>
  <title>Encoding template for inscription</title>
  <respStmt>
    <resp>EpiDoc encoding</resp>
    <persName ref="part:jodo">
    <forename>John</forename>
    <surname>Doe</surname>
    </persName>
    </respStmt>
  <respStmt>
    <resp>intellectual authorship of edition</resp>
    <persName ref="part:jodo">
    <forename>John</forename>
    <surname>Doe</surname>
    </persName>
  </respStmt>
</titleStmt>
<publicationStmt>
  <authority>DHARMA
    <note>This project has received funding from the European Research Council ERC under the European Union's Horizon 2020 research and innovation programme grant agreement no 809994.
    </note>
  </authority>
  <pubPlace></pubPlace>
  <idno type="filename">DHARMA_EncodingTemplateInscription</idno>
  <availability>
```

```
    <licence target="https://creativecommons.org/licenses/by/4.0/">
      <p>This work is licensed under the Creative Commons Attribution 4.0 Unported Licence. To
view a copy of the licence, visit https://creativecommons.org/licenses/by/4.0/ or send a letter
to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.</p>
       <p>Copyright c 2019-2025 by John Doe.</p>
    </licence>
  </availability>
  <date from="2019" to="2025">2019-2025</date>
</publicationStmt>
<sourceDesc><!-- At present, only handDesc needs your attention see §11.2; do not touch the rest
of this section. -->
  <msDesc>
    <msIdentifier>
      <repository>DHARMAbase</repository>
      <idno/>
    </msIdentifier>
    <msContents>
      <summary></summary>
    </msContents>
    <physDesc>
      <handDesc>
        <p></p>
      </handDesc>
    </physDesc>
  </msDesc>
</sourceDesc>
</fileDesc>
<revisionDesc>
<change who="part:axja" when="2020-03-18" status="draft">Version 2: addition of handDesc and
summary</change>
<change who="part:axja" when="2019-12-18" status="draft">Creation of the template</change>
</revisionDesc>
</teiHeader>
<!-- All text-related content must be wrapped in the element <text>. In TEI, the text container
may include elements other than <body>, but EpiDoc convention does not use any of these elements,
so <text> and <body> are always opened and closed at the same point. -->
<text xml:space="preserve">
<body>
<!--The body container includes a mandatory division containing the edition.-->
<div type="edition" xml:lang="san-Latn">
<!--Edition encoded as per §2-§7.-->
</div>
<!--The edition may be followed by optional divisions. -->
<div type="apparatus">
<!--Apparatus encoded as per §9.1. -->
<listApp>
<app loc="line">
<lem></lem>
<rdg source="bib:AuthorYear_01"></rdg>
</app>
</listApp>
</div>
<div type="translation" xml:lang="eng">
<!--Translation encoded as per §9.2. -->
</div>
<div type="commentary">
<!--commentary encoded as per §9.3. -->
</div>
<div type="bibliography">
<!--Bibliography encoded as per §9.4. -->
  <p></p>
  <listBibl type="primary">
    <bibl n="siglum"/>
  </listBibl>
  <listBibl type="secondary">
    <bibl n="siglum"/>
```

```
  </listBibl>
</div>
<!--The elements <body>, <text> and <TEI> must be closed in a sequence opposite to the one in
which they were opened. -->
</body>
</text>
</TEI>
```

2. Marking up Intrinsic Structure in the Edition

2.1. Block-level Containers for Intrinsic Structure

2.1.1. Overview

- within `<div type="edition">`, all of the text in an EpiDoc edition must be wrapped in block-level container elements for intrinsic structure, namely
  - `<p>` or `<ab>` for prose (detailed in §2.2)
  - `<lg>` and `<l>` for verse (detailed in §2.3)
- any number of these elements may be used in any sequence as called for by the nature of the text, but these elements may never be nested in one another
  - see §8 for further guidance on how markup elements must be structured
- these block-level elements must contain everything within your edition, except
  - `<div>` elements encoding boxlike partitions (§3.4), which must always be outside block-level containers (§8.2.1)
  - page beginnings and pagelike milestones (§3.5), which are normally inside block level containers but may in some special cases be outside them (§8.2.3)
  - lacunae (§5.4), which are normally inside block level containers but may in some special cases be outside them (§8.2.4)

2.1.2. Container boundaries and text segmentation

- when marking up the end of a block-level container for intrinsic structure and the start of the next block, the respective tags must be placed at word boundaries as accurately as transliteration allows, and may be inserted at any point where an editorial space can be used (as per TG §2.6.1)
  - thus, emphatically, structural units may be split on a semantic or metrical basis at points across which the original text applies sandhi (without vowel fusion) and/or employs a single character of the original script, e.g.
    - `<p> ... ājñāpayaty</p> <p>astu vo viditam ... </p>`
    - `<l>yasya yasya yadā bhūmis</l><l>tasya tasya tadā phalaṁ</l>`
  - optionally, you may flag (or flag and normalise) non-standard usage (§6.3) when sandhi is applied over a major semantic or metrical break
    - to do so, employ the applicable markup on both sides of the break
- if a punctuation mark is present at a container boundary, the punctuation mark is to be included at the end of the earlier containing block
- a container boundary may be necessary at a point where an editorial space is not permitted, in the following rare and specific cases
- if **the container boundary falls inside a compound**,
  - see §2.3.6 and text segmentation for the encoding of verse lines ending inside a compound
  - avoid creating a prose block that ends inside a compound
    - but should you find this absolutely essential, end one block at the desired point and place the editorial hyphen (for compound segmentation) at the beginning of the next block
- if **the container boundary is obscured** by sandhi involving **vowel fusion**, whether inside a compound or between independent words, proceed as follows:
  - should this happen between one verse line (`<lg>`) and the next line of the same stanza, see §2.3.6 for the applicable encoding
  - should this happen across the boundary of `<lg>`, `<p>` or `<ab>` elements, use the following workaround method
    - put the end-tag of the earlier container and the start-tag of the latter container after the fused vowel
    - begin the text of the latter unit with the consonant following the fused vowel
    - add two separate editorial normalisations (§6.3.2) to restore two separate vowels as they would be if sandhi had not been applied:

  - one at the end of the earlier block
  - and one at the beginning of the later block
- you will need to resort to this workaround in the occasional cases where *iti* is fused in sandhi to the end of a stanza ending in *-i*, as shown in Example 2.1.2.A below
- it is strongly recommended that you avoid splitting prose containers at a point where vowel fusion is present, but if you find it essential to do so, you may use this workaround, as shown in Example 2.1.2.B below

Example 2.1.2.A: extraneous text fused to a stanza is split off

```
<lg>
  ...
  <l n="d">pitṛbhiḥ saha majjat<choice><orig>ī</orig><reg>i</reg></choice></l>
</lg>
<ab><choice><orig>ti</orig><reg>Iti</reg></choice></ab>
<p>saṁvatsara-śate ... </p>
```

- the string *majjatīti* is resolved into *majjati Iti* to allow a container break between these words
- since *iti* does not belong semantically to the paragraph after the stanza (being used simply as an end-quote mark), a separate `<ab>` container is created for *iti* between the preceding `<lg>` and the following `<p>`

Example 2.1.2.B: paragraph boundary fused in sandhi

```
<p> ... puṇye tithau muhūrtte c<choice><orig>ā</orig><reg>a</reg></choice></p>
<p><choice><orig>smin</orig><reg>Asmin</reg></choice> divasa-māsa-samvatsare ... </p>
```

- the string *cāsmin* is resolved into *ca Asmin* to allow a paragraph break between these words

2.2. Prose Containers

2.2.1. Paragraphs

- the basic container element for prose text is the paragraph, `<p>`
- short prose inscriptions consisting of at least one complete sentence should be wrapped in a single `<p>` element
- we shall break up longer prose sections into **semantic paragraphs**[7] on the basis of their content:
  - at any point where you feel the topic changes sufficiently to comprise a new semantic unit, end the current paragraph element and start a new one
  - splitting a continuously inscribed text into semantic paragraphs is arbitrary and somewhat subjective; when exercising your own judgement, it may help to imagine translating the text and to put paragraph breaks where you would start a new paragraph in your translation

2.2.2. Anonymous blocks

- wrap prose text in the tag `<ab>` instead of `<p>` when a distinct unit of text does not constitute at least one complete sentence, as in the following cases
  - if the entirety of your inscription (or the entirety of a textpart, for which see §3.4) constitutes less than a complete sentence due to its shortness or lack of syntax, e.g.
    - a sealing with just a name; a label inscription on an image; a graffito
    - a copperplate seal with just a name (in the genitive, nominative, or without a case ending)
    - an auspicious word or symbol in a field set off from the rest of the inscription
  - if a segment of the text is semantically or metrically distinct from adjacent containers (`<p>` or `<lg>`), and constitutes less than a complete sentence due to its shortness or lack of syntax, e.g.
    - an opening invocation consisting only of the word *siddham* or an auspicious symbol

[7] When representing original documents in TEI, the element `<p>` is normally used for paragraphs visually demarcated as such in the original physical manifestation of a text (i.e. units that start in a new line and may begin with an indent). Since our inscriptions seldom employ such visual paragraphs, yet often contain longer sections of coherent prose that lends itself to such segmentation, we refer to such units as semantic paragraphs and encode them with `<p>`. This editorial segmentation is analogous to segmenting *scripto continua* with editorial spaces and will be likewise helpful in display and interpretation.

  - a colophon not comprised of complete sentences
  - a connective particle or phrase (e.g. *iti*, *api ca*) used to introduce or end stanzas and not functioning as an integral part of the unit adjacent to the stanza
- in addition to prose meeting the above conditions, sections of text so heavily damaged that you cannot determine whether they are in prose or verse should also be wrapped in `<ab>`

2.3. Verse Containers

2.3.1. Terminology and definitions

- the terms we use here to discuss metrical structure are as follows:
- **verse**: used as an uncountable noun to refer to text characterised by rhythmically iterated units (generally prosodic units in our case)
  - to avoid ambiguity, this Guide never uses "verse" as a countable noun meaning "stanza" or "line" (as defined below), though both are legitimate meanings of this word
- **stanza**: a unit of verse characterised by a (prosodic) pattern and consisting of a (usually) set number of smaller units (lines) that do not, as a rule, occur in less than a full stanza
  - in Sanskrit syllabo-quantitative verse, *stanza* is equivalent to *catuṣpadī*
  - the term *quatrain* may be used as a synonym for a stanza consisting of four lines
  - in Tamil verse, a *stanza* in the usage of this guide is equivalent to a "poem" (*pā*, *pāṭṭu*, *ceyyuḷ*)
- **line**: a unit of verse characterised by a (prosodic) pattern, with several lines (which may have identical or different prosodic patterns) making up a stanza
  - in Sanskrit syllabo-quantitative verse, *line* is equivalent to *pāda*
  - in Sanskrit/Prakrit quantitative verse, *line* is for our purposes equivalent to *hemistich*, i.e. stanzas of the *āryā* family shall be marked up as consisting of two lines
  - in Tamil verse, *line* is equivalent to *aṭi*
  - **quarter** is used as a synonym of *line* in the context of Sanskrit syllabo-quantitative verse in order to reduce ambiguity by clearly distinguishing verse lines (an element of intrinsic structure) from physical lines (an element of extrinsic structure, §3.2)
- **hemistich**: a half-stanza, the first or the second pair of lines in a quatrain
  - note that the term *hemistich* originates from European classical prosody where it is used in a different sense (a half-line), but the term has been widely applied in European discussions of Indic prosody in the sense in which we use it here
- **break** or line break: a boundary between lines or hemistichs, which usually coincides with a word boundary (not identical to a line break in the context of extrinsic structure, for which see §3.2.1)
  - **enjambement** in our usage means the occurrence of a line break within a word (usually between members of a compound; rarely within a morpheme)
- **caesura**: a boundary within a line, which divides the line into smaller prosodic units (*cola*) and as a rule coincides with a word boundary
- **colon** (plural *cola*): a prosodic unit smaller than a line, a division resulting from the presence of a caesura in a line
- **foot**: a small prosodic unit that has no regard for word boundaries
  - in the present Guide this term is only used as applicable to Sanskrit and Prakrit quantitative verse (*mātrāvṛtta*), where feet (*gaṇa*) consist of a set number of morae
- **mora** (plural *morae*): a unit of prosodic length defined as the length of a short syllable
  - the length of all long syllables is conventionally counted as two morae in Sanskrit and Prakrit quantitative verse
- **metre**: a prosodic template for a stanza (a fixed pattern of syllables or feet), which has a conventional name

2.3.2. Overview

- verse must always be marked up as distinct from prose

  - text in verse shall be marked up only for metrical structure, i.e. semantic paragraphs in a longer verse text must be ignored
- the scheme in this section applies to all verse forms in all languages relevant to our project
  - should you have difficulties applying the scheme to a particular verse form, please contact the authors and the XML-TEI Data Manager with the details to devise a solution
  - this section has been written primarily with Sanskrit syllabo-quantitative verse (*varṇavṛtta*) in mind; subsections below give guidance on some cases that are to be handled differently, but input is eagerly requested on the structure of non-Sanskrit verse and the special considerations it may need
- see §2.3.9 for various examples of verse encoding
- **stanzas as a whole** must be wrapped in the element `<lg>` (for "line group"), with the following mandatory attributes
  - `@n` to assign a number to the stanza (see §2.3.3)
  - `@met` to identify the metre of the stanza by a conventional name (see §2.3.4)
- **each line** must, in addition, be wrapped in the element `<l>` (for "line"), with the following attributes
  - mandatorily, `@n` to assign a number to the line, with values as follows:
    - in Tamil verse, always use Arabic numerals (1, 2, 3, 4)
    - in Sanskrit/Prakrit quantitative verse, use pairs of lowercase Latin letters (ab, cd)
    - in all other cases, use lowercase Latin letters (a, b, c, d)
      - unless you are dealing with stanzas that have (or may have) 10 or more lines, in which case use Arabic numerals instead
  - for stanzas anomalously consisting of more or fewer than the expected number of lines, simply encode the actual number of lines, numbering them in sequence (continuing the applicable numbering scheme as described above)
  - as required, `@enjamb` with the value `"yes"` to encode the fact that the break at the **end of the line does not coincide with the end of a word**, i.e. enjambement is present in this line
    - note that this attribute must be added to the `<l>` element containing the initial part of the broken word, not to the one containing the final part
    - see also §2.3.6
  - optionally, `@real` for lines that deviate from the metre of the stanza as a whole (see §2.3.5)
- **caesuras** shall not be marked up in quantitative verse
  - however, if you notice a caesura that was disregarded by the composer or involves sandhi that blurs its location, you may optionally mark it up as follows: `<milestone type="yati" break="no">`[8]
  - see Example 2.3.9.C for an illustration
- any **original punctuation and internal verse numbering** should be included at its actual locus within the `<l>` element for the line in which it appears
  - all rules for marking up punctuation characters (§4.2.4) and numerals (§) apply
  - **do not supply** editorial **numbers or** editorial **punctuation** at the ends of stanzas
    - editorial numeration is handled through the `@n` attribute of the `<lg>` element, which must be used even when internal verse numbering is present

2.3.3. Numbering stanzas

- **every stanza** in your edition **must have a number** encoded in the `@n` attribute of the corresponding `<lg>` element
- this editorial numbering is mandatory whether or not text-internal numeration is present
  - if the text includes original numeration, editorial stanza numbering shall follow the rules stated here even if this results in a discrepancy with the original numbering

[8] The TEI element `<caesura>` does not take the attribute `@break` (nor `@type`). Since our use of conventional metre names leaves open the possibility to automatically determine the location of caesuras at the prescribed points, the only reason why we might want to encode them is to facilitate research on non-standard caesuras and *yatibhaṅga*, hence our suggestion to use `<milestone>`.

  - original stanza numbers should be encoded as part of the text, and tagged as any other number (see §7.1 about encoding the value of numerals, and §4.2.2 about numeral symbols other than decimal digits)
- stanza numbers shall always be Arabic numerals starting from 1
  - when it comes to displaying editions, we may choose to auto-convert stanza numbers into Roman numerals throughout our corpus or for specific subcorpora
- by default, stanzas shall be numbered **consecutively** throughout an inscription, with the following exceptions
  - if an inscription includes boxlike partitions (§3.4), then stanza numbering must be mandatorily restarted in each textpart division
  - if an inscription includes pagelike partitions (§3.5), then stanza numbering may be optionally restarted after each division in order to follow the numbering scheme of a previous edition or the conventions of your specific field

2.3.4. Encoding metre for stanzas

- we encode the metrical templates of stanzas by using the traditional/conventional names of metres (e.g. *upajāti*, *śārdūlavikrīḍita*, etc.) as values for the `@met` attribute of `<lg>` elements
  - consult the list of metres (Table 3 of Appendix B) for help with metre identification
- if you come across a stanza in a metre to which you can put a name, but that **name is not on the list**
  - use the name as a value
  - contact the authors and the XML-TEI Data Manager to have the name and template or definition added to the list
- if you find a stanza that follows a set metrical template, but **you cannot put a name to the metre**
  - establish the prosodic template and record it in prosodic code as the value of `@met` (see Table 2 of Appendix B for the prosodic code used in attribute values)
- if a text is damaged and you cannot identify the metre with absolute certainty, but you have a **reasonably sound guess** (based on the prosody of the extant part, the length of the lacunose part and/or the metre of surrounding stanzas)
  - add the element `<certainty match="../@met" locus="value"/>` directly after the opening `<lg>` tag (before the first `<l>` element), where
    - `@match="../@met"` indicates that we are encoding uncertainty regarding the `@met` attribute of the parent element (i.e. `<lg>`), and
    - `@locus="value"` indicates that the uncertainty concerns the value of this attribute (i.e. the identification of the metre)
- if a part of your text seems (with reasonable certainty) to be in verse, but it is **too heavily damaged to identify the metre even tentatively**, tag it as `@met="uncertain"`
  - but for heavily lacunose verse where the stanza structure cannot be established with any certainty, consider encoding the text in an `<ab>` element (§2.2.2) instead of marking it up as verse

2.3.5. Encoding metre for individual lines

- the metre of individual verse lines shall not be encoded separately so long as they conform to the metre encoded for the stanza to which they belong
- however, the actual prosodic instantiation of lines may optionally be encoded when a line deviates from the standard pattern for the stanza,
  - **including cases of**
    - legitimate metrical variation or constraint of the conventional template, such as *vipulā anuṣṭubh* or *capalā āryā*
    - licence, such as employing a short vowel followed by a stop and a semivowel (*muta cum liquida* in classical European prosody)
    - lines with anomalous metre, including hypermetrical and hypometrical lines
  - but **excluding cases of**

  - presumable scribal error (e.g. omission, dittography) or non-standard usage which you as editor have corrected (§6.2) or normalised (§6.3), thereby restoring the expected metre (see also §6.1.4)
  - the rare clerical/scribal quirk where the end of a verse is joined in sandhi to a closing *iti*
  - unobserved caesuras (which cannot be marked up in this scheme, but may be encoded as mentioned under §2.3.2)
- to encode the actual prosody of a line, optionally add the attribute `@real` to the `<l>` element concerned
  - the value of this attribute shall be the actual prosodic pattern of the line recorded in the notation described in Table 2 of Appendix B
  - see Example 2.3.9.D, Example 2.3.9.E and Example 2.3.9.F for a illustrations
- as a further option, legitimate variant *anuṣṭubh* lines (*vipulās*) may be explicitly encoded by adding `@met` to the `<l>` element concerned
  - the value of this attribute shall be the name of the prosodic template for that particular line, as listed in Table 3 of Appendix B
  - this, in effect, overwrites locally the value of `@met` encoded for the stanza as a whole
  - if you choose to avail of this option, please use it in addition to, not instead of, the option of encoding `@real` on the irregular line

2.3.6. Verse lines and text segmentation

- as already stated in §2.1.2, the tags for structural elements must be at word boundaries even where a word boundary is separable only in transliteration and not in a native script
  - thus, unlike editions in an Indic script, verse lines must be separated at the exact boundary
    - e.g. `kṣaṇād</l><l>unmūlya` and `śambhor</l><l>gguhām`
    - NOT `kṣaṇā</l><l>dunmūlya` and `śambho</l><l>rgguhām`
  - note that so long as vowel fusion is not involved, non-standard sandhi is irrelevant to the placement of line markup, including
    - the presence of hiatus where sandhi is expected, e.g. at the end of an odd line (`@n="a"` or `@n="c"`)
    - the presence of sandhi where hiatus is expected, e.g. at the end of an even line (`@n="b"` or `@n="d"`), including sandhi between the end of a stanza and the next stanza or the following prose (for which see also the end of this subsection)
    - however, feel free to flag such occurrences as non-standard (§6.3.1)
- where the break is **between** words that are **members of a compound** and vowel fusion sandhi is not involved:
  - place the break as above
  - add the attribute `@enjamb` with the value `"yes"` to the first `<l>` element involved
  - if you are using editorial hyphens for compound analysis (see TG §2.6.2), put your editorial hyphen at the beginning of the second `<l>` element involved
    - e.g. `<l enjamb="yes">... maṇḍal-ānta</l><l>-vyakta-bhrū-bhaṅga...`
  - see Example 2.3.9.A and Example 2.3.9.C for full illustrations
- when **two lines of a stanza are joined in vowel fusion sandhi**, i.e. when the final vowel of a line merges into the initial vowel of the next line,
  - put the line break after the fused vowel (do not type a hyphen anywhere)
  - add the attribute `@enjamb` with the value `"yes"` to the first `<l>` element involved
  - e.g. `<l enjamb="yes">... tathā</l><l>yam`
  - the editorial normalisation described in §2.1.2 should not be used in this case
- when the **end of a stanza is joined in vowel fusion sandhi to text outside** that stanza, use the workaround described in §2.1.2
  - do not use `@enjamb` in this case
  - if the particle *iti* is fused in this way to the end of a stanza, and it is not semantically a part of the following paragraph of text (i.e. not translatable as e.g. “therefore” or “having said so,” but appears simply in the function of a closing quotation mark), then preferably create a separate `<ab>` container for this word between the preceding `<lg>` and the following `<p>`

2.3.7. Verse markup versus other markup

– all **markup** applicable to text **can and must be used** within verse elements
  – note in particular that the beginnings of physical lines and, if applicable, the beginnings of pagelike partitions (§3.5) must always be encoded, even if these coincide with the beginnings of verse lines
    – if the beginning of a physical line (or page) coincides with the beginning of a verse line, place the `<lb/>` (or other applicable element) within the corresponding `<l>` element and before the text of that verse line, e.g. `<lg n="2" met="anuṣṭubh"><l n="a"><pb n="5r"/><lb n="42"/>ṣaṣṭi-varṣa-sahasrāṇi...`
– remember that overlapping hierarchies (§1.3.3) must always be avoided
  – when phrase-level markup (such as that for reading difficulties, §5.3; editorial intervention, §5.5; or additional features, §7) is applicable to a chunk of text that stretches across the boundary of an element for verse structure, create the applicable phrase-level markup in two parts, on both sides of the structural break (see also §8.2 for details)
  – should you find it indispensable to split a stanza or a verse line across more than one structural container, contact the authors of this guide and the XML-TEI Data Manager to discuss how to encode this

2.3.8. Marking up structure in lacunose verse

– see §5.4 about marking up lost and illegible text in general, §5.4.4 about marking up lost text with a known metre, and §5.4.7 about dealing with massive lacunae
– the **structural framework** of stanzas (i.e. the `<lg>` and `<l>` tags) **must** as a rule **be fully encoded** even if a significant part of a stanza is lost; for instance
  – if only the first three lines of a quatrain are extant, do not omit the last line, but add its markup structure and mark up the entire fourth line as a lacuna
– structural markup cannot be placed inside the markup for that lacuna, so if a lacuna extends from one line of a stanza to the next (or from one stanza to the next, or from a stanza into the preceding or following prose), create lacuna markup separately on both sides of the structural break

2.3.9. Markup examples for verse

Example 2.3.9.A: *varṇavṛtta* verse structure with enjambement

```
<lg n="1" met="pṛthvī">
  <l n="a">pradāna-bhuja-vikkrama-praśama-śāstra-vākyodayair</l>
  <l n="b">uparyyupari-sañcayocchritam aneka-mārggaṁ yaśaḥ</l>
  <l n="c" enjamb="yes">punāti bhuvana-trayaṁ paśupater jjaṭāntar-guhā</l>
  <l n="d">-nirodha-parimokṣa-śīghram iva pāṇḍu gāṅgaṁ payaḥ</l>
</lg>
```

Example 2.3.9.B: *āryā* verse structure

```
<lg n="42" met="āryā">
  <l n="ab">śaśineva nabho vimalaṁ kaustubha-maṇineva śārṅgiṇo vakṣaḥ|</l>
  <l n="cd">bhavana-vareṇa tathedaṁ puram akhilam alaṁkṛtam udāraṁ||</l>
</lg>
```

Example 2.3.9.C: unobserved caesura (*vipulā āryā*), with enjambement

```
<lg n="33" met="āryā">
  <l n="ab" enjamb="yes">smara-vaśaga-taruṇa-jana-va<milestone type="yati"
break="no">llabhāṅganā-vipula-kānta-pīnoru-|</l>
  <l n="cd">stana-jaghana-ghanāliṅgana-nirbhartsita-tuhina-hima-pāte||</l>
</lg>
```

- the unobserved caesura is optionally encoded as per §2.3.2
- `@enjamb` is added mandatorily to the first *pāda* as per §2.3.6

Example 2.3.9.D: *muta cum liquida* licence

```
<lg n="12" met="anuṣṭubh">
  ...
  <l n="c" real="+-+++++-">yasya vittaṁ ca prāṇāś ca</l>
  <l n="d">deva-brāhmaṇasād gatāḥ</l>
</lg>
```

- the word *ca* in *pāda c* is expected to be a short syllable, but by regular prosodic rules it is positionally long
- the encoding of `@real` is optional (§2.3.5)

Example 2.3.9.E: *vipulā anuṣṭubh*

```
<lg n="9" met="anuṣṭubh">
  <l n="a" met="na-vipulā" real="+-++---+">śaurya-satya-vrata-dharo</l>
  <l n="b">yaḥ prayāga-gato dhanī</l>
  ...
</lg>
```

- the encoding of `@real` and `@met` on an individual line are both optional (§2.3.5)

Example 2.3.9.F: anomalous metre

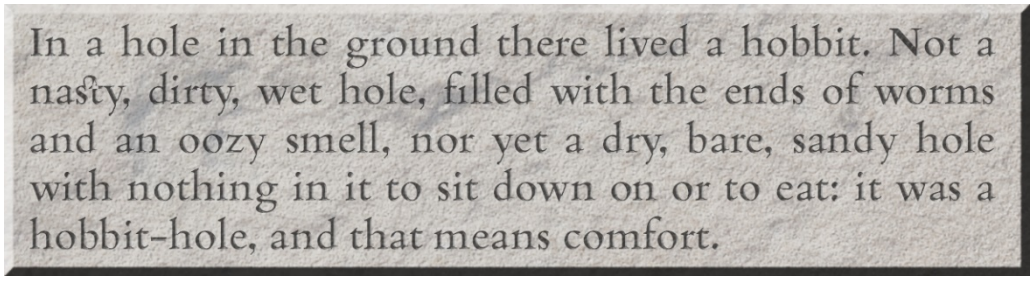
```
<lg n="1" met="āryā">
  <l n="ab" real="----+-+-+--+--------+++">jayati vibhuś catur-bhujaś catur-arṇṇava-vipula-
salila-paryyaṅkaḥ</l>
  ...
</lg>
```

- there is syncopation from the second to the third foot: the foot boundary should be halfway through the long syllable *tu*
- mentally, the author's pronunciation may have been *caturabhujaś*, which would scan correctly: should be *jayati vi|bhuś catu|ra-bhujaś…*
- the encoding of `@real` is optional (§2.3.5)

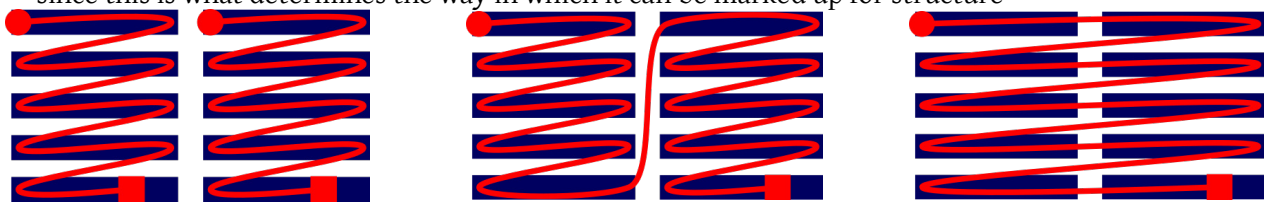
3. Marking up Extrinsic Structure in the Edition

3.1. Overview

– the basic unit of an inscription's physical structure is the line, and line beginnings must be represented in all encoded inscriptions as set out in §3.2 below
– many inscriptions are engraved within **a single**, more or less rectangular **field** as in the illustration below
  – the only aspect of extrinsic structure that needs to be encoded in such cases is the sequence of line beginnings (§3.2)



– **slightly more complex** cases may involve
  – text presented in a single field but sectioned by empty lines (§3.3.2)
  – an auspicious opening symbol, word, phrase or invocation added outside or floated within the principal field (§3.3.3)
  – a closing formula inscribed in a margin (§3.3.4)
  – the presence of folio numeration (§3.3.5)
– when the principal field itself is **partitioned into several zones**, the methods for encoding partitions differ according to how this extrinsic structure relates to the text's intrinsic structure
  – the crux is the **abstract functional pattern** by which the text manifests in multiple inscribed zones, since this is what determines the way in which it can be marked up for structure



A: text stops boxlike partition, §3.4 | B: texts flows on pagelike partition, §3.5 | C: text runs across gridlike partition, §3.6

– the question to ask yourself while deciding how to mark up partitions is this: what does the text do as it reaches the boundary of one zone?
  – if it stops (scheme A), you have a boxlike partition: §3.4
  – if it flows on to the next zone (scheme B), you have a pagelike partition: §3.5
  – if it runs across into the next zone (scheme C), you have a gridlike partition: §3.6
– although the above abstract schemes show text zones side by side, the actual spatial pattern in which they are laid out relative to one another (e.g. one below the other, on separate faces of a three-dimensional object, etc.) is a metadata issue and has no bearing on the markup required
  – the same markup method may be applicable to physically very different objects
– it is recommended that you read through at least the overview and the markup examples for each partition type to familiarise yourself with the possibilities

3.2. Physical Lines

– to make the distinction from verse lines (§2.3.1) explicit, inscribed lines are referred to in this guide as **epigraphic lines** or **physical lines**

– for the purpose of encoding in our project, we define a physical line as a stretch of text whose characters comprise a physically and textually contiguous sequence while being physically distinct from characters belonging to other lines
– this definition includes no presumptions concerning a line's
  – position (some lines of a text may be set off from other lines; see §3.3 for specific cases)
  – directionality (e.g. some lines run right to left while others run left to right; see §7.5.3)
  – orientation (e.g. some lines may be vertical while others are horizontal; see §7.5.3)
  – shape (lines may bend in various ways to follow an uneven surface, or the ends of certain lines may bend upward or downward to accommodate extra characters before the margin or the edge of the support)

3.2.1. Marking up line beginnings

– since epigraphic lines do not necessarily coincide with the intrinsic structure of a text, the physical lines of an inscription are not treated in markup as units with content, but instead, line beginnings are marked as points, using the empty element `<lb/>`
  – this element must always have the attribute `@n`; see §3.2.2
– keep in mind that this element stands for Line Beginnings, not Line Breaks, so even though the two are largely synonymous when used in the context of extrinsic structure, nonetheless
  – `<lb/>` must be used at the beginning of the first line, not only for subsequent lines
  – `<lb/>` must be present even in inscriptions (or textparts, §3.4) consisting of a single line

3.2.2. Numbering lines

– **every physical line** of text in your edition **must have a number** encoded in the `@n` attribute of the corresponding `<lb/>` element
  – line numbering is mandatory even if an inscription (or textpart, §3.4) contains only one line
  – line numbering will be utilised both for display and for machine-readable referencing
– in order to eliminate ambiguity in referencing (both human- and machine-readable), every line number in an XML document **must be unique**
  – except that if your document has textpart divisions (§3.4), then line numbering must be restarted in each textpart, as the requirement of uniqueness only applies within such a division
– by default, **simple line numbers** shall be **Arabic numerals** starting from 1
  – however, line numbers for visually separate incipits may be (or begin with) 0 (see §3.3.3)
– our project uses two schemes of line numbering in documents involving pagelike partitions (§3.5)
  – the preference for either system shall be determined on the level of subcorpora, but may be overridden on a case-by-case basis
  – in the **consecutive scheme**, all line numbers are **simple line numbers** as defined above
    – in this system, the numbering of lines cannot be restarted in pagelike partitions, even though it must always be restarted in boxlike partitions
    – in rare cases (namely, copperplate grants with a lost medial plate encoded without the use of textparts; §5.4.8) you will have to use complex line numbers (see below) even if you normally use the consecutive system
  – in the **repetitive scheme**, line numbering is restarted for each successive pagelike partition
    – to ensure uniqueness, **complex line numbers** must be used in this system
    – these consist of a simple line number as above, preceded by a prefix that is the identifier of the current partition, i.e. the value of the `@n` attribute of the corresponding `<pb/>` or `<milestone/>` element in pagelike partitions, and of the containing `<div type="textpart"/>` element in boxlike ones
      – for example,
        – for a stele inscribed on faces A and B, the lines must be numbered `"A1"`, `"A2"`, etc., and `"B1"`, `"B2"`, etc. on these faces
        – for a set of copper plates with pages 1v and 2r, lines must be numbered `"1v1"`, `"1v2"`, etc., and `"2r1"`, `"2r2"`, etc. on these pages

- should the number of your partitions be a numeral or end with a numeral[9] (including Roman numerals), use a . (period, full stop) as a separator character between the partition number and the simple line number
  - e.g. if your partitions are numbered A1, b1, etc., then your line numbers should be `"A1.1"`, `"A1.2"`, etc., and `"b1.1"`, `"b1.2"`, etc.
- if your subcorpus follows the repetitive scheme, then it is recommended that for consistency's sake you use complex line numbers even on copper plates with a single inscribed page
- also for consistency's sake, if your subcorpus follows the repetitive scheme, then complex numbers should be preferred for numbering lines across boxlike partitions

3.2.3. Placement of line beginnings

- the `<lb/>` element must always be on the same level as the text (see also §8.2.3), i.e. *inside* rather than outside block-level elements representing intrinsic structure; thus,
  - the first line beginning must be encoded after all required block-level elements have been opened
    - e.g. `<lg n="1" met="anuṣṭubh"><l n="a"><lb n="1"/>Āsīt...`
  - if the end of a block-level element coincides with the end of a physical line, the next line beginning element must come at the beginning of the next structural unit, not between the two
    - e.g. `...śāntiM</l></lg><lg n="2" met="upajāti"><l n="a"><lb n="2"/>guptānvaya…`
  - the only exception to this rule is a massive medial lacuna (§5.4.7), where reconstructed line beginnings may be encoded outside block-level containers
- if there is **lost text** both before and after a line beginning, then it may not be possible to determine the exact number of characters lost on either side of the transition
  - if the lacuna is **not restored**, simply encode a gap (§5.4) of unknown or uncertain length at the end of the former and the beginning of the latter line
  - if you **supply the lost text** (as per §5.5), encode the line beginning at its most likely position vis-à-vis the text, and if you feel that the uncertainty of this positioning matters, mention it in your commentary to the text[10]
- be careful with spaces and new lines in your XML code around encoded line beginnings; see §8.1.2 for further details
  - never add a space between the `<lb/>` element and the following text
  - adding a space or starting a new line in your XML file before each `<lb/>` is permitted (but not required), provided that the line beginning coincides with a word break (§3.2.4)
  - it is, however, fully acceptable to start a new line within the `<lb/>` element (whether it interrupts a word or not), if this makes your XML document easier for you to scan while working

3.2.4. Line beginnings interrupting words

- a physical line beginning is deemed to fall inside a word if it occurs at a point other than between two independent words not fused in vowel sandhi, i.e. at a place where you would not be able to add an editorial space (TG §2.6.1), including cases where
  - an initial vowel is fused to the end of the previous line in sandhi (e.g. *tathā/yam*, *maha/ṛṣi*, *asā/v api*; but not *so/yam*)
  - the words before and after the interruption are compounded to one another (e.g. *mahā/rāja*)
  - and even if there is another feature intervening between the two words separated by the line beginning, such as
    - space filler signs (§4.2.5) at the end of the previous line
    - a space imposed on the engraver by physical features (§4.3.5) either before or after the interruption

[9] This will not be the case if you follow the numbering schemes recommended by this Guide, but other numbering schemes are permitted for all kinds of partition when there is good reason for their use.

[10] It is possible in TEI to encode uncertainty regarding the location of a line beginning in a machine-readable way, but the required markup is somewhat complex and we deem that the benefit is not worth the effort.

  - pre-modern deletion (§4.5.1) either before or after the interruption
  - a lacuna (§5.4) before or after the line interruption, provided that the original presence of an interrupted word can be inferred with fair likelihood (see below for details)
- for such line beginnings, the `<lb/>` element signifying the start of the new line must take the attribute `@break` with the value `"no"` to encode the fact that the line beginning does not signify a break in the text
  - this allows hyphens to be automatically displayed at the ends of lines where necessary, and it lets computer processing know that a word continues from one line to the next
- it **may not be possible to decide** for certain whether the line beginning interrupts a word **when text** at the end of the preceding line or at the beginning of the current line **is lost**, badly damaged **or unintelligible** for some other reason
  - in such cases, always use `@break="no"` when you are reasonably certain that an interruption is present, but do not do so if you are uncertain, specifically:
  - when the **end of the previous line is affected**,
    - use `@break="no"` if the current line begins with an incomplete indivisible morpheme or with the final part of what you are certain was a compound word
    - but do not use `@break="no"` if the text in the current line is intelligible (and plausible in context) as it is, even if there is some chance that the previous line contained a prefix or a compound member attached to the extant word
    - the above also applies if the entire previous line, or all text above the current line is lost
  - when the **beginning of the current line is affected**,
    - use `@break="no"` if the end of the previous line is clearly not the end of an independent word
    - but do not use `@break="no"` if the end of the previous line may be the end of an independent word, even if there is a chance that this word continued in the current line
- never start a new line of XML code for line beginnings interrupting words (see §8.1 for further details)
- **never add a hyphen, nor a space** before a line beginning inside a word
- however, if you use **editorial hyphens** for the segmentation of compounds (TG 2.6.2),
  - treat boundaries marked by editorial hyphens as being inside words (as above)
  - and put the editorial hyphen at the start of the new line instead of the end of the previous line
  - e.g. `tomara<lb break="no" n="18"/>-bhindipāla-nārāca`

3.3. Not-quite Partitions

3.3.1. Stuff in margins

- there can be many kinds of "stuff" written in the margins of an inscription, and their encoding requires an editorial decision about their nature
- **premodern additions and corrections** written in a margin shall be handled according to §4.5
- **pagination or foliation** engraved in a margin comprises forme work, to be handled according to §3.3.5
- **auspicious words or symbols** engraved at the beginning of an inscription are spatially offset opening lines (incipits), to be handled according to §3.3.3
- **colophons** engraved below the principal text or vertically in the left or right margin are spatially offset closing lines, to be handled according to §3.3.4

3.3.2. Sectioning with space

- if sections of a reasonably coherent text are separated by vertical (interlinear) space in what is otherwise a fairly well-defined single zone, then, depending on how semantically distinct these sections are, you may choose to
  - encode such sections as pagelike partitions (§3.5), since the actual spatial arrangement of partitions one below the other is irrelevant
  - encode the text as a single unit, ignoring the interlinear space in your edition and only describing it in the layout description
    - in such cases the lines of the inscription must be numbered consecutively throughout the text

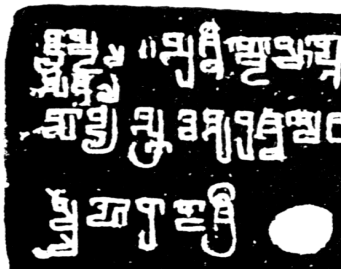
– if the first or last line or few lines of an inscription are set apart visually from the rest, see the following subsections (§3.3.3 and §3.3.4) for some further encoding options

3.3.3. Spatially offset opening sections (incipits)

– opening symbols, words, phrases or stanzas (called *incipit* in the Western tradition) shall need no explicit semantic markup in our practice
  – enclose the contents of the incipit in one or more `<ab>`, `<p>` or `<lg>` elements as applicable (§2)
  – if the text of an incipit is **within the regular field and line structure** of an inscription, then no further markup is necessary
– however, if an incipit is visually set apart from the body text,
  – you may **optionally** use the numbers 01, 02 etc. for the `<lb/>` elements encoding the beginning of the applicable line
    – and assign the number 1 to the first line after the invocation
  – if applicable (i.e. if different from the body text), encode the orientation (§7.5.3) and/or script (§7.5.4) of the opening section
  – all other details of presentation (e.g. line length, relative location, interference with the regular lines) shall be recorded for human readers in your metadata (layout description), but not encoded explicitly
– the same is applicable if an incipit **floats** partly or wholly **outside the principal field**, as in the examples below

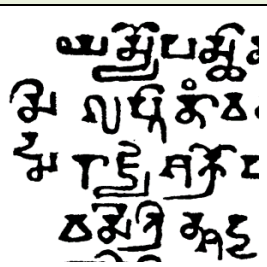
Example 3.3.3.A: incipit of two lines inset in the top left corner

```
<ab>
  <lb n="01"/>dṛṣṭaM
  <lb n="02"/>siddhaM
</ab>
<p>
  <lb n="1"/>agniṣṭomāpto...
  <lb n="2"/>sādyaskra-catur-aśvamedha...
  <lb n="3"/>m mahārāja-śrī...
...
</p>
```



Example 3.3.3.B: incipit written vertically, with upright characters, in the left margin

```
<ab>
  <lb n="01" rend="tb-upright"/>siddhaM
</ab>
<lg n="1" met="sragdharā">
  <l n="a"/><lb n="1"/>yasyopasthāna...</l>
  <l n="b"/><lb n="2"/>guptānāṁ...</l>
  <l n="c"/><lb n="3"/>rājye...</l>
  <l n="d"/><lb n="4"/>varṣe...</l>
</lg>
```



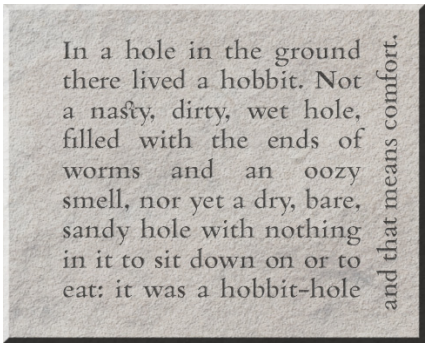
3.3.4. Spatially offset closing lines (colophons)

– in some inscriptions, the last line or two may be written outside the principal field,
  – either because the designer of the inscription wanted to separate a colophon visually from the rest of the text
  – or, occasionally, because the engraver had simply run out of space in the principal field and engraved the final line(s) in a margin or interpolated between the regular lines
– as with incipits, colophons do not require explicit semantic markup
– if the last lines are visually set apart from the body text (whether they are colophons or not), treat them as any other part of the text
  – numbering the lines consecutively after the last regular line
  – the contents of the last line(s) may be incorporated in the last block-level container of the principal text if the two are semantically contiguous

– but if they are semantically distinct (as is the case with colophons proper), it is better to create a new container for the closing section, using `<ab>`, `<p>` or `<lg>` as called for (§2)
– if applicable (i.e. if different from the body text), encode the orientation (§7.5.3) and/or script (§7.5.4) of the opening section
– all other details of presentation (e.g. line length, relative location, interference with the regular lines) shall be recorded for human readers in your metadata (layout description), but not encoded explicitly

Example 3.3.4.A: last line inscribed vertically in the right margin

```
<p>
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  <lb n="8"/>in it to sit down on or to
  <lb n="9"/>eat: it was a hobbit-hole
  <lb n="10" rend="bt-rotated"/>and that means comfort.
</p>
```



3.3.5. Pagination or foliation: “forme work”

– if your inscription has a dominant field with one or more much smaller supplementary additions that cannot be integrated with the principal text, for the purpose of markup these supplementary items are classified as **forme work**
  – the term is borrowed from printing, where *forme* means the frame constructed to hold the blocks of movable type that constitute a page
– epigraphic occurrences of forme work are primarily foliation or pagination marks on copper plates; if you encounter a different feature that you believe ought to be marked up as forme work, please discuss the issue with the authors of the Guide
– as in the case of boxlike partitions (§3.4), the content of forme work is a complete and meaningful unit in itself, but unlike boxlike zones, it is a supplement to (rather than a subunit of) the principal text of an inscription
– forme work items shall be wrapped in the element `<fw>`, with the following mandatory attributes
  – `@n` (even if there is only one forme work item in your document)
    – the value of `@n` shall be the same as the `@n` of the `<pb>` element marking the beginning of the page on which the forme work item appears
  – `@place`, with values as shown on the right

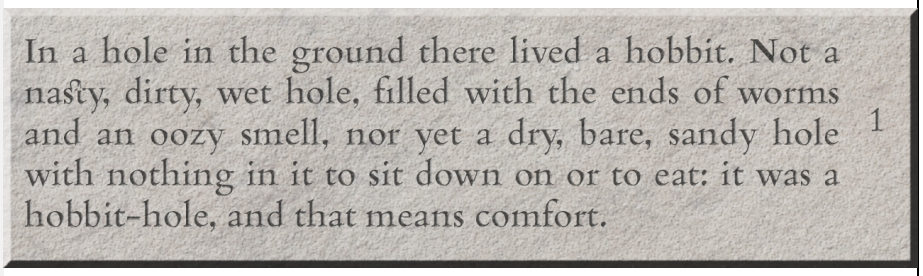
top-left	top	top-right
left	primary inscribed zone	right
bot-left	bottom	bot-right

– if applicable, encode the orientation (§7.5.3) of the forme work
– all other details of presentation (e.g. accurate position, interference with the regular lines) shall be recorded for human readers in your metadata (layout description), but not encoded explicitly
– the `<fw>` element shall appear immediately after the `<pb/>` element marking the start of the page on which the forme work item is found, therefore
  – it must come before the first `<lb/>` element on that page
  – it will normally appear inside block-level containers for intrinsic structure (§2), often interrupting the course of the text within such containers
    – the occurrence of such an interruption is encoded in the page and line beginnings and does not affect the markup for forme work
  – the `<fw>` element may be outside block-level containers when forme work is present on a page whose `<pb/>` element is outside block-level containers, i.e. in the rare but potentially possible cases where a page bears forme work, but the rest of the page is blank (§3.5.2) or lacunose (§5.4.7)

– should your inscription have two (or more) foliation marks on a single page, encode two (or more) `<fw>` elements one after the other, in an order that seems most logical
– for practical purposes the `<fw>` element is a structural container, therefore do not wrap the contents of this element in `<ab>` (or any other container)
– moreover, since foliation marks are not an integral part of the text, **do not mark up line beginnings** within forme work[11]
– numbers used in foliation/pagination must be marked up as usual (§4.2.2, §7.1)

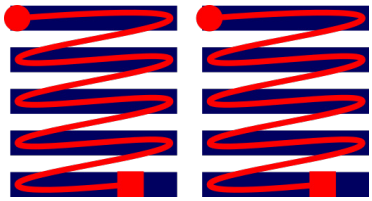
Example 3.3.5.A: foliation in the right margin

```
<p>
  <pb n="1r"/>
  <fw n="1r" place="right">
    <num value="1">1</num>
  </fw>
  <lb n="1"/>In a hole in
the ground...
  <lb n="2"/>nasty,
dirty...
  ...
</p>
```



3.4. Boxlike Partitions: Self-contained Zones

3.4.1. Overview



– **if your text stops** at the end of a zone and something else begins in the next (as in the abstract scheme on the right), you have a **boxlike partition**
  – here, there is a semantic discontinuity between the two zones, and there is no unit of intrinsic structure commencing in one and ending in another
– such spatial partitions of an inscribed text are called boxlike here, because each partition is a functional box enclosing a discrete segment of text
– since the hierarchy of such partitions does not overlap with the hierarchy of the text's internal structure, they can be encoded in XML as containing elements called textpart divisions
– in our practice, the encoding for boxlike partitions shall only be used in warranted cases, specifically:
  – **copperplate sets with an inscribed seal** (regardless of whether the seal is soldered to a plate, attached with a ring to a set of plates, or currently detached)
    – see Case study 2A in Appendix C for an illustration
  – **non-contiguous fragments**, where the physical structure of the lost intervening fragments cannot be reconstructed, especially when even the order in which the fragments must be read is doubtful
    – see Example 3.4.5.A for an illustration
    – keep in mind that fragments for which it is possible to reconstruct the structure of the lost connecting section do not require encoding as textparts, nor do copperplate sets with a known number of medial plates lost
      – see §5.4.7 for advice on encoding massive lacunae where the structure can be restored
  – inscriptions consisting of **visually distinct parts that convey the same message in two (or more) languages**, if it is deemed necessary to encode these as a single inscription
    – keep in mind that this does not apply to all bilingual (or multilingual) inscriptions: if different parts of a single text are written in different languages, then the use of textparts is not warranted
    – see §7.2.1 about multilingual inscriptions

[11] Should you come across a multi-line foliation mark, contact the authors and the XML-TEI Data Manager with the details to devise a solution.

– beyond the specific cases set out above, boxlike partitions are only warranted when there is no obvious order in which the zones of text ought to be read, but there is nevertheless good reason for treating them as a single document
– in any other case where you think boxlike partitions may be relevant, consider carefully whether this encoding method is essential, or alternatives could be used
  – if the connection between the texts is weak, preferably encode separate inscriptions (in separate XML documents), especially if there is reason to believe they were created on separate occasions
  – if the connection between the texts is strong, encode a pagelike partition (§3.5) between them, especially if it makes sense to read them in a particular sequence

3.4.2. Encoding boxlike partitions

– each boxlike partition must be wrapped in the element `<div type="textpart">`
  – note the mandatory presence and value of `@type`
  – see §3.4.3 below about additional attributes and optional headers
– note that the markup represents only the fact that such text partitions exist, but contains no encoded information about their relative positions and sizes
  – such information shall be described for human readers in the metadata of your inscription
– if your `<div type="edition">` includes a `<div type="textpart">`, then **all text** within the edition must be contained within textpart divisions
  – the technical expression for this is that the textpart divisions must *tessellate*, i.e. cover the entire surface of the text with no gaps (and, of course, no overlaps)
  – the practical purport is that if you create one textpart division for a section of an inscription, then you must also create another textpart division to wrap the remainder of the text
– note that in principle, textpart div elements *could* be nested within other textpart div elements; however, to avoid complications (in markup and referencing), our project policy is never to do so
  – when encoding a structurally complex inscription, instead of resorting to textparts within textparts, try to make use of visually offset intrinsic units (§3.3) and pagelike partitions (§3.5)
  – if you encounter a case where nested textpart divisions seem to be the best solution, please discuss it with the authors of the Guide and the XML-TEI data manager
– encode textparts in the order you deem to be the logical reading order
  – for the sake of consistency throughout the corpus, inscribed **seals** of copper plates shall always be encoded **before the plates** themselves
– within each textpart division, use structural and other markup as you would elsewhere; this includes in particular
  – wrapping all text in block-level containers to represent intrinsic structure (§2)
  – marking up line beginnings (§3.2) even if a given partition consists of just one line
  – numbering all line beginnings (§3.2.2) and any stanzas (§2.3.3) even if a given partition contains only one

3.4.3. Textpart identification: subtype, number and headers

– every textpart division must carry the **mandatory attribute** `@n`
  – each textpart in an XML document must have a unique number
  – uppercase Latin letters are generally recommended for numeration, but any scheme may be used depending on your preference and the conventions of your specific field
– in addition to numbering, the **optional attribute** `@subtype` may be used to encode the nature of textparts
  – the use of this attribute is not mandatory, but it is strongly recommended when multiple textparts are of the same nature
  – use any single word for the value of `@subtype`
    – we recommend using values that describe the general nature of a unit rather than its function or appearance; preferably, use one of the following:
      – `"fragment"` for fragments bearing non-contiguous text

- "face" for the surfaces of an object with no more than 4 sides
- "facet" for the surfaces of an object with a polygonal cross-section
- "faces" and "facets" in texts where each line of a textpart continues across two or more surfaces such as the frontal and lateral face of a four-sided stele
- "zone" for visually distinct zones on a single two-dimensional surface
- "column" for zones placed side by side and generally taller than they are wide (as in newspaper columns)
- "item" for physically distinct objects such as architectural elements, e.g. when an inscription is engraved on two pillars

- if you are certain none of the above are satisfactory, you may use other values with the following constraints:
  - the value should be in lowercase throughout to avoid inconsistencies; display can easily be rendered with a capital initial
  - the value should not include spaces; if you absolutely need a multi-word value, use an underscore (_) instead of a space, which can be rendered as a space in display
  - having introduced a custom value, try to use it consistently and send the value and a short definition/description of the case where you have used it to the authors of this Guide, so it can be included in later versions

- the above two attributes will be used in the generation of a title when your digital edition is displayed (see Example 3.4.3.A below), and @n may also be utilised for internal references that can be processed by a computer
- to add further flexibility to the titles displayed for textparts, you may also add the **optional element** <head xml:lang="eng"> immediately after the start tag of a textpart division
  - such elements in our editions will by default be regarded as editorial and therefore need not be marked up explicitly as such; the mandatory language attribute makes it sufficiently clear that their content is not part of the original text
  - the use of this element is recommended when the textparts of an inscription are different in nature, so they cannot be conveniently described by a combination of subtype and number: in this case omit @subtype and add a <head>
  - you are free to create headers as you deem best for the inscription you are editing, but for the sake of consistency it is generally recommended that you stick to concise headers in English, such as
    - "Seal" and "Plates" for a copperplate charter with an inscribed seal
    - "Head", "Halo", "Back" and "Pedestal" (etc.) on a statue
  - the content of editorial headers will replace the title auto-generated from @n (and @subtype, if present) in display (see Example 3.4.3.B to Example 3.4.3.D below)
    - however, the use of the attribute @n on the element remains mandatory even if a <head> is present, and @subtype remains recommended when multiple textparts are of the same nature
  - the contents of the editorial heading will not be altered in display, so
    - please use a capital initial and feel free to include spaces, additional capitals and punctuation as necessary
    - however, to avoid complications, do not use any further markup within this element, except the element <foreign> (§10.3.3), which you may employ if you deem necessary

Example 3.4.3.A: textpart identification, two or more fragments with non-contiguous text

```
<div type="textpart" subtype="fragment" n="A">
  ...
<div type="textpart" subtype="fragment" n="B">
  ...
```

➢ auto-generated headings will show "Fragment A", "Fragment B", etc.

Example 3.4.3.B: textpart identification, two or more fragments with non-contiguous text

```
<div type="textpart" subtype="fragment" n="1"><head xml:lang="eng">Upper left corner</head>
  ...
<div type="textpart" subtype="fragment" n="2"><head xml:lang="eng">A small piece not adjacent to
any edge</head>
  ...
```

➢ explicitly encoded headings will show “Upper left corner”, “A small piece not adjacent to any edge”, etc.

Example 3.4.3.C: textpart identification, faces of a quadrangular stele

```
<div type="textpart" subtype="face" n="A"><head xml:lang="eng">Frontal Face</head>
  ...
<div type="textpart" subtype="face" n="b"><head xml:lang="eng">Lateral Face</head>
  ...
```

➢ explicitly encoded headings will show “Frontal Face”, “Lateral Face”, etc.
➢ see Case Study 1 (A and B) in Appendix C for a similar stele where each line runs across two adjacent faces

Example 3.4.3.D: textpart identification, set of copper plates with two inscribed seals

```
<div type="textpart" n="A"><head xml:lang="eng">First seal</head>
...
<div type="textpart" n="B"><head xml:lang="eng">Second seal</head>
...
<div type="textpart" n="C"><head xml:lang="eng">Plates</head>
...
```

➢ explicitly encoded headings will show “First seal”, “Second seal” and “Plates”
➢ see Case study 2A in Appendix C for the full markup of a set of plates with one seal

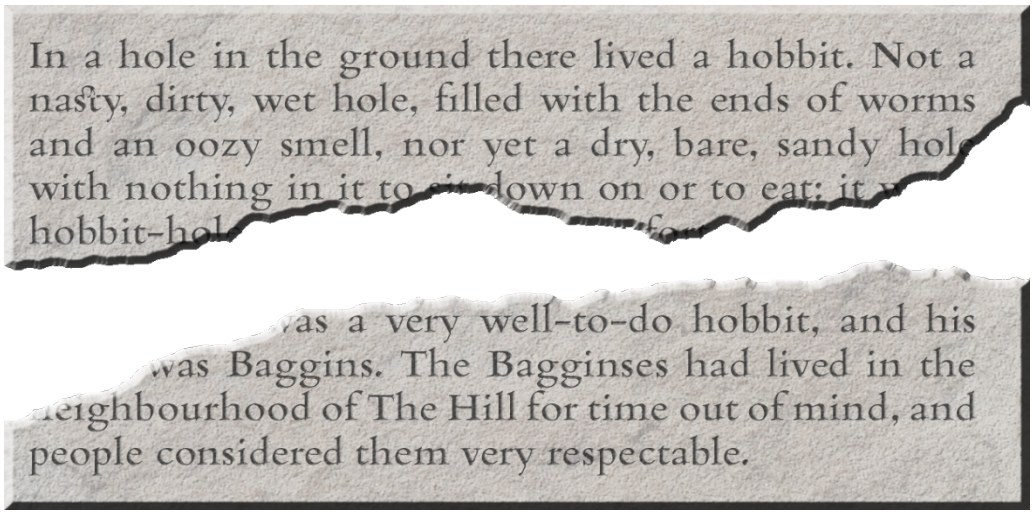
3.4.4. Numbered elements in textparts

– when your document is divided into textparts, you must mandatorily restart the numbering of all of the following elements in each new textpart, since there is as a rule no straightforward sequence of progression from one textpart to another
  – physical lines
  – stanzas
  – pages[12]
– however, do not restart the numbering of the following elements: should they occur in more than one textpart, keep their numbers unique throughout your XML document
  – pagelike milestones (§3.5.3)
  – gridlike milestones (§3.6)

[12] Pages may occur in more than one textpart in copperplate sets divided into textparts because of an unknown number of lost medial plates (§5.4.8). Another theoretically possible scenario might be two sets of copper plates bound with a single ring and edited as a single text.

3.4.5. Full markup example for boxlike partitions

Example 3.4.5.A: textparts for non-contiguous fragments

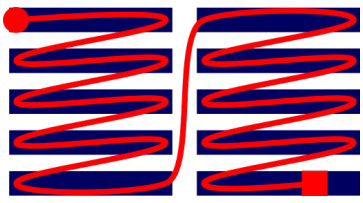


- here we have two fragments of a slab, which are clearly from the top and bottom of a single inscription, but there is no way to know how much text is lost between the two
- therefore the fragments must be encoded as boxlike partitions, with the lost text handled as per §5.4.7

```
<div type="textpart" subtype="fragment" n="A">
  <p part="I"><!--Paragraph interrupted by the lacuna marked up as an initial part. -->
    <lb n="1"/>In a hole in the ground there lived a hobbit. Not a
    <lb n="2"/>nasty, dirty, wet hole, filled with the ends of worms
    <lb n="3"/>and an oozy smell, nor yet a dry, bare, sandy <unclear>hole</unclear>
     <lb n="4"/>with nothing in it to <supplied
reason="lost">s</supplied><unclear>i</unclear><supplied reason="lost">t</supplied> down on or to
eat: it <unclear>w</unclear><supplied reason="lost">as a</supplied>
    <lb n="5"/>hobbit-hol<supplied reason="lost">e</supplied> <gap reason="lost" quantity="18"
unit="character" precision="low"/><unclear>f</unclear><unclear cert="low">o</unclear><gap
reason="lost" quantity="12" unit="character" precision="low"/>
  </p>
</div>
<!--Lost lines are not encoded in either fragment, because the presence of textparts implies them
§5.4.7. -->
<div type="textpart" subtype="fragment" n="B">
  <!--Line numbering is reset to 1 in the second textpart §3.4.4. -->
  <p part="F"><!--Paragraph interrupted by the lacuna marked up as a final part. -->
    <lb n="1"/><gap reason="lost" quantity="12" unit="character"
precision="low"/><unclear>w</unclear>as a very well-to-do hobbit, and his
    <lb n="2"/><gap reason="lost" quantity="5" unit="character"
precision="low"/><unclear>w</unclear>as Baggins. The Bagginses had lived in the
    <lb n="3"/><unclear>ne</unclear>ighbourhood of The Hill for time out of mind, and
    <lb n="4"/>people considered them very respectable
  </p>
</div>
```

3.5. Pagelike Partitions: Text Flows through Successive Zones

3.5.1. Overview



– **if your text flows on from one zone to the next** (as in the abstract scheme on the right), you have a **pagelike partition**
  – here the text, having filled one zone completely (normally from top to bottom), continues seamlessly at the beginning (normally the top) of the next zone, so that the spatial zones do not correspond to units of the text's intrinsic structure

  - a single virtual text field is thus created from a string of successive zones which may appear in any spatial arrangement
- spatial partitions which are disregarded by the intrinsic structure of the text are called pagelike here, since they functionally behave like the pages of a book containing a single continuous text
- in pagelike partitions, units pertaining to the intrinsic structure of a text (such as verse stanzas and prose paragraphs) may begin in one zone and end in the next
  - this, however, is not a criterion: a pagelike partition may also be coterminous with units of intrinsic structure
- to avoid the problem of overlapping hierarchies (§1.3.3), such partitions are handled in a manner similar to physical lines (§3.2): instead of wrapping each zone in tags, empty elements are employed to mark the spot where each zone begins
  - these elements may be page beginnings or pagelike milestones as explained in the following subsections
- epigraphic examples of pagelike partitions include
  - text laid out in consecutively readable zones laid out in any arrangement on a single surface
  - text laid out in consecutively readable zones on multiple faces of a three-dimensional object (e.g. stele or pillar)
  - text laid out in consecutively readable zones on multiple linked objects (e.g. copperplate sets; two jambs of a doorway)

3.5.2. Genuine pages

- to encode **genuine pages**, as in copper plates (and manuscript folios), use the empty element `<pb/>`
  - this element signifies a page beginning (and not a page break, though the two are in many cases synonymous)
  - as with line beginnings, `<pb/>` elements must be used to mark the beginning of every page including the first
- **every copperplate page**[13] in your edition **must have a number** encoded in the `@n` attribute of the corresponding `<pb/>` element
  - the **recommended values are** 1r, 1v, 2r, 2v etc., referring to the recto (front) and verso (back) face of plates identified with Arabic numerals, e.g. `<pb n="1r"/>`
  - should pages occur in more than one textpart of a complex inscription, page numbers must be reset in each textpart
  - page numbers will be utilised for display and may in the future be utilised for machine-readable referencing
  - if you have a good reason to do so, you may opt to use a different numbering scheme for pages with the following constraints:
    - the value of `@n` must not contain a space (use an underscore _ instead if a space is essential)
    - each page must have a unique number within your edition (or, if applicable, within a textpart division)
  - see §3.3.5 about encoding any original pagination or foliation
- plates must always be treated as having two pages per folio, even if one of these pages is blank; this will facilitate the eventual linking of images (including images of blank faces) to individual pages of the digital edition
  - blank faces shall be encoded as the corresponding `<pb/>` element[14]

[13] In this section and elsewhere when discussing copper plates, “plate” is used in reference to one discrete physical item; “face” for one side of a plate as a physical surface; “page” as a unit of text that is inscribed on one face of a plate; and “folio” as an abstract unit of text comprised of two pages that belong to a single plate. Of these terms, only “page” has an exact markup equivalent.

[14] For a single copper plate inscribed on one face only, designate the inscribed face as the recto, and the blank face as the verso. For sets of copper plates where the first and/or last plate is only inscribed on one face, designate the blank faces to be the outer faces of the set, i.e. the recto of the first plate and the verso of the last plate.

- the `<pb/>` elements for first/last blank pages should be placed just inside the start/end tag of the enclosing division (i.e. the edition division or a textpart division)
  - thus, these `<pb/>` elements will be outside block-level containers in spite of the general rule (§8.2.3) that such empty elements must be inside block-level containers
- see Case study 2 (A, B and C) in Appendix C for an illustration of pages in an EpiDoc document

3.5.3. Other pagelike zones

- to encode physically separate **inscribed zones where text flows on** from the bottom of one to the top of the next, use the empty element `<milestone type="pagelike"/>`
  - a "milestone" is a generic element of which `<lb/>` and `<pb/>` (and a few others permitted by TEI but not used in our project) are special cases
  - the specialised milestones `<lb/>` and `<pb/>` implicitly specify the nature of the transition (viz. line and page) they represent
  - the generic `<milestone/>`, on the other hand, must carry the attribute `@unit` to encode the nature of the transition explicitly; see §3.5.4 below for values recommended in pagelike partitions
  - the mandatory attribute `@type` with the value `"pagelike"` serves in our corpus to explicitly distinguish these elements from other milestones used in an edition (§3.6)
- zones may be arranged in any pattern on a single surface, or on several faces of a three-dimensional object
- as with line beginnings, milestones must be used to mark the beginning of every zone including the first
- see Example 3.5.7.A for a full illustration of columnlike zones in an EpiDoc document, and Case study 1 (A and B) in Appendix C for a more complex scenario

3.5.4. Zone identification: unit, number and label

- every zone encoded with a `<milestone/>` element must **mandatorily** carry the **attributes** `@n` and `@unit`
  - both of these attributes will be used to generate a title for partitions when your digital edition is displayed (see Example 3.5.4.A below), and may also be utilised for internal references that can be processed by a computer
    - this title will probably be displayed as a heading in diplomatic editions, but as an inline label (so as not to break the flow of text) in logical editions
- the `@unit` of a milestone encoding a pagelike partition shall be a single word describing the nature of the transition in the same way as the `@subtype` of textpart divisions (§3.5.4)
  - we recommend using values that describe the general nature of a unit rather than its function or appearance; preferably, use one of the following:
    - `"face"` for the surfaces of an object with no more than 4 sides
    - `"facet"` for the surfaces of an object with a polygonal cross-section
    - `"faces"` and `"facets"` in texts where each line of a pagelike zone runs across two or more surfaces such as the frontal and lateral face of a four-sided stele
      - gridlike partitions (§3.6) may be optionally used to encode the boundary of each face constituting a pagelike zone of this kind
    - `"zone"` for visually distinct zones on a single two-dimensional surface
    - `"column"` for zones placed side by side and generally taller than they are wide (as in newspaper columns)
    - `"item"` for physically distinct objects such as architectural elements, e.g. when an inscription is engraved on two pillars or doorjambs
  - if you are certain none of the above are satisfactory, you may use other values with the following constraints:
    - the value should be in lowercase throughout to avoid inconsistencies; display can easily be rendered with a capital initial
    - the value should not include spaces; if you absolutely need a multi-word value, use an underscore (_) instead of a space, which can be rendered as a space in display

  - having introduced a custom value, try to use it consistently and send us the value and a short definition/description of the case where you have used it, so it can be included in later versions of this guide
- every pagelike milestone element in an edition must have a unique identifier encoded in its **attribute** `@n`
  - uppercase Latin letters are generally recommended for identification, but any scheme may be used depending on your preference and the conventions of your specific field
    - in particular, feel free to use
      - the uppercase letters N, S, E, W to indicate cardinal directions
      - lowercase letters alternating with uppercase ones to denote major/frontal and minor/lateral faces of a three-dimensional object such as a Southeast Asian stele, e.g. A, b, C and d (for faces inscribed as separate zones); or Ab and Cd (where pairs of faces constitute single virtual zones)
  - in addition to the generation of titles, this attribute will be utilised for internal references that can be processed by a computer
  - should you need to encode pagelike milestones with two or more different units (e.g. columns and faces) within a single document, use a different numeration scheme for the two in order to keep their `@n` attributes unique
- to add further flexibility to the titles displayed for zones, you may also add the **optional element** `<label xml:lang="eng">` immediately after the `<milestone/>` element
  - such elements in our editions will by default be regarded as editorial and therefore need not be marked up explicitly as such; the mandatory language attribute makes it sufficiently clear that their content is not part of the original text
  - only add labels to zones if you find that the combination of `@unit` and `@n` cannot produce a sufficiently meaningful title; complex details such as the size and relative position of zones should be described in the metadata, not encoded within the edition
  - for the sake of consistency it is recommended that you stick to concise labels in English
  - the content of editorial labels will replace the title auto-generated from `@unit` and `@n` in display (see Example 3.5.4.B below)
    - however, the use of the attributes `@unit` and `@n` on the `<milestone/>` element remains mandatory even if a `<label>` is present
  - the contents of the label will not be altered in display, so
    - please use a capital initial and feel free to include spaces, additional capitals and punctuation as necessary
    - however, to avoid complications, do not use any further markup within this element, except the element `<foreign>` (§10.3.3), which you may employ if you deem necessary

Example 3.5.4.A: zone identification, two faces of an object

```
<milestone type="pagelike" unit="face" n="A"/>
  ...
<milestone type="pagelike" unit="face" n="B"/>
  ...
```

➢ auto-generated headings will show “Face A”, “Face B”, etc.

Example 3.5.4.B: zone identification, two doorjambs

```
<milestone type="pagelike" unit="item" n="N"/><label xml:lang="eng">Northern
Doorjamb</label>
  ...
<milestone type="pagelike" unit="item" n="S"/><label xml:lang="eng">Southern
Doorjamb</label>  ...
```

➢ explicitly encoded headings will show “Northern Doorjamb”, “Southern Doorjamb”, etc.

3.5.5. Placement of page and zone beginnings

- keep in mind that the elements `<pb/>` and `<milestone/>` do not replace line beginnings, which must always be encoded as per §3.2, immediately after the page or column beginning

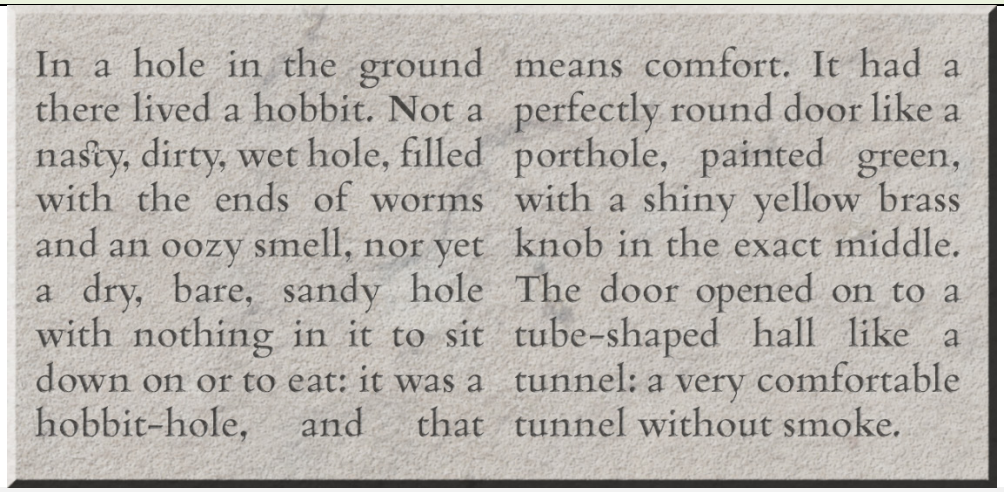
  - except that in rare cases it is possible for a page or zone beginning not to be followed by a line beginning (e.g. when a medial plate of a set is lost, but the page structure is reconstructed for it)
- the elements `<pb/>` and `<milestone/>` must always be on the same level as the text (see also §8.2.3), i.e. *inside* rather than outside block-level elements representing intrinsic structure; thus,
  - the first such element must be encoded after all required block-level elements have been opened
    - e.g. `<lg n="1" met="anuṣṭubh"><l n="a"><pb n="1r"/><lb n="1"/>Āsīt...`
  - if the end of a block-level element coincides with the end of a pagelike partition, the next line beginning element must come at the beginning of the next structural unit, not between the two
    - e.g. `...śāntiM</l></lg><lg n="2" met="upajāti"><l n="a"><pb n="1v"/><lb n="2"/>guptānvaya…`
  - the only exception to this rule is the case of lost medial plates (§5.4.8), where reconstructed page beginnings may be encoded outside block-level containers
- never add a space in your XML document after a page or zone break; see §8.1.2 for further details
  - adding a space or starting a new line before such a break is permitted (but not required), provided that it coincides with a word break
  - when a new page or zone begins inside a word, this must be explicitly encoded in the `<pb/>` or `<milestone/>` element in the same way as (and in addition to) the `<lb/>` element marking the start of the first line of the new unit, i.e. using the attribute `@break` with the value `"no"`
    - e.g. `tomara<pb n="2" break="no"/><lb break="no" n="18"/>bhindipālanārāca`
  - your XML file must never contain a space or a new line before a pagelike partition that interrupts a word
  - see §3.2.4 for further details of what qualifies as an interrupted word
- if you use editorial hyphens for the segmentation of compounds (see TG §2.6.2), you must remember to put the editorial hyphen at the start of the new line in the new page/column/zone
  - e.g. `tomara<pb n="2" break="no"/><lb break="no" n="18"/>-bhindipāla-nārāca`

3.5.6. Numbered elements in pagelike partitions

- as set out under §3.2.2, **physical line** numbering may be either
  - consecutive throughout successive pagelike partitions, or
  - restarted in each pagelike partition, provided that complex line numbers are used, which incorporate the number of the page or zone
- stanzas should be generally numbered throughout a text with pagelike partitions, but, as permitted under §2.3.3, you may optionally reset stanza numbering in each new partition in order to follow the numbering scheme of a previous edition or the conventions of your specific field

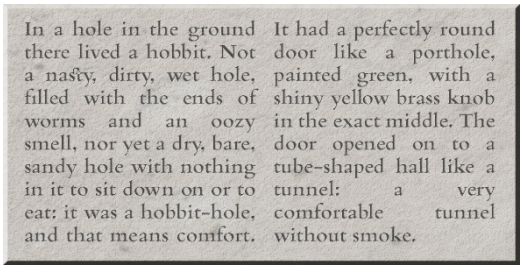
3.5.7. Full markup example for pagelike partitions

Example 3.5.7.A: text in two columns



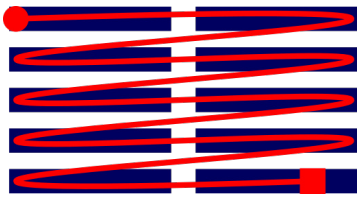

```
<p>
  <milestone type="pagelike" unit="column" n="A"/>
  <lb n="1"/>In a hole in the ground
  <lb n="2"/>there lived a hobbit. Not a
  <lb n="3"/>nasty, dirty, wet hole, filled
  <lb n="4"/>with the ends of worms
  <lb n="5"/>and an oozy smell, nor yet
  <lb n="6"/>a dry, bare, sandy hole
  <lb n="7"/>with nothing in it to sit
  <lb n="8"/>down on or to eat: it was a
  <lb n="9"/>hobbit-hole, and that
  <milestone type="pagelike" unit="column" n="B"/><!-- Line numbers continue in the second
column. Alternatively, the lines numbered 1 to 9 here could be A1 to A9, and those numbered 10 to
18 here could be B1 to B9. See §3.2.2 and §3.5.6 for guidance, and Case study 1 in Appendix C for
an illustration of such numbering. -->
  <lb n="10"/>means comfort. It had a
  <lb n="11"/>perfectly round door like a
  <lb n="12"/>porthole, painted green,
  <lb n="13"/>with a shiny yellow brass
  <lb n="14"/>knob in the exact middle.
  <lb n="15"/>The door opened on to a
  <lb n="16"/>tube-shaped hall like a
  <lb n="17"/>tunnel: a very comfortable
  <lb n="18"/>tunnel without smoke.
</p>
```

- in the illustration above, the partition occurs within a sentence, and it is therefore *technically impossible* to encode it as a boxlike partition (§3.4)
- a partition may, however, coincide with a semantic boundary as in the slightly altered illustration here
- while it is technically possible to encode a boxlike partition in this latter case, our practice shall be always to encode pagelike partitions except in the cases explicitly set out under §3.4.1 above
- therefore, the illustration on the right must also be encoded as a pagelike partition



3.6. Gridlike Partitions: Text Runs Across Contiguous Zones

3.6.1. Overview



– **if your text runs across zone boundaries** (as in the abstract scheme on the right), you have a **gridlike partition**
  – here each line of text, having reached the edge (normally the right margin) of a zone, continues at the edge (normally the left margin) of the next zone; the next line in turn begins in the first zone
    – a single virtual text field is here created from a patchwork of zones which share a vertical boundary
  – spatial partitions that cut across the lines of an inscription are called gridlike here because the spatial structure behaves like a grid projected onto the inscription's own layout
  – as with pagelike partitions, units pertaining to the intrinsic structure of a text (such as verse stanzas and prose paragraphs) may begin in one zone and end in another
    – to avoid overlapping hierarchies (§1.3.3), such partitions are also encoded with the aid of empty elements employed to mark the spot where each patch begins
    – but unlike the case of pagelike partitions, the empty elements employed for this purpose must be reiterated within each affected epigraphic line
  – epigraphic examples of gridlike partitions include text engraved on
    – a simplex (flat or curved) surface vertically segmented into units where each line runs across two or more such quasi-columns (which often correspond to metrical units such as verse lines), as illustrated in Example 3.6.6.A
    – a complex surface (such as that constituted of several facets of a polygonal pillar) with each line running across two or more subsurfaces
    – a composite surface (such as several architectural blocks) with each line running across several blocks, as illustrated in Example 3.6.6.B
    – a broken support where a fracture cuts across some or all lines, as illustrated in Example 3.6.6.C

3.6.2. Encoding gridlike partitions

– gridlike partitions may be encoded with the element `<milestone/>` introduced under §3.5 above, but with the following differences
  – gridlike milestones shall not carry the attribute `@type`
  – the `<label>` element is not permitted in conjunction with these milestones
  – such milestones will typically be iterated many times in a single document, whereas a milestone representing a particular pagelike transition point will always appear only once

3.6.3. Gridlike milestone identification: unit and number

– as in pagelike partitions, every gridlike area encoded with a `<milestone/>` element must **mandatorily** carry the **attributes** `@n` and `@unit`
  – both of these attributes will be used in the generation of a title when your digital edition is displayed, which will always be shown as an inline label so as not to break the flow of the text
– the `@unit` of a milestone encoding a gridlike partition shall be a single word describing the nature of the transition in a way similar to that of pagelike partitions (§3.5)
  – we recommend using values that describe the general nature of a unit rather than its function or appearance; preferably, use one of the following:
    – `"fragment"` for objects with two or more extant inscribed fragments
    – `"block"` for inscriptions on physically separate architectural blocks
    – `"face"` for the sides of an object with no more than 4 sides
    – `"facet"` for the sides of an object with a polygonal cross-section
    – `"area"` for patches of text demarcated by some physical feature

  - `"column"` for zones placed side by side and generally taller than they are wide (as in newspaper columns)
- if you are certain none of the above are satisfactory, you may use other values with the following constraints:
  - the value should be in lowercase throughout to avoid inconsistencies; display can easily be rendered with a capital initial
  - the value should not include spaces; if you absolutely need a multi-word value, use an underscore (_) instead of a space, which can be rendered as a space in display
  - having introduced a custom value, try to use it consistently and send us the value and a short definition/description of the case where you have used it, so it can be included in later versions of this guide
- every gridlike milestone element in an edition must have a number encoded in its **attribute** `@n`
  - the number referring to every zone of text should be unique, but, as noted above, gridlike milestones with a given combination of `@unit` and `@n` will normally be iterated several times in a document, namely once in every line that touches the zone to which that combination pertains
  - lowercase Latin letters are generally recommended for numeration, but any scheme may be used depending on your preference and the conventions of your specific field
    - in particular, feel free to use lowercase letters alternating with uppercase ones to denote major/frontal and minor/lateral faces of a three-dimensional object such as a Southeast Asian stele, e.g. A, b, C and d
  - in addition to the generation of titles, this attribute may be utilised for internal references that can be processed by a computer
  - should you need to encode gridlike milestones with two or more different units within a single document (e.g. `"column"` alternating with `"fragment"` to encode an inscription on whose original gridlike layout a secondary gridlike layout was superimposed by fragmentation), use a different numeration scheme for the two in order to keep thee `@n` attributes of each patch of text unique

3.6.4. Gridlike partitions interrupting words

- just as line, page and zone beginnings, gridlike milestones must take the attribute `@break="no"` if the transition interrupts a word
  - this applies even if the milestone is right next to another interruption (such as a line beginning) that already has `@break="no"`
- if you use editorial hyphens for the segmentation of compounds (see TG §2.6.2), the editorial hyphen shall be placed after the milestone, at the beginning of the text that follows it
- your XML file must never contain a space or a new line before a milestone that interrupts a word; see §8.1.2 for more details

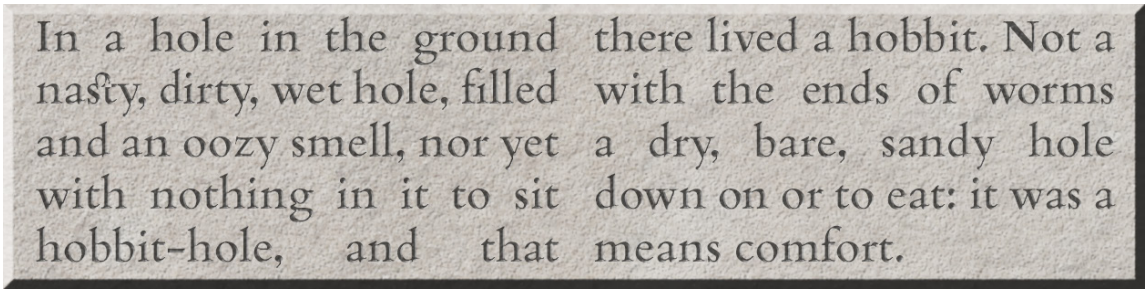
3.6.5. When to encode gridlike partitions

- encoding gridlike partitions with milestones is **not mandatory** and should be applied on a case-by-case basis, judging the feasibility of encoding versus the anticipated usefulness of having such partitions represented in the edition
  - such representation is particularly useful if some elements of description apply only to certain partitions (e.g. certain fragments are kept in a different place; or certain facets of the support are in a different state of preservation)
- this encoding is **strongly recommended for composite** (physically disjoined) **surfaces** such as
  - **fragments**, especially if they have not been reconstituted
    - but only if it is possible to tell which segments of text belong to which fragments
    - and only if the number of fragments involved is not inordinately large
  - **building blocks**, especially if they are not currently assembled
- this encoding is **recommended for visually demarcated areas** on a simplex surface, such as
  - quasi-columns consisting of a metrical unit (e.g. verse line)
- this encoding is recommended **only if deemed useful for complex surfaces** such as

– two or more adjacent faces of a stele or pillar with a rectangular or polygonal cross-section
– if you opt not to encode milestones in any of the above cases, simply encode the text as if it occupied a simple surface, and describe the layout in as much detail as you wish in your metadata

3.6.6. Full markup examples for gridlike partitions

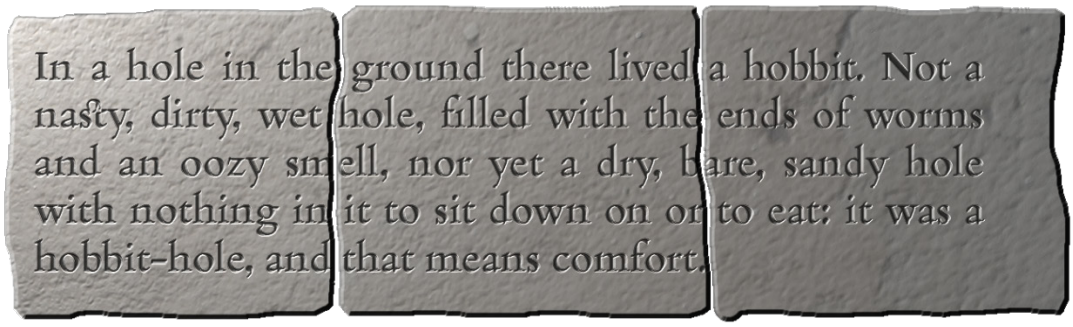
Example 3.6.6.A: gridlike partitions for verse inscribed in quasi-columns



- for the sake of the illustration assume that the sample text is verse, with one stanza (of two lines) occupying one epigraphic line
- attributes for `<lg>` and `<l>` are omitted in this illustration to reduce clutter

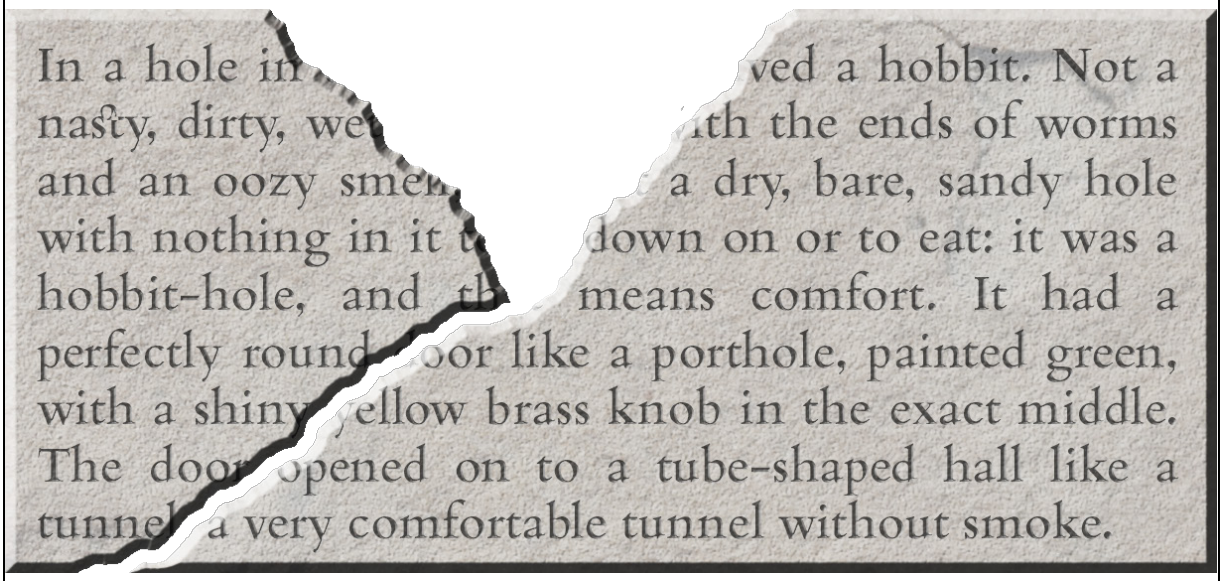
```
<lg>
  <l><lb n="1"/><milestone unit="zone" n="a"/>In a hole in the ground</l>
  <l><milestone unit="zone" n="b"/>there lived a hobbit. Not a</l>
</lg>
<lg>
  <l><lb n="2"/><milestone unit="zone" n="a"/>nasty, dirty, wet hole, filled</l>
  <l><milestone unit="zone" n="b"/>with the ends of worms</l>
</lg>
<lg>
  <l><lb n="3"/><milestone unit="zone" n="a"/>and an oozy smell, nor yet</l>
  <l><milestone unit="zone" n="b"/>a dry, bare, sandy hole</l>
</lg>
<lg>
  <l><lb n="4"/><milestone unit="zone" n="a"/>with nothing in it to sit</l>
  <l><milestone unit="zone" n="b"/>down on or to eat: it was a</l>
</lg>
<lg>
  <l><lb n="5"/><milestone unit="zone" n="a"/>hobbit-hole, and that</l>
  <l><milestone unit="zone" n="b"/>means comfort.</l>
</lg>
```

Example 3.6.6.B: gridlike partitions for text inscribed across architectural blocks




```
<p>
  <lb n="1"/><milestone unit="block" n="a"/>In a hole in the<milestone unit="block" n="b"/>
ground there lived<milestone unit="block" n="c"/> a hobbit. Not a
  <lb n="2"/><milestone unit="block" n="a"/>nasty, dirty, wet<milestone unit="block" n="b"/>
hole, filled with the<milestone unit="block" n="c"/> ends of worms
  <lb n="3"/><milestone unit="block" n="a"/>and an oozy sm<milestone unit="block" n="b"
break="no"/>ell, nor yet a dry, b<milestone unit="block" n="c" break="no"/>are, sandy hole
  <!--Notice the use of @break="no" for two milestones in line 3. -->
  <lb n="4"/><milestone unit="block" n="a"/>with nothing in<milestone unit="block" n="b"/> it to
sit down on or<milestone unit="block" n="c"/> to eat: it was a
  <lb n="5"/><milestone unit="block" n="a"/>hobbit-hole, and<milestone unit="block" n="b"/> that
means comfort.
</p>
```

Example 3.6.6.C: gridlike partitions for contiguous fragments



- here, two extant fragments of a slab can be joined because they share some lines, though a smaller missing fragment gives rise to gaps in other lines
- the fragments are optionally encoded as gridlike milestones
- the lacunae in the first five lines are arbitrarily allocated to one of the encoded fragments (fragment a, in the code below)
- but restorations of partially lost words are always allocated to the fragment bearing their extant segments (thus, to fragment b in lines 1 and 2)

```
<p>
  <lb n="1"/><milestone unit="fragment" n="a"/>In a hole in<gap reason="lost" quantity="17"
unit="character" precision="low"/><milestone unit="fragment" n="b"/><supplied
reason="lost">li</supplied>ved a hobbit. Not a
  <lb n="2"/><milestone unit="fragment" n="a"/>nasty, dirty, we<supplied reason="lost"
precision="low">t</supplied><gap reason="lost" quantity="14" unit="character"
precision="low"/><milestone unit="fragment" n="b"/><supplied
reason="lost">w</supplied><unclear>i</unclear>th the ends of worms
  <lb n="3"/><milestone unit="fragment" n="a"/>and an oozy sme<unclear>ll</unclear><gap
reason="lost" quantity="10" unit="character" precision="low"/><milestone unit="fragment"
n="b"/>a dry, bare, sandy hole
  <lb n="4"/><milestone unit="fragment" n="a"/>with nothing in it <unclear>t</unclear><gap
reason="lost" quantity="6" unit="character" precision="low"/><milestone unit="fragment"
n="b"/>down on or to eat: it was a
  <lb n="5"/><milestone unit="fragment" n="a"/>hobbit-hole, and th<supplied reason="lost"
cert="low">at</supplied> <milestone unit="fragment" n="b"/>means comfort. It had a
  <lb n="6"/><milestone unit="fragment" n="a"/>perfectly round <milestone unit="fragment"
n="b"/><supplied reason="lost">d</supplied>oor like a porthole, painted green,
  <lb n="7"/><milestone unit="fragment" n="a"/>with a shiny <milestone unit="fragment"
n="b"/>yellow brass knob in the exact middle.
  <lb n="8"/><milestone unit="fragment" n="a"/>The doo<unclear>r</unclear> <milestone
unit="fragment" n="b"/>opened on to a tube-shaped hall like a
  <lb n="9"/><milestone unit="fragment" n="a"/>tunne<unclear>l</unclear><supplied
reason="lost" cert="low">:</supplied> <milestone unit="fragment" n="b"/>a very comfortable
tunnel without smoke.
</p>
```

4. Encoding the Originally Inscribed Text

4.1. Alphabetic Characters

– alphabetic characters do not, as a rule, need markup on their own
  – they, including several special character forms, are handled through transliteration alone; see TG §3

4.1.1. Tagging transliterated characters as one *akṣara*

– in certain cases you may need to identify strings of transliterated text as belonging to a single *akṣara* of the original, in order to eliminate ambiguity
  – the transliteration shorthand involving the = (equals) sign, described in TG §3.3.5 and §3.3.8, is recommended for such cases[15]
  – should you prefer to use only XML markup, omit the = sign from your transliteration and wrap all transliterated characters that constitute a single original *akṣara* in the element `<seg type="aksara">`, e.g.
    – `<seg type="aksara">kka</seg>` to encode the Tamil ligature *k=ka* as distinct from both *kka* (with an implicit vowel killer) and *k·ka* (with an explicit vowel killer)

Example 4.1.1.A: character with two vowel marks tagged as a single *akṣara*

```
<seg type="aksara">duā</seg>
```

- this character is probably an engraving mistake for *ddhā*
- the encoding corresponds to the shorthand markup du=ā

  – editorial spaces and hyphens may freely appear between the characters thus enclosed, wherever necessary
  – thus, if a word or compound boundary occurs within such an *akṣara*, encode respectively:
    – `<seg type="aksara">k ka</seg>` instead of *k= ka*
    – `<seg type="aksara">k-ka</seg>` instead of *k=-ka*

4.1.2. Tagging parts of alphabetic characters

– when you need to single out transliterated characters as representing specific parts of an original complex character, you can optionally use the following markup method
– this method, which we shall call *sub-akṣara* markup, has been devised to facilitate the encoding of component-level lacunae (§5.4.5), and is offered as an optional encoding method for unusually arranged complex characters (§4.1.3), but we suggest that you avoid it in all other situations
  – see §4.1.4 about handling character components separated from others by an intervening physical feature, a situation for which *sub-akṣara* markup is not applicable
  – see §5.3.4 about handling reading difficulties concerning character components, a situation for which *sub-akṣara* markup is not normally warranted
– if you choose to tag a specific sub-*akṣara* component, wrap it in `<seg type="component">` and add a `@subtype` attribute with one of the following values:
  – `"body"` for the principal component of a complex character, which may be a single consonant or a conjunct
  – `"superscript"` for any components above the body, such as a superscript *r* or a superscript vowel marker
  – `"subscript"` for any components below the body, such as the subscript consonant of a conjunct or a subscript vowel marker

[15] The markup alternative described here is not expressly limited to the cases discussed in the TG, but is redundant in normal circumstances. If you encounter any other situations where you think its use is warranted, feel free to employ it, but please inform the authors of this Guide and the XML-TEI Data Manager about the details.

– `"prescript"` for any components to the left of the body, generally a vowel marker but also applicable to part of a horizontally composed ligature
– `"postscript"` for any components to the right of the body, generally a vowel marker but also applicable to part of a horizontally composed ligature
– `"consonant"` for exactly one consonant component whose graphic location cannot be determined or is irrelevant
– `"conjunct"` for two or more consonant components belonging to a single *akṣara*, when their locations cannot be determined or is irrelevant
– `"vowel"` for the vocalisation of an *akṣara*, when the location of the vowel marker cannot be determined or is irrelevant

4.1.3. Unusual spatial arrangement in conjuncts

– as our primary objective is to encode texts, the place to record information about unusual character composition is in the commentary to your edition
– however, in order to facilitate future palaeographic research, you may optionally use the above markup for tagging parts of alphabetic characters to specify what part of a complex original character corresponds to any given transliterated character
– when doing so, aim to minimise the complexity of your markup and add tags only to the components that most conspicuously deviate from the expected composition
– however, if you deem that there is any ambiguity regarding *akṣara* boundaries, feel free to wrap transliterated characters and lacunae belonging to a single original character in the element `<seg type="aksara">` as per §4.1.1

Example 4.1.3.A: conjunct with regular *r* instead of superscript

```
<seg type="component" subtype="body">r</seg>ya
```

➢ *rya* is here written with a regular *r* and a subscript *y* instead of a superscript *r* and a regular *y*
➢ the fact that the *r* is tagged as a body component dispenses with the need to explicitly tag the *y* as a subscript component

Example 4.1.3.B: conjunct composed horizontally instead of vertically

```
r<seg type="component" subtype="prescript">g</seg><seg type="component"
subtype="body">gh</seg>a
```

➢ *rggha* is here written in a horizontal composition, with *g* to the left of a regularly positioned *gh*
➢ in this example the second and third consonant components have both been tagged explicitly for their position, though it may arguably be sufficient to tag the prescript *g* in this way

4.1.4. Complex characters split by an intervening feature

– prescript and postscript vowel markers split off from their consonant bodies by an intervening feature shall be handled in transliteration by means of the placeholder characters ⌜ (left ceiling, U+2308) and ⌝ (right ceiling, U+2309), as per TG §3.3.10
  – the intervening features may be line beginnings (§3.2) or space imposed by physical features (§4.3.5) and must be encoded as applicable
  – all the transliterated characters pertaining to an original *akṣara* must be placed on that side of the interruption where the consonant body is located, while the applicable placeholder character must be placed on the other side of the interruption
  – the placeholder characters comprise part of the transliterated text and do not have markup equivalents (in other words, they are not shorthand notation to be replaced by markup)
– split *akṣara*s in themselves need no markup other than the above placeholder characters, but they may be further complicated by the presence of additional markup of the following kinds
  – in all examples here, <> represents an interrupting element of any nature
  – if some of the components involved are **unclear** (§5.3.1)
    – apply this tag to whichever transliterated characters are affected, but not to unaffected ones

- separately apply the tag to the placeholder only if the split-off component is itself affected
- do not include the interruption itself in the markup
- for example:
  - 𑀓𑀳<> 𑀺𑀝 with grey text signifying unclear) would be encoded as `k<unclear>o</unclear><><unclear>]</unclear>`
  - 𑀓𑀳<> 𑀺𑀝 would be encoded as `<unclear>ko</unclear><>]`
  - 𑀓𑀺<>𑀳𑀝 would be encoded as `<unclear>[</unclear><>k<unclear>o</unclear>`

- if the reading of some components is **ambiguous** (§5.3.3), in the interest of minimising complexity consider whether encoding the *akṣara* as unclear would be sufficient for a reader familiar with the script to deduce the possible alternatives
- if you deem that encoding an ambiguity is essential
  - do so for the transliterated characters concerned
  - add an unclear(!) tag to the placeholder if the split-off component is affected
  - do not include the placeholder in the markup for the ambiguity, and do not include the interruption itself in any markup
- for example:
  - 𑀓𑀳<> 𑀺𑀝 where the unclear strokes after the interruption may be the character 𑀭 *ra* instead of the vowel marker 𑀺𑀝 (called *kāl*) may be encoded as `k<choice><unclear>o</unclear><unclear>era</unclear></choice><><unclear>]</unclear>`
    - note that in this case the second option of the `<choice>` element produces the text "*kera*<>]", where the placeholder sign must be understood to mean that the entire preceding *akṣara* (i.e. *ra*), rather than just one component of it, is located after the interruption[16]

- if an adjacent **lacuna** has obliterated a split-off component which you **supply** (§5.5)
  - mark up the affected vowel
    - as supplied if it consists only of the supplied split-off component
    - as unclear if it consists of an extant component *and* a supplied split-off component
  - mark up the placeholder as supplied
  - do not include the interruption itself in any markup
  - for example
    - 𑀓𑀳<> 𑀺𑀝 (with grey text for the restored lacuna) would be encoded as `k<unclear>o</unclear><><supplied reason="lost">]</supplied>`
    - 𑀓𑀺<>𑀳 would be encoded as `<supplied reason="lost">[</supplied><>k<supplied reason="lost">e</supplied>`
- if an adjacent **lacuna** may have obliterated a split-off component but you **do not supply** this even tentatively
  - simply mark up the *extant* vowel component as unclear and leave it to the proficient reader to deduce the fact that an additional component may have been destroyed in the lacuna

4.2. Non-alphabetic Characters

4.2.1. Overview

- we will use the element `<g>` (for "glyph" or "gaiji"[17]) in the encoding of all characters other than alphabetic ones and decimal digits
  - the use of this element indicates that no perfect equivalent to the original character is available in our transliteration system

[16] We must resort to this compromise because the internal logic of EpiDoc does not permit `<space>` or `<lb>` elements inside `<unclear>`, so it is not possible to encode alternative readings which include an interruption, even if the location of that interruption with respect to the text is different in the alternatives.

[17] https://en.wiktionary.org/wiki/gaiji

  - the characters prescribed in our Transliteration Guide are deemed to be perfect equivalents to original alphabetic characters and decimal digits and therefore require no encoding as glyphs
- the element `<g>` shall be used in two different ways:
  - as a text-containing element `<g>text</g>` to **wrap** one or more transliteration characters that convey information about the function of the glyph, namely (1) **numerals** and the dedicated characters for (2) **space fillers** and (3) **punctuation marks**
  - as an empty element `<g/>` to **represent** a glyph about whose function our encoding makes no assertions, i.e. all glyphs not covered by these three specific cases, such as auspicious symbols at the beginning or end of an inscription
- in addition to serving as an indication of a character for which no exact transliteration equivalent exists, `<g>` shall always carry the attribute `@type` to encode further details of the original glyph, as discussed in the subsections below
- this twofold encoding (involving a `<g>` element in all cases and text enclosed within that element in specific cases) serves to represent the fact that a glyph of a particular shape may be used in more than one function across the corpus, a subcorpus, or even within a single inscription, so that
  - glyphs that are definitely numeral characters (other than decimal digits) are transliterated into Arabic numerals wrapped in `<g type="numeral">` as described in §4.2.2
  - glyphs that are definitely punctuation marks (§4.2.4) are transliterated as the abstract punctuation character . and wrapped in `<g>` with an appropriate `@type`
  - glyphs that are definitely space fillers (§4.2.5) are transliterated as the dedicated transliteration character § and wrapped in `<g>` with an appropriate `@type`
  - non-alphabetic glyphs that do not clearly fall into any of the above categories (§4.2.6) are not transliterated with any character, but represented by the empty element `<g/>` with an appropriate `@type`

4.2.2. Numeral symbols other than decimal digits

- keep in mind that all numbers originally recorded in numeral symbols (including those in decimal digits) must also be encoded for their value as described under §7.1
  - the encoding introduced here pertains to specific numeral characters used in the original script, that have no exact equivalent in our transliteration system, while the encoding of value introduced in §7.1 applies to all numbers written in numeral signs and adds semantic information to our encoding scheme
- TG §4.1 and its subsections provide a shorthand notation to distinguish numeral signs transliterated in any way other than by a single Western numeral or vulgar fraction sign
  - namely
    - two or more Arabic digits transliterating a single glyph in the original (e.g. “10+” for the Brahmi glyph ∝ meaning “10”)
    - one or more iterations of a Latin uppercase I transliterating Cambodian numeral notation involving vertical bars (e.g. “III+” for a triple vertical bar meaning “3”)
    - fractions other than halves, thirds and fourths (e.g. “1/8+” for an original character denoting “one eighth”)
  - this shorthand notation will be automatically converted to the XML markup presented below
    - however, it is recommended that you use only the XML markup when encoding a new edition in XML, as the shorthand is mainly intended to facilitate the conversion of e-texts prepared earlier into DHARMA-compliant XML encoding
    - never combine the shorthand markup involving a + sign with XML markup for the same purpose
- the transliteration string corresponding to a single original numeral must be
- wrapped in the XML element `<g type="numeral">`, e.g.
  - `<g type="numeral">200</g>` corresponds to the shorthand 200+
  - `<g type="numeral">100</g> <g type="numeral">20</g> 3` corresponds to the shorthand 100+20+3
    - note that the transliterated 3 is not wrapped in `<g>`, because it is a single Arabic digit

- `<g type="numeral">1000</g> 8 <g type="numeral">100</g> 3 <g type="numeral">10</g>` corresponds to the shorthand 1000+ 8 100+ 3 10+
  - 8 and 3 are not wrapped in `<g>`, because they are single Arabic digits
- `<g type="numeral">I</g>` corresponds to the shorthand I+ (for a vertical bar denoting "1" in a Cambodian inscription)
  - note that even though "I" in transliteration is a single character, the <g> tag is necessary in this case to mark up this character as non-alphabetic[18]
- `<g type="numeral">1/8</g>` corresponds to the shorthand 1/8+ (for "one eighth" written as a single original character)

4.2.3. Symbol tokens

- as indicated in §4.2.1 above, non-numeric symbols encoded with a `<g>` element (i.e. punctuation marks, space fillers and miscellaneous symbols, covered separately in the following subsections) must be encoded with a variety of values for `@type`
  - the value of used in each case shall be a simple description of the symbol's visual appearance (or in a limited number of cases its traditional name), hereafter referred to as a *token*
  - the token must contain no spaces, but it may contain any combination of letters and numbers
- at this stage of our project there is no constraint on the permitted symbol tokens
  - at a later stage, we intend to harvest tokens that have been used and utilise them as a starting point for a controlled vocabulary for symbol description, involving a limited number of `@type` values and a larger number of permitted `@subtype` values for each `@type`
- however, for the sake of making that future work easier, and to facilitate the development of display solutions for symbols, it is strongly recommended that you follow certain basic constraints in naming your symbols:
  - use a **simple character set** consisting only of the letters of the English alphabet and numerals, i.e. avoid symbol characters and letters with diacritic marks
  - use a **hierarchical approach**, in which tokens may be
    - simple, consisting of a single term that identifies a broad category of shapes ("genus"), e.g.
      - `"circle"`, `"dash"`, `"flower"`, etc.
    - complex, beginning with a term for a species as above, and followed by one or more qualifications of a subcategory ("genus"), using camelCase (i.e. starting each subsequent word with an uppercase initial) for segmentation, e.g.
      - `"circleSmall"`, `"circleCross"`, `"circleSmallHigh"`, etc.
      - `"dashHook"`, `"dashConcave"`, `"dashHookHigh"`, etc.
    - it is, however, recommended that you resist the temptation of creating highly elaborate complex tokens, since our ultimate aim is to devise a versatile but limited vocabulary for symbol classification
      - keep in mind that symbols can be described in detail in the Hand Description (§11.2.1), and doing so is strongly recommended for all symbols whose shape will not be self-evident to a reader familiar with the subcorpus
  - while there is no such thing as an incorrect symbol token, all of us should from this early stage onward try to **avoid excessive diversity** in the naming of symbol shapes
    - for this purpose, we have created an online Supplement to the EGD on Symbol Taxonomy[19] in which we have entered some of the symbols we have encountered in our work so far, with the recommended tokens for each
    - all encoders are requested to refer to that list before creating a token for a symbol
    - all encoders are encouraged to contribute to that document by
      - inserting clippings of symbols they have encoded with a token already featured in the list

[18] Notwithstanding the fact that for a so-called 'independent vowel' *i* you would normally use *qi* rather than *I* in Cambodian inscriptions (TG §3.3.4), the use of *I* for numerals without this explicit markup would create an inconsistency in the corpus as a whole.

[19] https://docs.google.com/document/d/1glfyQnFqPrbVOYzegfjKIOVrc-vMgznEQ1iNsFf7DE8/edit?usp=sharing

  - inserting new rows in the list with clippings of new symbols and the tokens they have come up with for those symbols

4.2.4. Punctuation marks

- as stated in TG §4.2.1, the term "punctuation mark" is used within this Guide in a sense restricted to symbols
  - which are (or are derivations of) simple non-figural shapes
  - and which are employed in the original for syntactic or metrical segmentation into relatively small units, similar in function to a modern comma, full stop, question mark, exclamation mark, colon or semicolon
  - generally excluding figural and ornamental signs as well as signs used to mark the end or beginning of an entire text or a major section of text
- this subsection concerns the encoding of punctuation marks as described above; for symbols that do not qualify as punctuation marks, use the encoding described in §4.2.6 below
  - we feel that this distinction in encoding is useful in many cases for distinguishing symbols definitely used for the purpose of punctuation from symbols used for a different or a less straightforward purpose
  - however, the above definition is not and cannot be entirely objective, and in some cases it will not be possible to decide whether a symbol is a "punctuation mark" in this sense, or a "miscellaneous symbol"
  - we recommend that you choose the encoding for miscellaneous symbols whenever in doubt
  - also keep in mind that encoding a miscellaneous symbol instead of a punctuation mark or vice versa is not an error and will have little ultimate impact on the quality of our corpus
- as also explained in TG §4.2.1, punctuation marks are to be transliterated as the abstract punctuation character . (full stop, period), and their shape must normally be encoded in the `<g>` wrapper of this character
  - this `<g>` element must mandatorily take the attribute `@type`, with a value as described under §4.2.3 above
    - since punctuation marks, despite their diversity across the entirety of our corpus, are generally variations on a small number of basic shapes, it is strongly recommended that your tokens (or the first component of your complex tokens) chosen for describing punctuation marks follow the nomenclature suggested in the Symbol Taxonomy[20]
- thus, for the purposes of our encoding, the only difference between the encoding of a miscellaneous symbol and that of a punctuation mark is that while a miscellaneous symbol is encoded as an empty `<g/>` element representing the symbol, a punctuation mark is encoded as a `<g>` element wrapping a . which serves in our system as an abstract punctuation character (i.e., it implies nothing about the shape of the punctuation mark in your document)
- the primary purpose of adding . within `<g>` for punctuation marks is to make it explicit on the lowest level (that of the text itself) that we consider certain characters to be punctuation marks[21]
- in addition, the use of the dedicated transliteration character for abstract punctuation marks permits us, when necessary, to use this character without a `<g>` wrapper, for representing a punctuation mark without any assertion as to its shape, exclusively in the following situations:
  - when supplying punctuation for the purpose of semantic segmentation
    - in this case, use a . character marked up as omitted in the original
    - see §6.3.6 for further details
  - when encoding a text from a previous edition, without access to the original or a surrogate, if that edition does not describe the appearance of original punctuation marks

[20] https://docs.google.com/document/d/1glfyQnFqPrbVOYzegfjKIOVrc-vMgznEQ1iNsFf7DE8/edit?usp=sharing

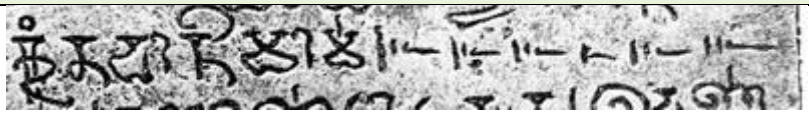
[21] TEI allows the semantic tagging of characters as punctuation marks. We may at a later stage decide to add such tags automatically, but at present we see no advantage to doing so.

– in this case, use a single . to represent a lower-level or generic punctuation mark (e.g. a full stop or *daṇḍa*) used in the previous edition,
– or a double .. to represent a higher-level punctuation mark (e.g. a double *daṇḍa*) used in the previous edition, if that edition employs two levels of punctuation

4.2.5. Space filler signs

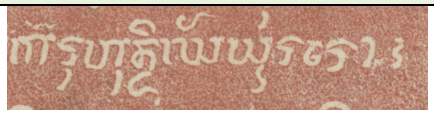
– symbols whose function is clearly and unambiguously to fill up space in a line to the binding-hole or margin are, as per TG §4.2.2, transliterated using the § sign
– in your XML markup, § characters must be wrapped in a `<g>` element with the mandatory attribute `@type`, with a value as described under §4.2.3 above
– thus, a symbol used as a space filler is distinguished from an identical-looking symbol used in a different (or unidentified) function by the presence of the § character within `<g>`
– multiple iterations of an identical space filler shall be wrapped in a single `<g>` tag, so that the number of § characters within that tag corresponds to the number of symbols in the original

Example 4.2.5.A: space fillers from South India



```
kṛtavān imām· <g type="gomutraFinal">§§§§§§</g>
```

Example 4.2.5.B: space filler from Southeast Asia



```
karuhun di:rghăyūrăro<g type="squiggleVertical">§</g>
```

4.2.6. Miscellaneous symbols

– this subsection applies to non-alphanumeric symbols which do not clearly fall into any of the following categories:
– premodern editorial marks, which are not encoded as textual content, but as per §4.5
– punctuation marks as defined in §4.2.4
– space fillers as defined in §4.2.5
– in our XML files, miscellaneous symbols must be represented by the empty element `<g/>` with the mandatory attribute `@type`, with a value as described under §4.2.3 above

4.2.7. Alphanumeric characters used for a different function

– it occasionally happens that alphabetic or numeric characters are used in a function other than their regular value
– when an **alphabetic character** functions as a symbol (such as the character *cha* or *chaḥ* used in some regions and periods as a closing symbol)
– do not use any markup to encode its function, but simply transliterate the character normally, separated by a space from any adjacent text
– when a **numeral sign** functions as a symbol (such as the glyph normally meaning 1, occasionally used as an auspicious opening mark)
– do not use any markup to encode its function, but also do not apply the semantic markup for numerals described in §7.1
– when a **numeral sign** functions as an alphabetic character (such as the numeral 2 used in Old Sundanese to represent the phonemes /ro/)
– do transliterate the character as the numeral, but do not apply the semantic markup for numerals described in §7.1

4.3. Space

4.3.1. Generic markup for original space

– if an inscription contains blank **space**, this should generally be encoded using the empty element `<space/>`, which can take the attributes `@unit` and `@quantity` to describe the size of the space, and `@type` for classification
  – the following subsections describe when to use which attribute, and with what values
  – the encoded size of spaces is by default always understood to be approximate
– the element `<space/>` should normally be separated from surrounding text by editorial spaces in your file, but if a `<space/>` occurs within a word of the text, no spaces should be added around the element
  – see §8.1.2 for more details

4.3.2. Space for semantic segmentation

– this subsection is about spaces employed within lines by the creator of an inscription, with the presumed purpose of highlighting some aspect of semantic structure, such as spacing
  – between words
  – after stanzas or lines
  – at a transition from verse to prose or vice versa
  – at points where the topic changes markedly, for instance
    – after an initial salutation or auspicious phrase
    – before a colophon
– regular TEI practice[22] is not to use `<space/>` for such spaces, but since our texts do not normally space words, we consider these to be "significant spaces" when they do occur and encode them as follows
– **small spaces** (from barely noticeable to less than two average character widths in extent) may be encoded using the `<space/>` element without any attributes
  – as per TG 4.3, you can use the _ character as shorthand for `<space/>` without any attributes; this will be automatically converted to markup
  – the encoding of small spaces is optional and should be decided on a case by case basis, with considerations such as the following:
    – no space should be encoded for widely spaced characters within a word, or between words if spaces of similar size also occur within a word
      – however, segments of text written in conspicuously widely spaced characters may be marked up as per §7.5.5
    – interword spacing used with fair consistency throughout an inscription may be mentioned in the metadata or commentary rather than being encoded at every instance
    – small spaces used before and/or after punctuation marks, numerals and other symbols do not normally need to be encoded
    – spaces used in lieu of punctuation (e.g. at the end of stanzas, verse lines or semantic units) should generally be encoded even if small
– **large spaces** (two or more characters wide) must always be encoded, using the element `<space/>` with the following mandatory attributes
  – `@quantity`, whose value shall be the width of the space given as the number of characters that could fit into it (i.e., the number of widths of an average *akṣara*)
  – `@unit`, with the value `"character"`
– if an encoded semantic space is at a boundary between XML elements (e.g. between stanzas, semantic paragraphs, etc.), place the `<space/>` element within the structural container to which it can be allocated more logically
  – space used instead of punctuation should generally be encoded at the end of the container which it separates from the next

[22] https://www.tei-c.org/release/doc/tei-p5-doc/en/html/PH.html#PHSP

4.3.3. Space left blank for subsequent filling

– this section is about areas that were left blank when the rest of the inscription was engraved, with the intent to be filled later on, e.g. with a name or a date
– for such spaces, called *vacat* in the western scholarly tradition, add the attribute `@type` with the value `"vacat"` to the element
  – always record the size of such spaces as described for large spaces above, even if they are smaller than two character widths
  – e.g. `<space type="vacat" quantity="3" unit="character"/>`
– for areas first left blank and subsequently partially filled (with some blank space remaining),
  – if there is any uncertainty about the presence of an addition or its exact extent, mark up only the remaining blank space in this way
  – if you are certain about both the existence and the size of the text filled in later, mark up a vacat for the entire length of the original space, and mark up the added text as a premodern inline addition (§4.5.2) either after or before the space

4.3.4. Space for visual layout

– space left blank in a text for the sake of visual appearance should not, as a rule, be marked up as space
  – instead, the following options are available
– for inline layout spaces:
  – optionally encode specially aligned lines (as per §7.5.2) when spaces appear
    – at the end of a line that begins flush with the left margin
    – at the beginning of a line that ends flush with the right margin
    – at the beginning and end of a line that is centred between the two margins
    – between all or most words or characters in a line that is justified to both margins
  – optionally encode gridlike partitions (as per §3.6) when spaces appear between segments of each line (such as stanza quarters) to divide the text into quasi-columns where each line is intended to be read across several of these
– for blank space between lines:
  – encode nothing, i.e. create no `<lb/>` elements for empty lines and insert no `<space/>` elements to represent them
  – the regular line spacing and any deviations from it should be described for human readers in the layout description
– for blank pages in copper plates:
  – encode `<pb/>` elements for the blank pages as per §3.5.2, but insert no `<space/>` elements to represent their content

4.3.5. Spaces imposed by physical necessity

– this subsection is in general about cases where the engraver was prevented from writing on a certain area of the support
  – specific features causing such prevention are discussed in the following subsections
– encoding such interruptions as “significant space” is helpful because their presence may be the cause of non-standard sandhi and scribal errors
– the encoding of such spaces is optional, especially when encoding a printed edition without access to the original or a surrogate
  – however, if you do choose to encode any such space in an edition, then do so consistently throughout the edition, i.e.
    – encode all spaces of the same type (as per the subsections below)
    – while remaining free to ignore imposed spaces of a different type in the same edition, and to ignore imposed spaces of any type in other editions even within the same subcorpus
– when encoding spaces imposed by physical necessity, distinguish these from functional spaces by adding the attribute `@type` to `<space/>`, with a value according to one of the following subsections

– should you encounter a space that you feel was imposed on the engraver by a physical feature, yet none of the types listed below classify it correctly, choose another word for the type and contact the authors and the XML-TEI Data Manager to discuss adding it to the Guide
– note that unlike other encoded spaces, this kind of space will frequently occur within a word
– in this case, do not add space characters around the `<space/>` element
– see §8.1.2 for further details
– whether or not you encode them as spaces, conspicuous features of the support may be mentioned in your layout description (global aspects) or your commentary (individual instances)

4.3.6. Binding holes in copper plates

– when the binding hole in a copper plate affects the text of a line, this may be encoded as `<space type="binding-hole"/>` at the locus of the hole
– use the above encoding regardless of the size of the space caused by the hole, i.e. never use `@quantity` and `@unit`
– keep in mind that it is not the presence of a binding hole that we encode here, but the fact that such a hole has obliged the engraver to skip horizontally, therefore
– **do not** encode a space for a hole that is fully outside a margin line (in the area surrounding the text field), as in Example 4.3.6.A/1
– **do not** encode a space for a hole that lies within the text field, but between lines (even if lines above/below the hole bend, or if characters in those lines are distorted in order to accommodate the hole), as in Example 4.3.6.A/2
– **optionally** encode a space for a hole that is on or within the margin line, causing one or more text line to begin with an indent, as in Example 4.3.6.A/3
– **preferably** encode a space for a hole that is fully within the text field, interrupting one or more text lines, as in Example 4.3.6.A/4
– keep in mind that binding holes, whether encoded individually or not, must be described in your layout description

Example 4.3.6.A: positions of a binding hole relative to text

1 2 3 4

– when a binding hole affects more than one line in this way, encode such a space for every affected line
– there is no need to explicitly encode the fact that a single hole interrupts more than one line

4.3.7. Surface defects

– when a **surface defect in the support** has prevented the engraver from writing on a certain area, you may use `@type` with the value `"defect"`
– without further attributes if they are less than two average character widths in breadth, e.g. `<space type="defect"/>`
– or with `@quantity` and `@unit` (as described under §4.3.2 above) when they are two or more characters wide, e.g. `<space type="defect" quantity="3" unit="character"/>`
– the significance of any such spaces as relevant to the understanding of the text and its creation may be discussed in an apparatus note
– when a defect affects more than one line in this way, encode such a space for every affected line
– there is no need to explicitly encode the fact that a single defect interrupts more than one line

4.3.8. Spaces imposed by other glyphs

– if a space was left blank in a line because (part of) another (pre-existing or pre-conceived) character, belonging to another line, was already occupying that space, then you may use `@type` with one of the following values
  – `"descender"` where a large **character hanging down** from the previous line encroaches on the current line
  – `"ascender"` where a large **character popping up** from the following line encroaches on the current line
– spaces for ascenders and descenders should be encoded without any attributes other than `@type`, i.e. their extent should not be explicitly encoded
  – e.g. `<space type="ascender"/>`
– the significance of any such spaces as relevant to the understanding of the text and its creation may be discussed in an apparatus note

4.4. Scribal Hands

– in epigraphic parlance, a "hand" means a particular combination of writing features, often indicative of one scribe taking over the work of another
– **if your inscription is written in several**, clearly distinguishable **hands** (as opposed to different scripts or styles, covered in §7.5.4 and §7.5.5), first of all you need to create short descriptions of each hand in the `<handDesc>` element of the TEI header (§11.2.1)
– within your edition, create the empty element `<handShift/>` at each point where the hand changes, including the beginning of the inscription where the first hand appears
  – in this element, include the mandatory attribute `@new`,[23] whose value shall be the XML identifier associated in the header with the hand in question, prefixed with a # mark
  – e.g. `<handShift new="#Pallava00001_hand1"/>`

4.5. Premodern Editorial Intervention

– this section covers deliberate alteration of the inscribed text carried out in premodern times, and thus does not apply to
  – purposeful or accidental effacement of the entire text or random parts of it
  – the engraving of a palimpsest over a pre-existing inscription
– most premodern editorial corrections presumably took place shortly after the full text was first engraved, though some may have happened at a later time
  – the precise deletion of clearly circumscribed sections of an inscription (e.g. of names) is included in the scope of this section

4.5.1. Premodern deletion

– text deleted (without adding a corresponding correction) in ancient or medieval time, e.g. by chiselling a stone surface, hammering the copper flat, or marking text for deletion by dotting or other conventional signs, should be wrapped in the element `<del>`
  – the contents of this element may include additional markup, e.g. for reading difficulties (§5.3)
  – `<del>` should be used only if you are certain that you are facing purposeful deletion, not damage to the support
– in our project, this element will by default be understood to represent text rendered illegible through **erasure** by means of chiselling, rubbing or hammering
– for **cancellation indicated by marks** (even if accompanied by partial erasure), add the attribute `@rend` to classify the method of deletion, with one of the following values
  – `"strikeout"` for text struck through or slashed

[23] The reason why the attribute is named "new" is presumably the fact that it identifies the new hand, i.e. the one taking over at the shift. TEI guidelines note that this attribute may be renamed in a subsequent major release.

  - `"ui"` for the combined application of vowel markers *u* and *i* to characters to be deleted
  - `"other"` for any deletion marker other than those listed above
  - further details about the form and placement of deletion marks used in your inscription may be described in your metadata
- keep in mind that the cancellation of a previously inscribed explicit vowel mark restores the inherent *a* of a consonant *akṣara* in the scripts we work with, so such deletion must be encoded as a correction (§4.5.3) of the explicit vowel into *a*

4.5.2. Premodern insertion

- for characters inserted into an inscription in premodern time, create the added text at the text location where it was meant to be read (regardless of its actual location in the inscription), and wrap it in the element `<add>` with the following attributes:
  - mandatorily, `@place`, with one of the following values:
    - `"inline"` when inscribed within the same line in the immediate vicinity of the locus
      - e.g. a character inserted between two pre-engraved characters or text engraved over a space previously left blank (see also §4.3.3)
    - `"below"` for an interlinear addition below the locus
    - `"above"` for an interlinear addition above the locus
    - `"top"` for an addition in the top margin
    - `"bottom"` for an addition in the bottom margin
    - `"left"` for an addition in the left margin
    - `"right"` for an addition in the right margin
  - when applicable, `@rend` with the value `"mark"` to encode the involvement of a premodern editorial mark (Sanskrit *kākapada*), as illustrated in Example 4.5.2.B
    - this encoding method shall apply regardless of where such an editorial mark appears (at the locus of insertion, next to the inserted text, or at both places)
    - the shape and placement of the marks shall be described in an apparatus note
- the inserted text may include additional markup
- it may sometimes be impossible to determine the intended locus of a piece of interpolated or marginal text; in this case
  - encode the addition at a likely place or, if one cannot be found, at any locus of your choice such as the beginning or end of a line, page or the entire inscription
  - and describe the situation in your commentary
  - alternatively, you may opt to encode the added text as an additional line of the principal text (§3.3.4)

Example 4.5.2.A: premodern interlinear insertion

➢ an originally inscribed word *dīnāram* was corrected to *dīnāra-dvayam* by adding *dvaya* between lines below this word

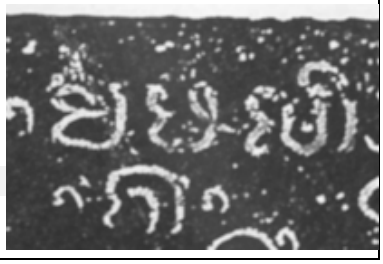
```
dīnāra<add place="below">-dvaya</add>m
```

Example 4.5.2.B: premodern insertion with an editorial mark

➢ an originally inscribed *maphalā* was corrected into *makaphalā* by adding *ka* between lines below this word (see the illustration)

➢ the scribal marks on the left and right of the added *ka* look like punctuation marks, but the scribe's intention was **not** to write *ma,ka,phalā* so we should not transliterate these marks as punctuation signs

```
ma<add place="below" rend="mark">ka</add>phalā
```



4.5.3. Premodern correction

- when a correction is written over previously engraved text, which was rendered completely illegible in the process, encode the correction as an insertion with a special value of `@place`:
  - `<add place="overstrike">abc</add>`

– when any of the pre-correction text can be read (or restored), correction must be represented as a combination of deletion and addition, wrapped in the element `<subst>` to show that one is meant to be substituted by the other
  – tag the deleted text with `<del>`, using the attribute `@rend`
    – with one of the values listed under premodern deletion above; or
    – with the value `"corrected"`, if the text to be replaced was neither erased, nor marked for cancellation (i.e. it is either overwritten with the post-correction text or left in place without any apparent alteration)
  – tag the added text with `<add>`, using the attribute `@place`
    – with one of the values listed under premodern addition above; or
    – with the value `"overstrike"`, if the replacement text is inscribed over the pre-correction text (rather than at some other position)
– bear in mind that the cancellation of an explicit vowel mark restores the inherent *a* of a consonant *akṣara* in the scripts we work with, so even though no act of correction separate from the act of deletion is involved, the result is in fact a correction of the reading
  – therefore such deletion must be encoded as a correction of the explicit vowel into *a*, with the corrected text struck over the pre-correction text (see Example 4.5.3.C below)

Example 4.5.3.A: premodern correction by overwriting

➢ an originally inscribed *droṇavāpam* was corrected to *kulyavāpam* by inscribing *kulya* over *droṇa*

```
<subst><del rend="corrected">droṇa</del><add place="overstrike">kulya</add></subst>vāpam
```

Example 4.5.3.B: premodern correction with an editorial mark

➢ an originally inscribed *droṇavāpam* was corrected to *kulyavāpam* by striking out *droṇa*, adding a *kākapada* at this spot, and adding *kulya* in the bottom margin

```
<subst><del rend="strikeout">droṇa</del><add place="bottom" rend="mark">kulya</add></subst>vāpam
```

Example 4.5.3.C: premodern correction by striking out a component

➢ an originally inscribed *prisaṅgi* was corrected to *prasaṅgi* by striking out the superfluous *i* marker

➢ the markup must include the correction encoded as `"overstrike"` even though no explicit *a* was engraved

```
pr<subst><del rend="corrected">i</del><add
place="overstrike">a</add></subst>saṅgi
```

Example 4.5.3.D: premodern correction with an editorial mark

➢ an originally inscribed *ri lata* was corrected to *ri tala* in a manuscript by adding an editorial mark

```
ri<subst><del rend="corrected">lata</del><add
place="overstrike" rend="mark">tala</add></subst>
```

➢ although the intended correction is not explicitly written anywhere, the intent is clear to a competent editor, so we encode the facts that can be encoded in the above scheme:

➢ that the pre-correction text is not explicitly deleted, but overruled by the correction (`@rend="corrected"` on `<del>`)

➢ that the post-correction text is right there and not somewhere else on the support (`@place="overstrike"` on `<add>`)

➢ that an editorial mark is present (`@rend="mark"` on `<add>`)

5. Physical Condition and Legibility

5.1. Overview

The criteria for determining the markup appropriate for specific problems may not be straightforward, so please read through this entire section to familiarise yourself with the options, then, whenever in doubt, return to this introduction for a guided decision. The factors to consider are as follows:

1. What is the condition of the support at that particular spot?
   a. wholly lost; or extant, but so damaged that all vestiges of writing are obliterated: `<gap reason="lost">`, see §5.4.2
   b. extant, with damage ranging from minor to extensive: go to point 2
   c. extant and undamaged: go to point 3
2. To what extent does the damage hinder the reading of the text?
   a. Though vestiges of text are discernible, they are too scant to favour one contextually possible restoration over another: `<gap reason="illegible">`, see §5.4.2
   b. Due to damage, characters cannot be identified with certainty without relying on their context, but given the context and your expertise, you can at least make an educated guess about them: `<unclear>`, go to point 4 for further details
   c. Despite some damage, all characters can be identified with certainty even if their context is disregarded: no markup (or optional `<damage>`, see §5.2)
3. Does an unusual, awkward or incompetent execution of the glyphs hinder the reading?
   a. Due to their form, characters cannot be identified with certainty without relying on their context: `<unclear reason="eccentric_ductus">`, go to point 4 for further details
   b. All characters can be identified with certainty even if their context is disregarded (though some irregularity of execution may be present): no markup
4. Informed by the context and your expertise, how confidently can you read/restore the affected characters?
   a. With complete confidence and a conviction that even if something other than your reading was intended, the difference is trivial: omit markup at your own discretion and simply treat the text as clearly legible.
   b. Quite confidently, but admitting for honesty's sake that there is a small chance of a non-trivial alternative reading being possible: `<unclear>`, see §5.3.1
   c. You recognise a small number of alternatives as being possible with fairly equal chance: ambiguity marked up as `<choice>` with `<unclear>`, see §5.3.3
   d. Tentatively, admitting a fair chance that a non-trivial alternative reading is possible `<unclear cert="low">`, see §5.3.2

Another way to look at the options is summarised by the following table:[24]

Table 1. Overview of legibility issues

Confidence in reading/restoration	Status of text		
	lost	illegible	doubtful
absolute	`<supplied reason="lost">`	`<unclear>`	no markup
reasonable			`<unclear>`
tentative	`<supplied reason="lost" cert="low">`	`<unclear cert="low">`	`<unclear cert="low">`
nil	`<gap reason="lost">`	`<gap reason="illegible">`	NA

– **status:**
  – **lost** = the support is gone or at least its surface layer is completely destroyed
  – **illegible** = the support is extant and there are vestiges of writing on its surface, but they cannot be read with any degree of confidence
  – **doubtful** = writing is extant and at least tentatively legible, but if the character(s) were taken out of their context, their reading would be equivocal (either because they are damaged or because they are unusually formed)
– **confidence**: as per 4a,b,c,d above, plus
  – **nil** = the text cannot be read or restored with any confidence

5.2. Damage Not Affecting Legibility

– when the physical features of the support or damage to its surface do not affect the reading of the inscription, such features **need not be marked up**
  – extensive patches of weathering or loss (which may include lacunae and reading difficulties intermingled with clearly legible text) may be described for human readers in your metadata
  – spaces left blank in an inscription because of pre-existing defects or features of the surface shall be encoded as per §4.3.5
– however, should you deem it essential to explicitly encode a stretch of text as damaged, wrap the affected stretch in the element `<damage>`
  – the contents of this element may include markup for lacunae and reading difficulties intermingled with clearly legible text, paying attention to the following:
    – avoid overlapping with other tags by splitting `<damage>` into several segments as necessary
    – regardless of the `<damage>` tag, any reading difficulties and lacunae within a spot of damage must always be marked up as described below

[24] As the table makes clear, EpiDoc does not allow the use of `<supplied reason="illegible">`, which would be expected in the middle column. The rationale behind this is that if any vestiges remain, and these can be reconstructed on the basis of context, then they meet the definition of `<unclear>` (see §5.3.1) and ought to be marked up as such. While one could argue for a distinction between "conjecturally restored text not explicitly ruled out by vestiges" and "text restored on the basis of vestiges," our encoding practice shall follow established EpiDoc convention. Note also that the tradition of epigraphic editions in India is actually quite in line with the EpiDoc approach. Even in the *Corpus inscriptionum indicarum*, Fleet (1888,194) uses the same editorial markup for "letters which are much damaged and nearly illegible in the original, or which, being wholly illegible, can be supplied with certainty."

5.3. Doubtful Readings

5.3.1. The EpiDoc element `<unclear>`

– the term “unclear”, represented by the XML element `<unclear>`, stands in EpiDoc for any character “of which at least traces survive, but not adequately to identify the letter unambiguously outside of its context”,[25] and therefore includes not only situations where a reading is tentative, but also
  – where the text is read in context with absolute confidence and would only be doubtful in isolation
  – where the vestiges are entirely illegible, but can be restored from the context
– while many of us tend to use editorial markup (such as brackets) only to indicate “I’m not sure this is really what the inscription said”, `<unclear>` in EpiDoc, as per the definition quoted above, means “this bit of text could conceivably be something else if the context was not there to help”
  – `<unclear>` would, by that logic, be used more extensively than indications of editorial uncertainty in most of our editions
  – given, however, that many of the inscriptions we work with are considerably damaged, it is desirable to avoid cluttering the edition with `<unclear>` markup and thereby distracting attention from spots where damage (or form) casts genuine doubt on a reading
  – therefore, at your own discretion, **ignore trivial doubts** if the text can be read in its given context with such confidence that there is no need to leave open the possibility of any alternative
– when marking up text as unclear, you must keep in mind that the EpiDoc schema permits only text and the XML element `<g>` within `<unclear>`
  – therefore, if a stretch of text you wish to mark up as unclear incorporates or overlaps with another stretch that needs different markup, you will need to split the tagged stretches of text accordingly
  – see §8.2.5 for details and examples
– to mark up **damaged text legible in context with reasonable confidence**, while allowing a slight chance that a different reading might be possible
  – use the element `<unclear>` without any attributes
– when the confidence of a reading is affected not by damage, but by the **unusual, awkward or incomplete execution of a glyph** by its original engraver
  – add the attribute `@reason` with the value `"eccentric_ductus"`, e.g. e.g. `<unclear reason="eccentric_ductus">jñ</unclear>āna`
  – when damage and eccentric ductus are (or may be) simultaneously present, use `<unclear>` with or without this attribute depending on what you consider to be the primary reason for the lack of clarity
  – when in doubt, prefer `<unclear>` without this attribute

5.3.2. Tentative readings

– if some of the text is only **tentatively legible** even in its context, add the attribute `@cert` with the value `"low"` to the element `<unclear>`
  – e.g. `mahā<unclear cert="low">puruṣa</unclear>`
  – this attribute may be used in conjunction with `@reason` when needed
  – no additional degrees of confidence shall be represented in markup, so `"low"` here may stand for anything between “not quite fully confident” to “desperate conjecture”
    – however, it is preferable to save desperate conjectures for your commentary or apparatus, and within the edition, only encode readings in which you have some confidence
– note that while a low certainty expressed by this attribute is often a consequence of extensive damage, the **degree of legibility** is not in direct correlation to the necessity of adding this attribute, thus:
  – even very badly damaged characters may be marked up with plain `<unclear>` if they can be confidently supplied in the context; whereas

[25] http://www.stoa.org/epidoc/gl/latest/trans-ambiguous.html.

- even characters that are only slightly damaged or have been executed with only slight awkwardness may need to be marked up as `<unclear cert="low">` if the context permits a variety of plausible alternative readings, chiefly in unintelligible or only partly understood contexts (e.g. names, words foreign to the language of the inscription, or in case of extensive damage to the context)

5.3.3. Ambiguous characters

- if a damaged or malformed character affords **a limited number of alternative interpretations** and the context gives no clear indication of which is correct, each alternative must be listed in individual `<unclear>` tags, wrapping the list in the `<choice>` element to show that only one of these goes in the given locus
  - e.g. `g<choice><unclear>r̥</unclear><unclear>ra</unclear></choice>ha`
- alternative readings may affect the editorial spacing of the text differently; see §8.1.2 for some guidance in such cases
- ambiguities involving **more than two alternatives** may be marked up simply by adding further `<unclear>` elements within `<choice>`
  - it is, however, recommended that you limit the number of alternatives to no more than three, and in the rare case where a higher number of genuinely plausible alternatives are possible, instead record the most likely one as `<unclear>` and mention the others in your commentary or apparatus
  - also keep in mind that those readers of your edition who are interested in possible alternatives can be expected to be familiar enough with the script in question to be able to work out those possible alternatives once they have received indication that a certain character is unclear
- we shall not attempt to rigorously **assign probabilities** to each alternative, not even by using `@cert` for some of the alternatives
  - instead, put what is by your judgement the most likely alternative first, and the others in order of decreasing probability
- as for unclear markup in general, feel free to **ignore trivial ambiguities** that can be resolved confidently on the basis of the context
  - in particular, when some pairs of characters look very similar (or wholly identical) in the script of your inscription, it is recommended that you record the expected reading without any markup, e.g.
    - if a word looks like *ṣahārāja* in an inscription where *ṣa* and *ma* are very similar, simply record *mahārāja*
    - if a word looks like *sambatsara* in an inscription where *ba* and *va* are very similar, simply record *samvatsara*
    - **but**, if these words were to occur in an inscription which elsewhere clearly distinguishes the relevant pairs of characters, you would record *ṣahārāja* (marked up as a scribal error, §6.2) and *sambatsara* (as a valid alternative spelling, optionally marked up as non-standard, §6.3)
    - and if an alternative reading alters the meaning **in a non-trivial way**, e.g. if in an inscription where the *akṣaras A* and *su* are very similar, a word could be read as *Adharma* or *sudharma*, then the ambiguity should definitely be marked up (unless, again, you are absolutely confident that it is ruled out e.g. by sandhi or by the wider context)

5.3.4. Reading difficulties below the *akṣara* level

- do not resort to sub-*akṣara* markup (§4.1.2) just because there is some uncertainty regarding one or more specific components of a complex *akṣara*; instead, aim to make the most of Romanisation, which allows you to single out precise segments of text smaller than one *akṣara* of the original
  - thus, if only a certain character component of the original script is unclear or ambiguous, then do not use tags around the transliteration of the entire *akṣara*, e.g.
    - `sph<unclear>u</unclear>rad` and NOT `<unclear>sphu</unclear>rad`
    - `ut<orig>ph</orig>annasya` and NOT `u<orig>tpha</orig>nnasya`
  - likewise, for local markup affecting several transliterated characters, feel free to put the start and end tags at boundaries not perceptible in the original script, e.g.
    - `jayamit<unclear>ray</unclear>ā` and NOT `jayami<unclear>trayā</unclear>`

– localising markup in this precise way allows you to rely on the expertise of the readers of your edition to figure out the exact locus of doubt within a complex original character and its possible implications
  – both in fairly straightforward situations, where the reading of a particular component is doubtful:
    – if you have a reasonable guess for the identity of this component, then simply mark it up as unclear (§5.3.1), e.g.
      – *rddh* with a tentatively read *e*: `rddh<unclear cert="low">e</unclear>`
        – readers conversant with the language and the script will still be able to think of other possible readings (e.g. what looks like an *e* marker may be damage, in which case the vowel is *a*; or the marker may be a damaged *ai* or *i* or *o*)
    – if you have a small number of reasonable guesses for an unclear component, then mark it up as ambiguous (§5.3.3), e.g.
      – *rddh* with what may be *i* or *ī*:
        `rddh<choice><unclear>i</unclear><unclear>ī</unclear></choice>`
    – if you do not know and prefer not to guess whether a vowel marker was attached to a damaged character, then you may still choose to mark it up as a tentative *a* (implying that a different vowel is conceivable) instead of using complex markup, e.g.
      – *k* with plenty of damage around it, which may obscure a vowel mark: `k<unclear cert="low">a</unclear>`
  – and in more complicated situations, such as
    – when **the identification of a component may affect its place in the reading sequence**:
      – if you have a reasonable guess for the identity of this component, then still simply mark it up as unclear (§5.3.1), e.g.
        – *ddha* with a probable *repha*: `<unclear>r</unclear>ddha`
        – readers conversant with the language and the script will still be able to think of other possible readings (e.g. what looks like a *repha* may be damage, in which case nothing is to be read in its place; or it may be a damaged marker for *ā* or *e*, in which case the *r* is to be dropped and the vowel is to be read after the other consonant(s) of the *akṣara*)
      – if you have a small number of reasonable guesses, then mark up the entire sequence involved as ambiguous, e.g.
        – *ddh* with what may be either a *repha* or an *ā* marker attached to its top:
          `<choice><unclear>rddha</unclear><unclear>ddhā</unclear></choice>`
    – when the uncertainty concerns a sequence of **strokes**, some of **which may belong to** either **one character or to another**, you will necessarily have to tag entire sequences as unclear (leaving it to readers to think up alternatives) or ambiguous, e.g.
      – if in the Tamil sequence கா the right-hand set of strokes may be an *ā* marker attached to the preceding *k* or a separate character such as ர *ra*:
        – `k<unclear>ā</unclear>`
        – OR `k<choice><unclear>ā</unclear><unclear>ara</unclear></choice>`[26]
      – if in the premodern forms of Nagari that use so-called *pṛṣṭhamātrā* notation, the sequence पात the central stroke may be an *ā* marker attached to the first character, or an *e* marker attached to the second:
        – `p<unclear>ā</unclear>ta`
        – or `p<choice><unclear>āta</unclear><unclear>ate</unclear></choice>`

[26] In the interest of preserving your sanity, it is advised that you avoid encoding the full spectrum of possibilities, `k<choice><unclear>ā</unclear><unclear>ara</unclear><unclear>ar</unclear><unclear>ra</unclear></choice>` unless all are indeed plausible and worth recording.

5.4. Lacunae

5.4.1. The EpiDoc element `<gap/>`

– this section concerns lacunae, i.e. situations where the originally inscribed text cannot be read at all because it is severely damaged, or because part of the support is altogether gone
– the TEI element `<gap/>` has a wide range of application, indicating "a point where material has been omitted in a transcription [for various reasons including that] the material is illegible, invisible, or inaudible"[27]
– in our EpiDoc editions, this element must always have the following attributes
  – `@reason`
  – either `@extent` or `@quantity`
  – `@unit` [28]
  – see the following subsections for detailed instructions on these attributes
– since a gap is, by definition, a point where text is not available, this element can **never contain text** (i.e. it is an empty element)
  – note that **if you supply the contents of a lacuna** (e.g. by conjecture or from a parallel text), then the lacuna itself must not be marked up as a gap; instead, see §5.5 about supplied text
  – however, if only part of the missing text is supplied, the remaining segment(s) of the lacuna are to be marked up as discussed here and separately from the supplied segment(s)

5.4.2. The reason for a lacuna: illegible or lost

– where parts of the support have been lost altogether, or where the support itself is extant, but its surface has peeled off so that not even the faintest traces of writing remain, the attribute `@reason` shall have `"lost"` as its value:
  – `<gap reason="lost"/>`
– where the inscribed text cannot be read at all, but the support is extant and vestiges of writing remain (i.e. there is a chance, however narrow, that with new insights, technological advances or sheer luck, some of the text can be recovered), the attribute `@reason` shall have `"illegible"` as its value:
  – `<gap reason="illegible"/>`
– where it is impossible to make the above distinction for a certain lacuna, you may use the attribute `@reason` with the value `"undefined"`:[29]
  – `<gap reason="undefined"/>`
  – resort to this if and only if
    – you are encoding your text from a printed edition without access to the original inscription or a visual representation of it
    – *and* the previous editor gives no indication whether a lacuna is illegible or wholly lost
    – *and* you cannot make a reasonable guess as to which of these it is

5.4.3. Inline lacunae

– in most situations, lacunae shall be treated as inline, i.e. line beginnings are to be marked up as usual (§3.2) and lacunae starting in one line and continuing in the next are to be marked up as two separate `<gap/>` elements
  – this applies even to lines that are wholly illegible, provided that you are certain about the presence and number of such lines
  – for exceptions, see §5.4.6 to §5.4.8 below

---

[27] https://www.tei-c.org/release/doc/tei-p5-doc/en/html/ref-gap.html. This definition uses the term "transcription" in a generic sense of the transposition of text from one medium to another, not in the specific sense in which it is distinct from transliteration (see TG §1.4.3).
[28] The EpiDoc schema permits the use of `@extent` without `@unit`, but in our practice, `@unit` shall always be specified.
[29] This was not permitted in earlier versions of the EpiDoc schema, but has been added at our request in January 2020 and will be available in the next release of the schema.

– if you **know the number of characters lost** accurately, encode the length of the lacuna in the same way as that of spaces (§4.3.1), using `"character"` as `@unit` and a numeric value as `@quantity`
  – e.g. `<gap reason="lost" quantity="1" unit="character"/>`
  – unlike spaces, where length expressed in characters is understood by default to be approximate, the length of lacunae expressed in the above way should be quite precise, as in the following circumstances:
    – the text is in syllabic verse which lets you determine the exact number of *akṣara*s lost (give or take a few potential final consonants and/or punctuation marks)
    – although the characters are damaged beyond recognition, they can nonetheless be counted with a very small margin for error
– if you **estimate the number of characters lost** but do not know them precisely, expand the above markup with the attribute `@precision` with the value `"low"`
  – e.g. `<gap reason="illegible" quantity="7" unit="character" precision="low"/>`
  – although TEI affords the facility to do so, we shall not encode any other degrees of precision, nor use minimum and maximum possible values for the length of a lacuna
  – use this method when your estimate can be expected to differ by no more than 20% or so from the actual number of characters lost, as in the following circumstances:
    – you can count the number of characters in the previous or next line for a span of the same width as that of the lacuna
    – the text is in quantitative verse, and you estimate the number of syllables lost on the basis of the number of morae missing from the verse[30]
  – if the **size of a lacuna cannot be counted or estimated in characters**, use the attribute `@extent` with the value `"unknown"` to encode this, e.g.
    – `<gap reason="lost" extent="unknown" unit="character"/>`
– note that there is no separate encoding method for lacunae from the beginning of a line to a certain point, or from a certain point to the end of a line
  – these cases shall simply be encoded by whichever of the above three options is applicable

5.4.4. Lacunae with known metre

– if text cannot be restored, but the prosodic pattern of a lacuna is known, encode an inline lacuna as above, and in addition
  – wrap the `<gap/>` element in a `<seg>` element with the attribute `@met`, encoding its prosody as per Table 2 of Appendix B
  – thus, in fully syllabo-quantitative verse (e.g. *vasantatilakā*), encode a lacuna as `<seg met="++-+---+-"><gap reason="lost" quantity="9" unit="character"/></seg> suvarṇṇa-dāne`
– note that the number of lost characters in syllabic verse can always be calculated accurately, but in moraic verse the size of the lacuna expressed in characters may be only an estimate, requiring the use of `@precision="low"` in the `<gap/>` element
  – thus, in moraic verse (e.g. *āryā*), encode a lacuna as `yo vīkṣya <seg met="3|4|4|4|-"><gap reason="lost" quantity="12" unit="character" precision="low"/></seg> bandhana-niruddhaM`
– when encoding the metre of lost text, disregard:
  – caesurae (which may or may not have been observed by the composer)
  – complex nuanced constraints, as in the first half of an *anuṣṭubh* line
    – instead, any syllable that is not fully determined by the template should be denoted as anceps, e.g. `<seg met="====-+"><gap reason="lost" quantity="6" unit="character"/></seg>vyāpi candraguptākhyam adbhutaM`

[30] In Sanskrit quantitative verse, the number of lost syllables may be estimated at 75% of the number of missing morae. Thus, if a 12-mora chunk is missing from a verse line, estimate the length of the lacuna at 9 characters if you have no better indication. This figure is based on a quick statistical look at a small sample; feel free to use a better approximation if you can, and do not worry too much about the accuracy of the character count.

– legitimate metrical variations, as in *vipulā anuṣṭubh* (instead, treat all lost *anuṣṭubh* verse as *pathyā*)
– and optional constraints such as *vipulā* and *capalā āryā* (instead, treat all lost *gāthā*-type verse as *pathyā*)

5.4.5. Lacunae below the *akṣara* level

– when a particular character component (such as the consonant body, a subscript consonant or the vowel marker) is lost or illegible and cannot be restored even tentatively, but you do have text (of any sort, including unclear and supplied) for other parts of the same *akṣara*, encode special inline lacunae as follows
  – 1. at its logical position in the transliterated text, encode the lacuna with the element `<gap>`, using `"component"` as the value of its `@unit`
    – mark up the gap as illegible or lost as you would normally
    – if more than one component of a character is affected, encode a separate `<gap/>` for each
  – 2. wrap the `<gap/>` element (or each `<gap/>` element separately) in `<seg type="component">` with an applicable value of `@subtype` as per §4.1.2
    – if the lost component is a vowel whose prosodic length is known (because it is in verse, or deduced from the extant phonemic context) then add the attribute `@met` (as per §5.4.4) to the `<seg type="component">` wrapper (instead of adding an extra `<seg>` element to be qualified by `@met`)[31]
  – 3. if you deem that there is potential ambiguity regarding *akṣara* boundaries, feel free to wrap transliterated characters and lacunae belonging to a single original character in the element `<seg type="aksara">` as per §4.1.1

Example 5.4.5.A: lost consonants

➢ a vowel marker for *ā* and an *anusvāra* are visible in the last partial line of a fragment

```
<seg type="aksara"><seg type="component"
subtype="body"><gap reason="illegible" quantity="1"
unit="component"/></seg>ā</seg>
<seg type="aksara"><seg type="component"
subtype="body"><gap reason="illegible" quantity="1"
unit="component"/></seg><unclear
cert="low">a</unclear>ṁ</seg>
```

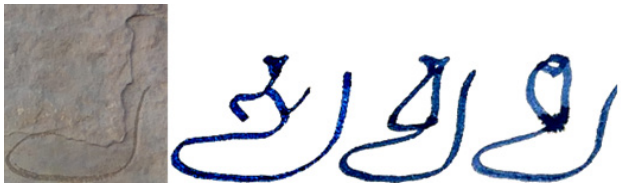
Example 5.4.5.B: lost vowel marker

➢ the consonant *t* has so much damage around it that it may have had any of several vowel marks or none, as in the hypothetical image here
➢ some candidates are shown on the right
➢ this is a scenario which some of us are used to transliterating as *tV*

```
t<seg type="component" subtype="vowel"><gap reason="illegible" quantity="1"
unit="component"/></seg>
```

[31] Keep in mind that the markup should reflect the positional value of the vowel and not its length by nature: if a vowel is followed by more than one consonant, then it is positionally long even if it is a short vowel by nature.

Example 5.4.5.C: lost body with subscript component



- a clear subscript *y* survives but the principal consonant(s) are obliterated along with any vowel marker, as in the hypothetical image above
- some candidates are shown on the right

```
<seg type="component" subtype="body"><gap reason="illegible" quantity="1"
unit="component"/></seg>y<seg type="component" subtype="vowel"><gap reason="illegible"
quantity="1" unit="component"/></seg>
```

Example 5.4.5.D: a complex sequence of partially lost characters

- a sequence comprised of the following elements, which are known to follow the prosodic pattern – – ⏑
- the legible character *ku*, which is simply transliterated
- one wholly illegible *akṣara*, which we prefer not to encode simply as a lacuna of one character which is prosodically long, as this would obscure the fact that since the preceding (clear) *ku* is prosodically long, the present lost *akṣara* must be a conjunct
- a clear regular *y* that may or may not have had a vowel marker attached

```
ku<seg type="aksara"><seg type="component" subtype="conjunct"><gap reason="illegible"
quantity="1" unit="component"/></seg><seg type="component" subtype="vowel" met="+"><gap
reason="illegible" quantity="1" unit="component"/></seg></seg>y<seg type="component"
subtype="vowel" met="+"><gap reason="illegible" quantity="1" unit="component"/></seg>
```

- in the above example, the tag `<seg type="aksara">` around the second (wholly lost) character is not strictly necessary, but it has been added to make it explicit that the illegible conjunct consonant and the illegible vowel comprise one *akṣara*

5.4.6. Entire lines lost

– when a small and precisely known number of lines is lost, encode each line beginning (§3.2.1) and populate each line with separate inline `<gap/>` elements (§5.4.3) with estimated quantity or unknown extent
  – e.g. for a gap of two full lines, `<lb n="3"/><gap reason="lost" quantity="30" unit="character" precision="low"><lb n="4"/><gap reason="lost" quantity="30" unit="character" precision="low">`
– the size of larger lacunae may be encoded using `"line"` instead of `"character"` as the value of `@unit`
  – see §5.4.7 below for additional considerations in such cases
– to encode **a precisely known number of lost or illegible lines**,
  – use `"line"` as the value of `@unit`
  – e.g. `<gap reason="lost" quantity="3" unit="line"/>`
– to encode an **unknown or uncertain number of lost or illegible lines**,
  – if **the number of lost lines can only be estimated**, but not counted precisely
    – use `@precision="low"` to show that the number of lines lost is an estimate
    – e.g. `<gap reason="lost" quantity="3" unit="line" precision="low"/>`
  – if **the number of lost lines is unknown**
    – use `@extent` with the value `"unknown"` instead of `@quantity`, but retain `@unit` with the value `"line"` to distinguish such spaces from inline spaces of unknown length
    – e.g. `<gap reason="lost" extent="unknown" unit="line"/>`
– to encode **lines possibly lost**,[32] i.e. situations where it is impossible to tell whether there were more lines to an inscription than are now extant

[32] Or, in principle, lines possibly illegible. But such a situation seems unlikely: if vestiges are so scant that you cannot even tell whether there ever was writing in a certain area, then that text is for all purposes better marked up as "possibly lost" instead of "possibly illegible".

  - within the `<gap>` element, add `<certainty match=".." locus="name"/>`, where
  - `@match=".."` indicates that we are encoding uncertainty regarding the parent element (i.e. `<gap>`), and
  - `@locus="name"` indicates that the uncertainty concerns the name of the parent element (i.e. the fact that what we have there is a lacuna)
  - note that in this single case, the `<gap>` element is not empty but comprised of separate opening and closing tags wrapping the `<certainty>` element
- for example,
  - one line possibly lost: `<gap reason="lost" quantity="1" unit="line"><certainty match=".." locus="name"/></gap>`
  - up to approximately two lines possibly lost: `<gap reason="lost" quantity="2" unit="line" precision="low"><certainty match=".." locus="name"/></gap>`
  - any number of lines possibly lost: `<gap reason="lost" extent="unknown" unit="line"><certainty match=".." locus="name"/></gap>`

5.4.7. Massive lacunae

- extensive lacunae can disrupt the extrinsic and intrinsic structure of the encoded text and shall therefore be handled as follows
  - see also §5.4.8 below for the special case of lost copper plates
- if you can restore part of the lost text, encode your restored text as per §5.5, but do not include any reconstructed structural elements in `<supplied>` tags
  - in the instructions below, all points concerning extant text apply equally to text restored by you in the edition
- for instructions concerning the numbering of elements where massive lacunae are involved, see the specific passages on final, initial and medial lacunae below
- the **general procedure** for encoding massive lacunae is as follows
  - carefully encode block-level containers (`<p>`, `<ab>`, or `<lg>` and `<l>`):
    - all (extant or restored) text must be within such a container, but do not create additional containers to hold only lacuna markup (and no text)
    - if the beginning or end of one of these containers falls within a lacuna, place the start-tag or the end-tag at the point where (extant or restored) text begins or ends, and add the attribute `@part` to the container,
      - with the value `"I"` if the initial part of the container is extant (the end is lost in a massive lacuna)
      - with the value `"F"` if the final part of the container is extant (the beginning is lost in a massive lacuna)
      - with the value `"M"` if only the medial part of the container is extant (both the beginning and the end are lost in a massive lacuna)
    - note that if an `<l>` element is interrupted by a massive lacuna, you will need to close both the current `<l>` and, after it, the `<lg>` element wrapping it, and add the attribute `@part` to both of these elements
      - while if the start of the lacuna coincides with the start of a line, `@part` is not applicable to that `<l>` element, but may still be applicable to the enclosing `<lg>` element if the stanza is incompletely preserved
  - carefully encode pointlike structural elements (`<lb/>`, `<pb/>` and `<milestone/>`), paying attention to the following:
    - when an epigraphic line is partially present, i.e. it has at least a little bit of (extant or restored) text at the beginning or end,
      - make sure the `<lb/>` for that line is present, but do not encode `<lb/>` elements for any additional lines believed or known to be lost
      - encode an inline lacuna for the final or initial part of that line where no text is available

  - if pages or a pagelike partitions are involved, make sure the `<pb/>` or `<milestone type="pagelike"/>` is present for any such unit that includes any (extant or restored) text, but do not encode these elements for any additional units believed or known to be lost
    - see §5.4.8 for specific guidance on dealing with incomplete copper plate sets
  - encode the rest of the lacuna outside any reconstructed elements, as a known, estimated or unknown number of lost lines or, if applicable, as possibly lost lines (see §5.4.6 for all of these methods)
- when according to the above instructions it would be **necessary to create a structural element** (a container or an empty element representing a transition point) **only for the sake of restored text**, i.e. when a restoration extends into a text container, line or page of which no part is extant,
  - you may **optionally forgo the creation** of such an element and, instead of including the restoration in the text of your edition, mention the restoration in an apparatus note (§9.1.7)
  - this method is recommended especially in the following cases:
    - for very short restorations (smaller than one word)
    - for the restoration of widely occurring text such as standardised genealogies or stock admonitory verses in land grants
    - for restorations where several alternatives are deemed possible
- the points below summarise specific applications of the above general procedure for final, initial, medial and bilateral lacunae, and give guidance on the numbering of elements when massive lacunae are involved
- to encode **a text whose end is lost** (a massive final lacuna):
  - close the currently open block-level container (`<p>`, `<ab>`, or `<lg>` and `<l>`) directly after the last (extant or restored) bit of text
    - add the attribute `@part` with the value `"I"` (for initial) to the interrupted container unless it is filled up to its end by extant or restored text
  - encode an `<lb/>` for the last line that has any (extant or restored) text and an inline lacuna for the end of that line if incomplete
  - outside the last block-level container, encode a multiline lacuna for subsequent lost text
  - number your lines and stanzas consecutively up to the last extant or restored item

Example 5.4.7.A: massive final lacuna

```
<p part="I">...
<lb n="5"/>tāpasāśrama-va<unclear cert="low">ne</unclear>
</p>
<gap reason="lost" extent="unknown" unit="character"><!--Lacuna to the end of the last line -->
<gap reason="lost" extent="unknown" unit="line"/><!--Lacuna after the last line -->
```

- to encode **a text whose beginning is lost** (a massive initial lacuna),
  - open the first block-level container (`<p>`, `<ab>`, or `<lg>` and `<l>`) directly before the first (extant or restored) bit of text
    - add the attribute `@part` with the value `"F"` (for final) to the interrupted container unless the lacuna happens to end at the precise point where the container begins
  - encode an `<lb/>` for the first line that has any (extant or restored) text
    - if the beginning of this line is also lost, encode an inline lacuna after the `<lb/>` element
    - if the beginning of this line is extant, but the first word is incomplete (i.e. if the beginning of that word was in the lost previous line), add `@break="no"` to the `<lb/>` element (§3.2.4)
  - outside the first block-level container, encode a multiline lacuna for preceding lost text
  - number your lines and stanzas consecutively
    - generally, start with 1 at the first encoded element of each type
    - but if the number of lost lines is precisely known, then assign numbers to each of those (in the `<lb/>` element for each, if you encoded them individually; or mentally, if you encoded a single lacuna of multiple lines) and start numbering the extant lines in logical succession

Example 5.4.7.B: massive initial lacuna

```
<gap reason="lost" extent="unknown" unit="line"/><!--Lacuna before the first line -->
<lb n="1"/><gap reason="illegible" precision="low" quantity="7" unit="character"/><!--Lacuna from
the beginning of the first line -->
<p part="F">bhagavat-paśupati-bhaṭṭāraka-pādānugṛhīto bappa-pādānu<lb n="12" break="no"/>dhyātaḥ
parama-bhaṭṭāraka-mahārājādhirāja-śrī-narendradevaḥ kuśali …
</p>
```

– to encode **a text with a chunk lost from the middle** (a massive medial lacuna),
  – encode the initial chunk of extant text as one with a final lacuna, but do not encode lost lines at the end
  – encode the final chunk of extant text as one with an initial lacuna, but do not encode lost lines at the beginning
  – if the total number of lost medial lines cannot be estimated confidently, encode the extant sections as textpart divisions (§3.4)[33]
    – here, you are essentially treating your inscription as consisting of unconnected fragments, even if the massive lacuna is merely due to surface damage
    – putting the final extant chunk in a new textpart division allows (and compels) you to restart line and stanza numbering from 1
    – since textpart divisions encoded as fragments necessarily imply the presence of lacunae between the textparts, lines lost between the fragments shall not be encoded explicitly
      – however, do encode an inline lacuna for the remainder (end or beginning) of any line that contains (extant or restored) text
  – if the total number of lost lines is known or can be confidently inferred, there is no need to split your edition into two textparts; instead,
    – between the last block-level container of the first extant chunk and the first block-level container of the second, encode a multiline lacuna
      – preferably by creating `<lb/>` elements for each lost line and populating these with inline `<gap/>` elements with an estimated number of characters
        – in this case, number the reconstructed `<lb/>` elements as you would normally, and continue numbering in the second extant chunk
      – or, if the number of lost lines is large (yet still known precisely), by encoding a single `<gap/>` element of multiple lines
        – in this case, mentally assign a line number to each lost line, and in the second chunk of text, continue the numbering of your encoded `<lb/>` elements in logical succession
    – if your inscription includes stanzas, ignore potential fully lost stanzas in the lacuna and continue numbering in the second extant chunk where you left off in the first
      – or, if you can infer the number of lost stanzas confidently (e.g. because stanzas are numbered in the original), mentally assign numbers to the lost stanzas and continue the numbering of encoded stanzas in logical succession
      – or, if you can infer both the number of lost stanzas and their position relative to line beginnings (e.g. because the inscription has exactly one stanza per line throughout), reconstruct the `<lg>` and `<l>` elements for each stanza and populate these with separate inline `<gap/>` elements instead of encoding a single inline `<gap/>` for each epigraphic line
– to encode **a text with chunks lost from both the beginning and the end** (a massive bilateral lacuna),
  – apply the considerations for an initial lacuna at the beginning of your extant chunk and those for a final lacuna at the end of your chunk
  – should it be the case that the surviving text is a single block-level element (`<p>`, `<ab>`, or `<lg>` and `<l>`) that is incomplete on both ends, add the attribute `@part` with the value `"M"` (for medial) to the interrupted container

[33] See Example 3.4.5.A for an illustration of such encoding.

5.4.8. Lost copper plates

– incomplete sets of plates shall be handled like any other massive lacuna (§5.4.7), with the following additional considerations
– although it would be theoretically permissible to use `"page"`, `"plate"` or `"folio"` as the value of `@unit` in a `<gap/>` element, we see no practical advantage to doing so
– lost pages do not as a rule need to be reconstructed in your edition, except for the special considerations for lost initial and medial plates, set out below
  – instead of encoding lacunae for lost pages, the fact that entire plates are lost shall be recorded in the commentary
  – if a restoration extends into a lost plate, it is preferable to record that restoration in an apparatus note (§9.1.7) rather than in the edition
– however, if you deem it essential to restore text for a lost plate within your edition,
  – then create all necessary `<lb/>` and `<pb/>` elements to accommodate the restored text
    – if only one face (recto or verso) of a lost plate is involved in restoration, then it is sufficient to reconstruct a `<pb/>` element only for that face
  – whenever you reconstruct a `<pb/>` for the sake of a line with restored text, the remaining (unrestored) lines on that page must be encoded as a multiline lacuna of a known, estimated or unknown number of lines
– depending on your corpus, it may be possible to **confidently estimate the number of lost pages, and** even that of **lines** on each lost page
  – if this is the case, feel free to reconstruct `<pb/>` elements for each face of each lost plate
  – in this case, populate each reconstructed page with a multiline lacuna of a known, estimated or unknown number of lines
– in a text with **lost final plate(s)**, simply end your edition at the end of the (extant or restored) text, closing the currently open block-level container and adding `@part="I"` to it if it is incomplete
– in a text with **lost initial plate(s)**,
  – number pages as follows:
    – if the number of lost plates is certain, then number each encoded page logically
      – i.e. if you reconstructed `<pb/>` elements for all lost pages, number them starting with 1r; otherwise, mentally start numbering with the first lost page, and in your edition, start your encoded numbering with the page number applicable to the first actually encoded (extant or reconstructed) page
    – if the number of lost plates is uncertain, start your numbering with the first actually encoded (extant or reconstructed) page as follows:
      – if there is no restored text before the extant text, start page numbering with 1r for the first extant page
      – if a restoration precedes the extant text, then start page numbering with 1v for the reconstructed page, and 2r on the first extant page
    – if the first line on the first extant page begins inside a word, then remember to add `@break="no"` to both the `<lb/>` and the `<pb/>` element encoding the beginning of that line and page (§3.2.4)
  – number lines as follows:
    – if the total number of lost lines is certain (because the number of lines per lost page and the number of lost pages are both known), then depending on the general preference for your corpus (§3.2.2), you may
      – either number lines consecutively in a logical scheme, i.e. mentally start numbering with the first lost line, and in your edition, start your encoded numbering with the line number applicable to the first actually encoded (extant or reconstructed) line
        – this is applicable regardless of whether you reconstruct page beginnings for the lost pages or not
        – if you do reconstruct lost pages, keep in mind that the line beginnings on those pages need not be reconstructed individually, but may be encoded as a lacuna of a known number of lines
      – or restart line numbering on each page, and use complex line numbers

- if the total number of lost lines is not known for certain (because there is uncertainty as to the number of lost pages, or to the number of lines per lost page), then depending on the general preference for your corpus(§3.2.2), you may
  - either number lines consecutively, but starting from the first actually encoded (extant or reconstructed) line (and ignoring the lost lines)
  - or restart line numbering on each page, and use complex line numbers
- in a text with **lost medial plate(s)**,
  - if the number of lost pages is known or can be confidently inferred,
    - end any open block-level containers before the lacuna (using `@part="I"` if applicable)
    - open new block-level containers after the lacuna (using `@part="F"` if applicable)
      - for the sake of consistency, open the new container before the first extant `<pb/>` element in the final part of the text
    - outside the above block-level containers, reconstruct `<pb/>` elements for each lost page (i.e. two per plate) regardless of whether they hold restored text or not
      - populate each of these with a multiline lacuna of a known number of lines
    - number all actually encoded `<lb/>` elements
      - if the number of lost lines per page is certain, you may number lines consecutively throughout your edition (in this case, mentally assign line numbers to the lost lines encoded as multiline lacunae, and after the lacuna, continue numbering the actually encoded `<lb/>` elements with the next number)
        - depending on corpus preferences, the option of restarting line numbering on each page (and using complex line numbers) is of course available even if the number of lost lines is certain
      - if the number of lost lines per page is uncertain or unknown, you must restart line numbering on each page (and use complex line numbers) even if the general preferences for your corpus dictate otherwise
        - note that if the last line of a lost medial page contains restored text and is preceded by an unknown or approximate number of lost lines, the number of that reconstructed line should be 1 in this numbering scheme (because it is the first actually encoded `<lb/>` element on that page)
    - see Case study 2B in Appendix C for an illustration of the encoding of a reconstructed medial plate
  - if the total number of lost medial pages cannot be estimated confidently, encode the extant sections as textpart divisions (§3.4, §5.4.7) to eliminate difficulties with page and line numbering
    - within each textpart, encode the relevant pages exactly as prescribed above for lost final and initial plates respectively
    - do not reconstruct any page or line beginnings believed or known to be lost, except when you find it essential to include a restoration in your edition that requires these
    - in the second textpart, restart page and line numbering
    - see Case study 2C in Appendix C for an illustration of the encoding of a missing medial plate with textpart divisions

5.4.9. Fractured inscriptions

- when an initial, medial or final fragment of an inscription is lost, the general guidelines for massive lacunae apply (§5.4.7)
- inscriptions consisting of only one extant fragment need no markup for partitions
- inscriptions with two or more extant fragments need to be encoded differently depending on whether the fragments can be connected (with the help of the extant text, through a restoration of at least some of the lost text, or through a reconstruction of the extrinsic structure)
- if **fragments are connected by** one or more lines of extant or restored **text** running across them, then they can be edited without resorting to textpart divisions
  - it is recommended that you encode the boundaries of such fragments using gridlike partitions (§3.6 and Example 3.6.6.C)
  - when encoding gridlike partitions, lacunae resulting from weathering at the fractured edges or from the loss of one or more fragments may be joined to either adjacent segment

  - but when lacunae are partially restored in such a case, it is preferable to join each restoration to the fragment whose surviving text serves as the basis of the restoration
  - the same method is applicable if no extant or restored text connects the fragments, but the number of lines lost between them can be confidently estimated (for instance on the basis of metre and content)
- if **no** extant or restored **text connects** some **fragments**, but it is possible to deduce their reading order and to confidently estimate the number of lines lost between them, then the above method is applicable with the following additional considerations
  - if the number of lost lines is more than zero, the lacuna must be encoded as a gap extending over a known number of lines
    - even in this case, the fragments shall be encoded as gridlike partitions (if at all), not as pagelike partitions
  - if the number of lost lines is zero (one or more lines of one fragment are assumed to belong to the same original line as one or more lines on another fragment even though no extant or restored text fills up the gap), then the overlapping lines of the fragments shall be encoded as parts of the same encoded line
- if **nothing** extant or restored **connects** some **fragments**, and thus the number of lines lost between them is uncertain and even their reading order may be doubtful, then they must be encoded as boxlike partitions (§3.4 and Example 3.4.5.A)
  - the corresponding textpart divisions shall follow one another in the (presumable) order in which they appeared in the original
  - lacunae shall not be encoded for any text between the surviving fragments
  - restorations shall be encoded attached to the fragment which serves as the basis of restoration
  - the same method is applicable if parts of the same original line may be preserved on several fragments, but the original structure cannot be reconstructed as a gridlike partition

5.5. Restoring Lacunae

5.5.1. Marking up restored text

- where you as editor restore parts of the text that can no longer be made out in the original, the restored segments must be wrapped in the element `<supplied>`, normally using `"lost"` as the value of `@reason`
  - e.g. `<supplied reason="lost">sodra</supplied>ṅgaḥ soparikaro`
- instead of `"lost"`, you may use the value `"undefined"` for `@reason` if and only if
  - you are encoding your text from a printed edition without access to the original inscription or a visual representation of it
  - *and* the previous editor gives no indication whether the supplied text was omitted (for which see §6.2.4) or lost
  - *and* you cannot make a reasonable guess as to which of these it is
- bear in mind that, as discussed in §5.1, restoration with `<supplied>` is for cases where the basis of restoration is solely the (immediate or wider) context
  - conversely, if vestiges of text can be made out to a degree sufficient to corroborate a restoration, then (regardless of how scant these vestiges are) the text should be treated as an unclear reading rather than a restoration (see also §5.1)
    - e.g. `<supplied reason="lost">pitṛ</supplied><unclear>bhiḥ</unclear> saha pacyate` (where vestiges give some confirmation for *bhiḥ*, whereas *pitṛ* is wholly gone)
- the element `<gap/>` must not be used for a restored lacuna:
  - from a text-encoding point of view, `<supplied>` marks a stretch of text as a restoration, while `<gap/>` stands for an absence of text *in the edition*, and not for a lacuna *on the support*
  - in partially restored lacunae, `<supplied>` and `<gap/>` must, of course, be used side by side
- restored text must, like extant text, be marked up for extrinsic and intrinsic structure

  - however, structural markup (including both containers and empty elements) must never be inside the `<supplied>` tag
  - therefore, some longer restorations will need to be split up into several `<supplied>` elements
- in addition to the mandatory attribute `@reason`, the optional attribute `@cert` with the value `"low"` may be added to indicate a **tentative restoration**, where different restorations (of a similar or a different ultimate meaning) may be feasible
  - e.g. `nirodha-parimokṣa-śīghram iva pāṇḍu gāṅgaṁ <supplied reason="lost" cert="low">payaḥ</supplied>`

5.5.2. The basis of restoration

- by default, restoration will be assumed to be conjectural
  - conjectural restoration thus needs no explicit encoding beyond that outlined above
- the attribute `@evidence` may be added to `<supplied>` to indicate a restoration based on something other than conjecture, with the following permitted values:
  - `"parallel"` - restoration on the basis of one or more parallel texts
    - in standard EpiDoc usage, this means a parallel specimen of a text as a whole, but in our usage, it can be expanded to epigraphic parallels of certain segments of a text, such as:
      - a genealogy found in (nearly) identical form in many copper plates or seals of a dynasty
      - a repeatedly used standard title of a ruler
      - a stanza found in more than one instance in your corpus
    - if your edition includes a restoration of this type, the parallel text(s) used as evidence should be identified in the commentary to your edition
      - such identification shall be in a human-readable form, but if the parallel text already has an ID in the DHARMABase, then this ID should be mentioned (and may be encoded as a reference, see §10.4.6)
  - `"previouseditor"` - text that has been read by a previous editor of the inscription, but which is no longer possible to make out at present
    - note that there is no facility to distinguish between multiple previous editors within this tag; if such a distinction is necessary, it shall be made in the apparatus attached to your edition (§9.1)
    - likewise, alternative conjectural restorations should be recorded in the apparatus (if at all); `@evidence="previouseditor"` is only for cases where an earlier editor reports an actual reading for text now lost

6. Editorial Intervention

6.1. Correction and Normalisation

6.1.1. Correction versus normalisation

– the editorial rectification of a phenomenon deemed to be a scribal mistake is here referred to as **correction**
  – a correction is thus a restoration of the text to the form that you believe the composer of the text had intended
  – as a corollary, just because the text is not up to textbook standards does not mean that it requires correction, and the text as corrected by us need not necessarily be up to textbook standards
– the editorial alteration of a phenomenon deemed to be non-standard usage into something that fits the standard more closely is here referred to as **normalisation**
  – if what you believe to have been the composer's intent differs from the standard for the language in question (inasmuch as a standard may be said to exist), you may normalise the original usage for purposes such as:
    – to help readers understand the text and to show how you interpret it
    – to facilitate text queries by ensuring that the standard form is present in the XML file and can thus be returned as a match for searches even if the actual text differs from the standard
– **distinguishing scribal error from non-standard usage** may be problematic and will often involve a subjective decision
  – deviations that involve the exchange of a character to a graphically similar one are likely to be scribal errors
  – deviations from expected forms are more likely to be non-standard usage if they occur repeatedly in an inscription
  – deviations that seem to be governed by the immediate phonemic context are more likely to be non-standard usage
  – deviations that involve the exchange of a character to a phonetically similar one are likely to be non-standard usage
  – grammatical solecisms are to be considered non-standard usage, not scribal error
  – when in doubt, prefer normalisation and use correction only in clear cases of scribal error

6.1.2. Markup methods for correction and normalisation

– TEI and EpiDoc afford the following methods for the editorial treatment of incorrect or non-standard text
– **no action**: depending on the nature of your text and corpus, you may opt not to mark up at all certain trivial scribal errors and common non-standard usage
– **flagging** without further action serves to highlight an erroneous or non-standard spot
  – the purpose of flagging is twofold:
    – it calls the attention of the reader to unexpected text, and
    – it makes it clear to the reader that the unexpected text is not your editorial mistake
  – see §6.2.1 about flagging erroneous or unintelligible text, and §6.3.1 about flagging non-standard usage
– **rectification by substitution**: when an error can be corrected or a non-standard form can be normalised by substituting some received characters with others, your encoded edition must include both the received and the rectified reading
  – both of these alternatives must be tagged as such, and wrapped together in an element signifying that one is an alternative to the other
  – see §6.2.2 about correcting errors in this way, and §6.3.2 about normalising usage in this way
– each of the above methods is available for both correction and normalisation, using the tags described in the subsections referred to above

– in addition, TEI and EpiDoc allow two more methods dedicated to the suppression of superfluous characters and the restitution of omitted characters
  – **correction by suppression**: erroneously engraved superfluous characters may be marked up for editorial suppression
    – see §6.2.3 about suppressing scribal errors of redundancy
  – **correction by restitution**: erroneously omitted characters may be supplied and marked up as an editorial restitution
    – see §6.2.4 about supplying erroneously omitted characters
  – our project has chosen to dedicate these encoding methods solely to the rectification of anomalies deemed to be erroneous (i.e. not in accordance with the composer's intent) as opposed to non-standard (i.e. deliberately used by the composer)
    – therefore, when you wish **to normalise orthography by adding or suppressing individual characters**, you must resort to substitution as described above
    – see §6.3.4 for advice on how best to do this in various situations

6.1.3. Good practice in editorial intervention

– keep in mind that everything in §5.5 concerns alterations made by a modern editor; premodern editorial alterations to the actual inscribed text are covered in §4.5
– the foremost rule for editorial alterations of the received text is that they must **never be silent**
  – your digital edition must always include the text as found on its support, and any changes you make to create an abstract text must be shown in markup, as detailed below
  – apparent exceptions to this rule (such as editorial hyphenation, *avagraha*s, etc.) are only apparent, as our system will know that they are editorial and will be able to strip them away to obtain a purely diplomatic edition
– **editorial rectification** of the text **is optional**; in many cases less is better
  – in particular, do not supply stanza punctuation or numbering where such are not present in the original (but do restore them as per §5.5 whenever you are certain that such things *were* present and have been lost to damage)
– when you rectify a feature by substitution, keep in mind that it must always be possible to produce the received text by ignoring the segment tagged as editorial and, vice versa, to produce the corrected text by ignoring the segment tagged as received
  – see §6.2.2 erroneous text and §6.3.2 for the specific markup involved and for examples
– editorial intervention should make it easy for a scholarly reader to see why the editor has flagged or altered the text, and this purpose can be facilitated by avoiding complex markup where possible
  – in general: try to find a common-sense optimum between minimising the scope of markup and minimising the complexity of markup
  – the **size of segments** to which you apply any of the tags discussed throughout §5.5 **is technically irrelevant** and there are no hard and fast rules to decide it
  – so long as the received text is faithfully reproduced (and, as mentioned in the previous point, if an editorial rectification is present, then that too is accurate), the tagging of a short segment and the tagging of a longer segment that includes characters not directly involved in the anomaly are functionally equivalent
  – the outcome of this is that you need not worry too much about the size of a text segment you flag or rectify: simply proceed as feels most appropriate in the given circumstances
– see §6.2.6 and §6.3.4 for further guidance specific to correction and normalisation

6.1.4. Correction and normalisation in verse

– the guidelines in this subsection apply when the prosody of a metrical segment is disrupted by the presence of a scribal error or non-standard usage, or by the correction/normalisation thereof
– the leading principles are the following:

  - if a correction or normalisation is encoded in the text, then it shall be the prosody of the text **after** correction/normalisation that determines whether or not it is necessary to encode a metrical deviation (with the attribute `@real`, as per §2.3.5)
  - correct prosody should be prioritised over linguistic neatness, so
    - you should always intervene where an intervention can restore faulty prosody to the expected
    - but preferably abstain from encoding either a correction or a normalisation (and instead, merely flag the spot) if doing so would disrupt otherwise correct metre
- thus, in specific cases, proceed as follows
- 1. if the **prosody is anomalous**, and the **correction** of an error **or** the **normalisation** of non-standard orthography or morphology **can restore it** to the expected pattern, then
  - mandatorily carry out this intervention, even if you would ignore or merely flag the same non-standard feature in other circumstances
    - moreover, mandatorily encode this as a correction, even if in other circumstances you would encode the same intervention as normalisation
  - do not add `@real` to the `<l>` element affected (see also §2.3.5)
  - the underlying assumption in this case is that the composer had the correct or standard form in mind, but that has been replaced by an incorrect or non-standard form in the process of the creation of the inscription
  - e.g. in an odd *pāda* of an *anuṣṭubh* stanza, `<l n="c">ṣaṣṭi <choice><sic>varuṣa</sic><corr>varṣa</corr></choice>-sahasrāṇi</l>`, where *varuṣa* is a vernacularised spelling of *varṣa* that is hypermetrical here, so its alteration to *varṣa* is encoded as a correction, and since the intervention restores the expected metre, `@real` is not encoded on the line
- 2. if the **prosody is anomalous**, and there is **no straightforward way to restore it** to the expected pattern **by correction or normalisation**, then
  - it is generally preferable in such cases to merely flag the spot and to add `@real` to the `<l>` element affected (§2.3.5), optionally mentioning the possible correction/normalisation in an apparatus note (§9.1.7)
  - however, if you judge it essential, you may choose to encode a correction or normalisation in the text itself
    - if you do apply correction/normalisation which still leaves the prosody deficient, then the pattern encoded in `@met` must correspond to the prosody of the text **after** correction/normalisation
- 3. if the **prosody is anomalous**, but the text is linguistically **standard, correct and meaningful**, then
  - keep in mind that incorrect/non-standard metre does not in itself constitute grounds for correction or normalisation
  - if you as editor perceive the metrical anomaly as an error on the part of the composer or the engraver *and* you can rectify it without disrupting the prosody and altering the meaning, then you are free to do so (i.e. to proceed as in case 1 above)
  - if, however, you cannot restore the prosody without altering the meaning and/or you believe the prosodically anomalous text may have been what the composer actually had in mind, then flag the spot of text associated with the prosodic anomaly as non-standard, but do not encode a correction or normalisation
    - in this case always add `@real` to the `<l>` element affected (§2.3.5)
    - e.g. in an even *pāda* of an *anuṣṭubh* stanza, `<l n="a" real="-++--+++-">sva-dattāṁ para<lb break="no" n="20"/>-dattām <orig>vāpi</orig></l>`
      - instead of *vāpi*, the standard version of this frequently used stanza has *vā*, which is metrically correct; but since *vāpi* is morphologically and orthographically correct, fully standard, and meaningful in the context, it is assumed to be deliberate and only flagged as original and not corrected to *vā*
- 4. finally, if **the text as received is prosodically regular while being erroneous or non-standard** in such a way that **correction or normalisation would disrupt the metre**, then
  - preferably abstain from carrying out the correction or normalisation, instead simply flagging the spot as erroneous or non-standard, and mentioning the correct/standard form in an apparatus note (§9.1.7)

  - e.g. in a *vasantatilakā* stanza, `<l n="a">sa<orig>ḥ ssa</orig>rvva-satva-satat<orig>ārtthibhi</orig> nitya-dātā</l>` (the metre is correct, but the form *°ārtthibhi* is a solecism for expected *°ārtthibhyo*, which in turn would be metrically incorrect)
  - if you deem that correction or normalisation within the text is essential, then you may encode it
    - but in this case, do add `@real` to the `<l>` element affected (§2.3.5), with a value corresponding to the prosody of the text **after** correction/normalisation
    - e.g. normalising the above example, `<l n="a" real="++-+---++++-++">sa<orig>ḥ ssa</orig>rvva-satva-satat<choice><orig>ārtthibhi</orig><reg>ārtthibhyo</reg></choice> nitya-dātā</l>`
- it may occasionally be the case that **the text as received is linguistically standard but prosodically irregular**, and the **prosody could be corrected** by the application of a straightforward "**de-normalisation**"
  - in such cases, leave the text without encoding, add `@real` to the `<l>` element affected (§2.3.5), and explain the situation in an apparatus note (§9.1.7)
  - e.g. in an even *pāda* of an *anuṣṭubh* stanza, `<l n="a" real="-+-+-+-+">kṣitīśa-siṅhavarmmaṇas</l>` (accompanied by a note explaining that the non-standard form *siṅhavarmmasya*, known to occur in related texts, would be metrically correct)

6.2. Encoding Correction

6.2.1. Flagging erroneous and uninterpretable text

- **to flag items** without correction, wrap the relevant characters with the element `<sic>`
  - by EpiDoc convention, this markup is used for text that is legible but does not seem intelligible
    - uninterpretable tentative readings of mostly unclear characters do not require flagging in this way, but it is permitted to flag a segment of text that includes unclear characters
- for example,
  - `mahār<sic>a</sic>ja` (*ā* would be correct)
  - `<sic>marnta kali-kulanām</sic>` (uninterpretable)

6.2.2. Correcting erroneous text

- **to correct** scribal errors **by substitution**,
  - flag the original text with `<sic>` as above
  - add the corrected alternative directly after this, wrapped in the element `<corr>`
  - and wrap both these elements in the element `<choice>`
- for example,
  - `mahār<choice><sic>a</sic><corr>ā</corr></choice>ja` (*a* corrected to *ā*)
- when correcting by substitution, keep in mind that it must always be possible to produce the received text by ignoring the segment tagged with `<corr>` and, vice versa, to produce the corrected text by ignoring the segment tagged with `<sic>`
  - thus, when for example encoding a correction of *pautrā* to *pautraḥ*, each of the following are **incorrect**:
    - `pautr<choice><sic>ā</sic><corr>a</corr></choice>ḥ` (this encodes a correction of *pautrāḥ* to *pautraḥ*)
    - `pautr<choice><sic>ā</sic><corr>raḥ</corr></choice>` (this encodes a correction of *pautrā* to *pautrraḥ*)
    - `paut<choice><sic>rā</sic><corr>aḥ</corr></choice>` (this encodes a correction of *pautrā* to *pautaḥ*)

6.2.3. Editorial deletion

- where you find that one or more characters were **erroneously added** by the scribe, and you correct the text by suppressing the superfluous segment (without substituting anything else for it),
  - enclose the superfluous characters in the element `<surplus>`

– before marking up an editorial deletion, be sure that you have read and understood §6.2.5 and that editorial deletion is the correct encoding in your case
– editorial deletion should always be used to highlight instances of dittography, e.g.
  – `naika-samara-śata<surplus>ta</surplus>vijayinā` (*śatata* was inscribed instead of *śata*)
  – `veda-vyāsena vyāsena <surplus>vyāsena</surplus>` (three iterations of *vyāsena* where two iterations are correct)
– other superfluous characters or components may, at your discretion,
  – be deemed erroneous and corrected in this way,
    – e.g. `datta<surplus>ḥ</surplus>s tataḥ`
  – or be considered non-standard usage and treated as such

6.2.4. Editorial addition

– where you find that one or more characters were **erroneously omitted** by the scribe, and you correct this omission by restituting the expected segment
  – wrap your editorial addition in the element `<supplied>`, using the mandatory attribute `@reason` with the value `"omitted"` to distinguish this from a restoration of a lacuna (§5.5)
– before marking up an editorial addition, be sure that you have read and understood §6.2.5 and that editorial addition is the correct encoding in your case
– for example,
  – `dhanada-varuṇendrānta<supplied reason="omitted">ka</supplied>-samasya` (the *akṣara ka* was omitted by the scribe)
  – `tasya <supplied reason="omitted">tasya</supplied> tadā phalaM` (*tasya* should have been written twice, but one was omitted in haplography)
– omissions of a single character may, at your discretion,
  – be deemed erroneous and corrected in this way,
    – e.g. `dha<supplied reason="omitted">r</supplied>mma`
  – or be considered non-standard usage and treated as such
– small components (such as a superscript *r* or an *anusvāra*), which are expected to be present but cannot be made out in the original or a facsimile, might better be marked up as lost and restored (as per §5.5) unless you are certain that the cause is scribal omission, not damage to the support

6.2.5. Distinguishing correction from deletion and addition

– in some cases it may not be immediately obvious whether a certain editorial intervention is a case of correction (and thus requires a `<choice>` with `<sic>` and `<corr>`) or a case of suppression/restitution (and thus requires `<surplus>` or `<supplied reason="omitted">` respectively)
  – the nature of intervention must always be considered on the level of phonemes, and not on that of characters or glyph components in the original script or the transliteration
  – therefore, all of the following situations require correction and cannot be handled by means of suppression or restitution
– 1. quite obviously, **if the presence or absence of a stroke** (glyph component) **changes one character to another** (as with uppercase Latin F and E), then the suppression or restitution of that stroke is in fact a substitution of one character by another; thus:
  – in a script where *ka* differs from *ra* only in the presence of a cross-stroke in the former, and the scribe erroneously engraved *lora* instead of *loka*, your rectification of that error is a correction of *r* to *k*: `lo<choice><sic>r</sic><corr>k</corr></choice>a`
    – and not an editorial restitution (of the cross-stroke), even though the scribe's physical error was the omission of a glyph component
  – in a script where *śa* differs from *ga* only in the presence of a cross-stroke in the former, and the scribe erroneously engraved *śuṇa* instead of *guṇa*, your rectification of that error is a correction of *ś* to *g*: `<choice><sic>ś</sic><corr>g</corr></choice>uṇa`
    – and not an editorial suppression (of the cross-stroke), even though the scribe's physical error was the engraving of a superfluous glyph component

- 2. analogously to the above, but perhaps less self-evidently, **a vowel marker** added to a consonant character in most of the scripts we work with *changes* the inherent *a* of that character to a different vowel (rather than adding a vowel to a standalone consonant), and therefore the restitution of a single omitted vowel marker and the suppression of a single superfluous vowel marker are cases of editorial correction, and not of restitution/suppression; thus,
  - if the scribe engraved *mahāraja* and you correct this to *mahārāja*, your rectification of that error is a correction of *a* to *ā*: `mahār<choice><sic>a</sic><corr>ā</corr></choice>ja`
    - and not an editorial restitution (of the vowel marker), even though the scribe's physical error was the omission of a glyph component
    - note that the encoding `mahār<supplied reason="omitted">ā</supplied>ja` is not incorrect, only inappropriate in this situation, as it encodes the fact that *mahārja* was engraved, which you correct to *mahārāja*
  - if the scribe engraved *viditim* and you correct this to *viditam*, your rectification of that error is a correction of *i* to *a*: `vidit<choice><sic>i</sic><corr>a</corr></choice>m`
    - and not an editorial suppression (of the vowel marker), even though the scribe's physical error was the engraving of a superfluous glyph component
    - note that the encoding `vidit<surplus>i</surplus>m` is not incorrect, only inappropriate in this situation, as it encodes the fact your intended correction is *viditm*
- 3. conversely, **digraphs in our transliteration system** represent a single phoneme, so the correction of a digraph (e.g. *th*) to a single character employed in that digraph (e.g. *t*) or vice versa (e.g. correcting *t* to *th*) are cases of substitution, not suppression or restitution; thus,
  - if the scribe engraved *sukhya* and you correct this to *saukhya*, your rectification of that error is a correction of *u* to *au*: `s<choice><sic>u</sic><corr>au</corr></choice>khya`
    - and not an editorial restitution (of *a*), even though in transliteration you are only adding a single character
  - if the scribe engraved *utphannasya* and you correct this to *utpannasya*, your rectification of that error is a correction of *ph* to *p*: `ut<choice><sic>ph</sic><corr>p</corr></choice>annasya`
    - and not an editorial suppression (of *h*), even though in transliteration you are only removing a single character

6.2.6. Good practice in correction

- **the size of segments** flagged as scribal errors or corrected should normally be kept to a minimum (i.e. restricted to the affected transliteration characters)
  - however, to **avoid non-essential complexity**, feel free to use a single set of tags on a chunk of text that contains several errors along with correct characters
  - e.g. `mahārāj<choice><sic>adhijājā</sic><corr>ādhirāja</corr></choice>`
  - rather than the meticulous markup:
    `mahārāj<choice><sic>a</sic><corr>ā</corr></choice>dhi<choice><sic>j</sic><corr>r</corr></choice>āj<choice><sic>ā</sic><corr>a</corr></choice>`
- in the **orthography** of your editorial corrections, attempt to
  - respect the orthography and, if applicable, the language usage of the rest of the document, e.g.
    - correct *karppa* to *karmma* (rather than fully standard *karma*) if the inscription normally doubles nasals after *r*
    - if a text consistently uses *upadhmānīya* or *jihvāmūlīya*, and your correction involves the restitution of an omitted *visarga* in a phonemic context that would call for one of these forms, then supply *upadhmānīya* or *jihvāmūlīya* instead of a regular *visarga*,
      - e.g. `sarvva-rājocchettu<supplied reason="omitted">ḫ</supplied> pṛthivyām apratirathasya`
  - presuppose a plausible minimum of scribal error, e.g.
    - correct *karpa* to *karma* (rather than an expected *karmma*), assuming the engraver made the simple mistake of inscribing *p* for *m* (rather than the complex mistake of inscribing *p* for *mm*)

  - correct *viṅgati* to *viṅśati* (rather than fully standard *vimśati*, since engraving *ṅga* in place of *ṅśa* is a very plausible mistake, while engraving *ṅga* in place of *śa* and simultaneously omitting an *anusvāra* is not)
  - but correct *viśati* to *vimśati* (even if the text tends to use *ṅś* elsewhere, since omitting an *anusvāra* is a much more plausible mistake than engraving *śa* instead of *ṅśa*)
- should you feel the need, feel free to add normalisation on top of a correction (§6.3.3)

6.3. Encoding Normalisation

6.3.1. Flagging non-standard usage

- **to flag non-standard text** without normalisation, wrap the relevant characters with the element `<orig>`
- for example,
  - `dine <orig>Āśvoja</orig>-śuklasya` (*Āśvayuja* or *'śvayuja* is expected)
  - `sahasrā<orig>n</orig>i` (*sahasrāṇi* is expected)

6.3.2. Normalising non-standard usage

- **to normalise** usage **by substitution**,
  - flag the original text with `<orig>` as above,
  - add the normalised alternative directly after this, wrapped in the element `<reg>`
  - and wrap both these elements in the element `<choice>`
- for example, `e<choice><orig>ś</orig><reg>ṣ</reg></choice>a` (*eśa* normalised to *eṣa*)
- when normalising by substitution, keep in mind that it must always be possible to produce the received text by ignoring the segment tagged with `<reg>` and, vice versa, to produce the normalised text by ignoring the segment tagged with `<orig>`
  - thus, when for instance encoding a normalisation of *yathāruha* to *yathārhaṁ*, each of the following are **incorrect**:
    - `<choice><orig>yathāruha</orig><reg>yathārha</reg></choice>ṁ` (this encodes a normalisation of *yathāruhaṁ* to *yathārhaṁ*)
    - `<choice><orig>yathāruha</orig><reg>rhaṁ</reg></choice>` (this encodes a normalisation of *yathāruha* to *rhaṁ*)
    - `yathār<choice><orig>uha</orig><reg>rhaṁ</reg></choice>` (this encodes a normalisation of *yathāruha* to *yathārrhaṁ*)

6.3.3. Nesting normalisation and correction

- should you find it necessary to do so, it is acceptable to use error markup (including flagging, deletion, correction and insertion) within the markup for non-standard usage (including flagging and normalisation)
  - for example, either of the following methods may be used to mark up the fact that the word *mahārajñaḥ* is non-standard (the standard genitive being *mahārājasya*) and also contains a mistake (as the "proper" non-standard form would be *mahārājñaḥ*)
    - `<orig>mahār<choice><sic>a</sic><corr>ā</corr></choice>jñaḥ</orig>`
    - `<orig>mahār<sic>a</sic>jñaḥ</orig>`
- however, you should avoid nesting in all other combinations, i.e.
  - do not nest a correction within another correction
  - do not nest a normalisation within a correction
  - do not nest a normalisation within another normalisation
- in situations where you feel that a received non-standard form is the result of two successive stages of non-standard alteration (e.g. non-standard morphology written with non-standard orthography), we recommend one of the following strategies
  - encode only the received form (as `<orig>`) and the ultimate normalisation (as `<reg>`), and record your ideas about an intermediate stage in an apparatus note (§9.1.7)

- e.g. `pāñcavarṣ<choice><orig>aI</orig><reg>i</reg></choice>kā`, accompanied by an apparatus note explaining that the received form is probably a non-standard way of writing the form **pāñcavarṣayikā*, itself of non-standard derivation
- or encode the intermediate stage as the correction of an error in the received text, and encode a normalisation with that correction nested inside it
  - e.g. `pāñcavarṣ<choice><orig>a<choice><sic>I</sic><corr>yi</corr></choice></orig><reg>i</reg></choice>kā`

6.3.4. Good practice in normalisation

- it is recommended that your normalisations should conform to the **orthographic style** of the rest of the document in details that you would not normalise elsewhere
  - e.g. normalise *varnna* to *varṇṇa* (rather than fully standard *varṇa*) if the inscription normally doubles nasals after *r*
- the **size of segments** flagged as non-standard or normalised should generally be whatever you deem to be a reasonable minimum to which the non-standard feature can be localised
  - when non-standard orthography manifests in a single character or short character sequence, it is sufficient to tag that character or sequence, but you may also include its immediate phonemic context in the following cases:
    - if it is not possible to apply the desired tag to just the affected character (see the points below on the difficulties of orthographic normalisation); or
    - if you feel that including additional characters in the tag is useful for highlighting the nature of the non-standard feature
  - "immediate phonemic context" is not an objectively defined entity and shall be judged on a case by case basis, but will generally consist of
    - adjacent characters representing phonemes that would normally determine or influence the nature of the non-standard one
    - the whole string of transliterated characters corresponding to a single complex character of the original that includes the non-standard feature, when this seems to be the most straightforward and convenient way of highlighting a non-standard feature
  - choosing the size of a segment to flag or normalise is thus not a wholly objective choice, and the choice has very little ultimate effect on our corpus so long as the text as received is faithfully reproduced in your encoding along with any normalisation you add
- to **avoid non-essential complexity**, feel free to use a single set of tags on a chunk of text that contains several non-standard features among standard text
  - in particular, for the stock admonitory stanzas cited in land grants, whose error-rate is often much higher than in the remainder of an inscription, feel free to include the contents of an entire `<l>` element in a single substitution, e.g.
  - `<l><choice><orig>sva-datnā para-datnā vvā</orig><reg>sva-dattāṁ para-dattāṁ vā</reg></choice></l>`
- as indicated in §6.1.2 above, there is no encoding method dedicated to the **suppression or restitution of individual characters in the framework of normalisation**
  - in order to prevent anomalies in display, we will, moreover, avoid using normalisation by substitution in such a way that one of the children of `<choice>` (i.e. `<orig>` or `<reg>`) is empty; therefore,
  - when non-standard orthography manifests as **the presence of an alternative character** (e.g. *nikki* instead of *nīkki*; *phālguṇa* instead of *phālguna*), then there is no difficulty in limiting your tags to the affected character
    - whether you only flag it, e.g.
      - `n<orig>i</orig>kki`;
      - `phālgu<orig>ṇ</orig>a`;
    - or normalise it, e.g.
      - `n<choice><orig>i</orig><reg>ī</reg></choice>kki`

- phālgu<choice><orig>ṇ</orig><reg>n</reg></choice>a
- when non-standard orthography manifests as the **presence of a superfluous character** (e.g. *eṉṉ eḻuttu* instead of *eṉ eḻuttu*; *saṁmvaT* instead of *samvaT*)
  - then flagging can be limited to the superfluous character without difficulty, e.g.
    - e<orig>ṉ</orig>ṉ eḻuttu
    - sa<orig>ṁ</orig>mvat
  - but when normalising by substitution, you must extend the markup to the immediate phonemic context in order to avoid the creation of an empty <reg> element, e.g.
    - e<choice><orig>ṉṉ</orig><reg>ṉ</reg></choice> eḻuttu
    - sa<choice><orig>ṁm</orig><reg>m</reg></choice>vat
- when non-standard orthography manifests as the **absence of an expected character** (e.g. *satva* instead of *sattva*; *qəcu* instead of *qəñcu*; *umulata* instead of *umulat ta*), then the immediate phonemic context must always be included in the markup in order to avoid the creation of an empty <orig> element
  - in flagging, e.g.
    - sa<orig>tv</orig>a
    - qə<orig>c</orig>u
    - umula<orig>t</orig>a
  - and in normalisation, e.g.
    - sa<choice><orig>tv</orig><reg>ttv</reg></choice>a
    - qə<choice><orig>c</orig><reg>ñc</reg></choice>u
    - sakuli<choice><orig>li</orig><reg>liṁ</reg></choice> ḍayəḥ
    - umula<choice><orig>t</orig><reg>t t</reg></choice>a
      - the last example also shows that a word break rendered invisible by the substandard spelling may be made visible and marked by a space in the normalized reading

6.3.5. How non-standard is non-standard?

- this subsection offers some general guidance on the level of editorial attention that various kinds of non-standard features merit
  - whether you should ignore a specific phenomenon, flag it as non-standard, or normalise it by substitution should always be judged on an individual basis, and no objective and universal criteria can be established for such a decision
  - in addition to the general guidance below, see Appendix F for some specific phenomena in specific languages
- (near-)**universal features of inscriptional orthography** prevalent throughout South and Southeast Asia, or throughout a particular region
  - should generally be ignored or, if considered important in a particular instance, preferably only flagged and not normalised
  - such features include for instance:
    - the doubling of plosives, nasals and glides after an *r* (e.g. *dharmma* for *dharma*)
    - the use of an *anusvāra* instead of the class nasal or vice versa (e.g. *maṁtra* for *mantra*; *kin tu* for *kiṁ tu*)
- less than universal, but still **common features of inscriptional orthography**
  - may be ignored or flagged depending on how widespread they are in a subcorpus (or even in a single text), but should not as a rule be normalised
  - such features include for instance:
    - the doubling of consonants in certain conjuncts that do not begin with *r* (e.g. *puttra* for *putra*; *sattya* for *satya*)
    - the exchange of a consonant for a phonetically similar one (e.g. *muṇi* for *muni*)
    - infidelity to the correct length of vowels in words borrowed from Sanskrit, in languages where inconsistency in spelling of vowel-length is rampant (e.g. *bhima* for *bhīma*)
- **non-orthographic deviations from standard language**
  - should normally be at least flagged and preferably also normalised
  - such features include for instance:

- non-standard or substandard grammar (e.g. *rājasya* for *rājñaḥ*; *kṛtedam* for *kṛtam idam*; *sā gataḥ* for *sā gatā*)
- presumable non-standard sandhi (e.g. *anugrahāya-m udaka-pūrvveṇa*; *paṁca-s-triṁśottaratame*)
  - see also TG §2.6.2 on the use of hyphens in non-standard sandhi
- presumable hyper-Sanskritisation (e.g. *dattvā* instead of *dattā*; *rakṣya* instead of *rakṣa*; *prārk-kriyamāṇaka* instead of *prāk-kriyamāṇaka*)

6.3.6. Supplying punctuation

- while original punctuation marks present in the text must always be transliterated and encoded as per §4.2.4, editorial punctuation marks must never be added silently to a text
  - emphatically, the silent addition of punctuation marks for the segmentation of verse into stanzas and half-stanzas must be avoided, since verse is always segmented by the encoding of intrinsic structure (§2.3)
- however, in some circumstances you may feel the need to supply editorial punctuation, and this is possible and permitted so long as editorial punctuation is clearly marked up as supplied
- editorial punctuation may be particularly useful in the following circumstances:
  - for semantic segmentation of long paragraphs into sentences or other semantic units, when no original punctuation is present, and you are not creating separate semantic paragraphs or anonymous blocks (§2.2) for each unit
    - supplying editorial punctuation at the end of a paragraph or block is, however, pointless
  - in lists (e.g. lists of donees), to mark the end of each list item
  - and especially if in either of the above circumstances the original does use punctuation marks, but does so inconsistently (i.e. only after some sentences or list items), because in this case the most logical assumption is that the lack of punctuation marks after certain items is a scribal omission
- to encode supplied punctuation,
  - use the transliteration character . (period, full stop) as per TG §4.2.1, but do not add a `<g>` tag around this character as you would for original punctuation (EGD §4.2.4)
    - this is to express the fact that this punctuation character is an abstract one, without any assertion of its physical appearance
  - and wrap the . in `<supplied reason="subaudible">`[34]

Example 6.3.6.A: supplied punctuation at the end of a sentence

```
<p> ... anumantavyo varddhanīyaś ca<supplied reason="subaudible">.</supplied> yo vājñānād ...
</p>
```

Example 6.3.6.B: supplied punctuation in a list with sporadic original punctuation

```
<p> ... Āruvaśarmmaṇa Ekkaṁśaḥ<supplied reason="subaudible">.</supplied> puna vedaśarmmaṇa
Ekkaṁśaḥ<supplied reason="subaudible">.</supplied> ... jakkiśarmmaṇa Ekkaṁśaḥ<supplied
reason="subaudible">.</supplied> ... vebaśarmmaṇa Ekkaṁśaḥ<g type="dandaPlain">.</g> ...
sarvvaśarmmaṇa ... </p>
```

➢ the original punctuation mark is tagged with `<g>`

6.3.7. Automated normalisation

- some specific cases of normalisation will be automated in our workflow, so certain characters in your transliteration will be converted to markup
- **editorial long vowels in Dravidian languages** where the script does not distinguish short and long *e* and *o*
  - as per TG §3.2, the transliterated characters *ē* and *ō* will be automatically marked up as normalised, i.e. that *e* or *o* were originally inscribed, but these represent long vowels, e.g.
    - `<choice><orig>e</orig><reg>ē</reg></choice>`
- **explicit short vowels in Sanskrit loanwords** where a long vowel is expected

[34] The rationale behind the choice of attribute value is that punctuation is implied by the semantic context, so in a way, a punctuation mark is "subaudible" to the native or informed reader.

  - as per TG §3.3.7, the transliterated characters *ă*, *ĭ* or *ŭ* will be automatically marked up as short in the original and normalised to their long equivalents, e.g.
    - `<choice><orig>a</orig><reg>ā</reg></choice>`
- **editorial** *avagraha*s
  - as per TG §2.6.3, any *avagraha* (i.e. ’ [right single quote] or ' [plain apostrophe] followed by an alphabetic character) found within the `<div type="edition">` will be assumed by default to be non-original and automatically marked up as `<supplied reason="subaudible">'</supplied>`[35]
  - original *avagraha*s transliterated as ’! will not be auto-tagged in this way, but the exclamation mark will be removed automatically

[35] The rationale behind the choice of attribute value is that the presence of an *avagraha* is implied by the phonetic and lexical context, so in a way, an *avagraha* is “subaudible” to the native or informed reader. Alternative encoding choices would imply that the scribe made an error or used non-standard language, which is not the case.

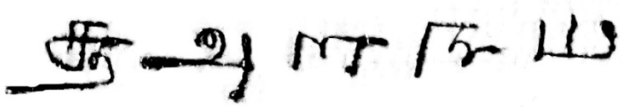
7. Encoding Additional Information in the Edition

7.1. Numeral Values

7.1.1. Generic numeral markup

– all numbers recorded in numeral signs in the original inscription must be mandatorily wrapped in the element `<num>`
  – when a glyph that would normally be a numeral sign is used in a function other than to represent a number (such as the glyph normally meaning 1, occasionally used as an auspicious opening mark), then the `<num>` tag must not be added to it (§4.2.7)
  – this tag does not replace the `<g>` tags discussed in §4.2.2 for numeral signs transliterated by something other than a single Unicode character, but must be used in addition to (and outside) those
– the element `<num>` must, as a rule, have the attribute `@value`, recording in a machine-readable form the final value of the entire number within the element
– **fractions** shall be represented here as decimal fractions, never dropping the 0 before fractions smaller than 1 and always using a decimal point, not a different decimal marker, e.g. `<num value="0.5">`
  – round the value to three digits after the decimal point for any fractions that would require a longer sequence of digits, e.g. encode ⅓ as `<num value="0.333">`
– **some examples of numerals with full markup**:
  – the number three denoted by a numeral character: `<num value="3">3</num>`
  – the number three denoted by three vertical bars in a Cambodian inscription: `<num value="3"><g type="numeral">III</g></num>`
  – one hundred and twenty-three, written in place value notation: `<num value="123">123</num>`
  – one hundred and twenty-three, written in additive notation with a sign for 100, one for 20 and one for 3: `<num value="123"><g type="numeral">100</g> <g type="numeral">20</g> 3</num>`
  – ninety written as the Khmer digit 80 and digit 10: `<num value="90"><g type="numeral">80</g> <g type="numeral">10</g></num>`
  – one thousand, written with one sign meaning "1000": `<num value="1000"><g type="numeral">1000</g></num>`
  – one half, written as a single character: `<num value="0.5">½</num>`
  – one eighth, written as a single character: `<num value="0.125"><g type="numeral">1/8</g></num>`
  – three and one third, written as a digit 3 and one character standing for "one third": `<num value="3.333">3 ⅓</num>`
  – three and one eighth, written as a digit 3 and one character standing for "one eighth": `<num value="3.125">3 <g type="numeral">1/8</g></num>`

Example 7.1.1.A: complex Tamil numeral



➢ the numeral 1830 is written as 1000 (plus) 8 (times) 100 (plus) 3 (times) 10

```
<num value="1830"><g type="numeral">1000</g> 8 <g type="numeral">100</g> 3 <g type="numeral">10</g></num>
```

7.1.2. Difficulties in reading numbers

– problems with reading numeral signs (e.g. lacunae, unclear and ambiguous readings) can and must be marked up in the same way as other reading difficulties (§5)
– if a numeral sign is **unclear or ambiguous**, the applicable **tags** go **outside** any `<g>` elements applied to numeral characters, but they go **inside** the `<num>` element that wraps numbers as a whole

– numbers whose reading is problematic will usually not have a definite value that you can encode in the `<num>` element; to deal with this problem, choose one of the following methods[36]
  – **1.** if you can establish a **range** that covers the **possible values** of a problematic numeral:
    – instead of the attribute `@value`, use `@atLeast` and `@atMost` in the `<num>` element to record the lowest and highest possible value of the number as a whole
      – e.g. for three digits in place value notation, where the first two digits are 1 and 0, and the last digit is illegible: `<num atLeast="101" atMost="109">10<gap reason="lost" quantity="1" unit="character"/></num>`
    – because of its relative simplicity, this method is also recommended for situations where only some figures in a relatively limited range are possible
      – e.g. for an unclear numeral sign that may be 80 or 90 with equal likelihood, you may use `<num atLeast="80" atMost="90"><choice><unclear><g type="numeral">80</g></unclear><unclear><g type="numeral">90</g></unclear></choice></num>` even though the values 81 to 89 are not possible
  – **2.** if you can establish a **single tentative value** for a problematic numeral:
    – encode this in the attribute `@value` and add the attribute `@cert` with the value `"low"` to flag the value as tentative
    – e.g. for three digits in place value notation, where the first two digits are 2 and 4, and the last digit seems to be 6: `<num value="246" cert="low">24<unclear>6</unclear></num>`
  – **3.** the above methods may be combined to encode a **range of tentative values**
    – e.g. for three digits in place value notation, where the first two digits are probably 1 and 0, and the last digit is illegible: `<num atLeast="101" atMost="109" cert="low"><unclear>10</unclear><gap reason="lost" quantity="1" unit="character"/></num>`
  – **4.** if none of the above seems adequate for a partially legible numeral,[37] or if a numeral is **wholly lost or illegible** (yet you are certain it was a numeral):
    – use the `<num>` element without `@value` around the partial reading or the lacuna
    – e.g. for one lost/illegible numeral character: `<num><gap reason="lost" quantity="1" unit="character"/></num>`

7.1.3. Editorial intervention and numerals

– occasionally, an editor may be able to restore a lost number, or even emend an incorrectly inscribed one, e.g. on the basis of the number being also written out in words
– tags for **editorial restoration** may be used inside or outside the `<num>` element depending on the scope of the restoration, but they must never be inside any `<g>` elements applied to numeral signs
  – a longer stretch of restored text may freely include both text and a numeral
  – the `@value` attribute of the `<num>` element should reflect only the restored value
– tags for editorial **correction** must be **outside** the `<num>` elements, which must separately encode both the pre- and post-correction number
  – e.g. `<choice><sic><num value="6">6</num></sic><corr><num value="7">7</num></corr></choice>`

7.1.4. Numbers expressed in words

– although the TEI element `<num>` may be used to tag anything that has a numerical meaning, our project policy shall be never to use this element for numbers written out in alphabetic characters

---

[36] Whichever method you use, possible values and their relative probabilities may be elaborated in your commentary to the edition.

[37] The EpiDoc Guidelines offer a further method for dealing with partly lost numerals whose range of possible values is not sequential (http://www.stoa.org/epidoc/gl/latest/trans-numnoncongruent.html). We discourage the use of this method because we do not foresee that our project would benefit from the increased accuracy to an extent that would justify the complexity of the markup involved.

– thus, neither numbers spelled out in words (such as *ekaḥ* and *aṣṭottaraśataṁ*), nor chronograms (*bhūtasaṁkhyā*, *candrasengkala/sengkalan*) shall be tagged in this way
– the rationale behind this choice is that complex numbers expressed in words may be distributed over stretches that also contain non-numeral words and that the interpretation of numerals expressed as phrases, especially chronograms, is sometimes ambiguous: the complexity of the markup required to encode such cases rigorously dwarfs the anticipated advantages of having such numbers encoded

7.2. Tagging Language in the Edition

– this section concerns encoding language within the edition
  – see §10.3 for wider applications of language encoding
  – see §10.3.3 for specific instructions applicable in other parts of your XML file
– the language(s) used in an inscription must be specified in your metadata
– the attribute `@xml:lang` may be attached to elements to encode the language of their contents; see §10.3.1 for details
  – normally, the edition division of your XML file must mandatorily carry this attribute, e.g. `<div type="edition" xml:lang="san-Latn">`
– wherever language change is concomitant with script change, the script should not be marked up separately (as per §7.5.4)
  – instead, your metadata should specify the scripts used for each language in your inscription

7.2.1. Inscriptions consisting of sections in different languages

– when two (or more) **major sections** of an inscription are in two (or more) different languages, consider the degree to which these sections are independent of each other
– if the inscription may be perceived as **a single, coherent text** with one or more language shifts,
  – select a primary language to encode as the `@xml:lang` of the edition division
  – apply `@xml:lang` to each of the block-level elements (viz. `<p>`, `<lg>` or `<ab>`) containing text in a language other than the primary one
– if one section **cannot be perceived as an integral continuation** of the other, because they are visually clearly distinct *and* semantically unconnected with no straightforward order in which they should be read (e.g. they convey the same message in two languages, or cover unrelated topics)
  – first, consider if it would be better to edit the inscription as two separate texts
  – if that is not feasible, then
    – encode the language-based sections as textpart divisions (§3.4)
    – add `@xml:lang` to each corresponding `<div type="textpart">` element
    – in this case only, the edition division should not carry the attribute `@xml:lang`
      – note, however, that in an inscription consisting of textparts in the same language, the language must still be encoded for the edition division, not separately for the textparts

7.2.2. Inscriptions containing foreign words or phrases

– if an inscription includes **foreign words** which are **followed by translations** into a local language (glosses),
  – use the element `<term>` to wrap each foreign word, and
  – use `<gloss>` to wrap each translation into the default language of the inscription
  – the attribute `@xml:lang` is not required in this scenario
– if a different language applies to **isolated words or phrases** of an inscription,
  – use `<foreign>` to wrap it and apply `@xml:lang` to that element
  – loanwords and foreign names should not, as a rule, be marked up as being in a different language, but do tag
    – complete sentences using vocabulary *and* morphology/syntax foreign to the default language
    – Sanskrit compounds which are not established as loanwords (i.e. do not appear in a standard dictionary of the local language such as the *Old Javanese-English Dictionary* or the *Madras Tamil Lexicon*)

– in less clear-cut cases, use your own discretion to decide whether or not to tag a segment as foreign

7.3. Abbreviations

– if your text includes abbreviations, it is recommended that you wrap these in the element `<abbr>` to flag them for computer processing
  – for the time being, we shall not encode expansions or resolutions for any abbreviations in our texts, though TEI and EpiDoc offer methods to do so
  – we may, at a later stage, automatically or semi-automatically add resolutions to the abbreviations flagged in this way

7.4. Optional Encoding of Semantic Features

Besides the tags prescribed in other sections of this Guide, TEI offers the possibility of using many others to encode additional semantic information in a text. Such tags, whose use is **optional and not recommended at this stage of the project**, enable the creation of indexes, for instance of all the persons or places mentioned in a (sub-)corpus with an exhaustive list of occurrences.

As adding such tags to XML editions renders the files less legible, we recommend postponing the application of such tags as long as two conditions are not fulfilled: (1) the entire (sub-)corpus has been encoded according to the guidelines exposed in other sections of this Guide and, (2) the choice of such tags has been determined after ripe reflection and in response to concrete aims of your (or the whole project's) research. Any such tagging that you do choose to apply should be implemented in accordance with the workflow of your task-force as determined by and in consultation with the PI of your task-force.

At the time this version of the Encoding Guide is released, only TF-A has started reflecting on which tags, types and subtypes could be used to answer specific research questions. The preliminary results of these reflections are given below, so that members of other task-forces can look at examples of what is possible and to stimulate a process of reflection on what tagging might be implemented in other corpora. The present section is thus provisional and mainly illustrative. It will be developed in a future version of this guide.

7.4.1. Personal names

– personal names may be tagged with the element `<persName>`
  – this element can be used to encode a complex name, tagging individually all elements of a personal name
– a first categorisation can be effected with attribute `@type`
  – propositions for the value of `@type`:
    – `"divine"`
    – `"human"`
    – `"personification"`
– subcategorisation is effected with `@subtype`, which may only be used if `@type` is also present
  – propositions for the value of `@subtype`:
    – `"coronation"` (Rājarāja, Rājendra, …)
    – `"sobriquet"` (*biruda*)
    – `"title"` (*pōttaraiyar*, *(kōp)parakēcarivarman̲* / *(kō)rājakēcarivarman̲*)
    – `"other"` (pre-coronation name, e.g. Arumo̲li, Arumo̲livarman̲)
– to indicate that the name in question is an alternative of some other name (perceived as a standard form), follow instructions in §10.6.3

Example 7.4.1.A: encoding a complex personal name

```
<persName type="human" subtype="sobriquet">caturummallan̲</persName> <persName type="human"
subtype="sobriquet">kuṇaparan̲</persName> <persName type="human"
subtype="coronation">mayēntira</persName>-p-<persName type="human" subtype="title">pōtt-
arēcaru</persName>
```

7.4.2. Adding ranks and roles to names

– the element `<roleName>` can be used to encode a position in society like a rank or status (`@type`) and associate it with a role (`@subtype`) in the transaction recorded
  – the element `<roleName>` is to be nested inside the element `<persName>`
  – propositions for the value of the attribute `@type`:
    – `"king"`
    – `"subordinateRuler"` (e.g. *pallavaraiyaṉ*)
    – `"landlord"` (e.g. *uṭaiyar*, *kiḻavar*)
    – `"godLegalEntity"` (e.g. *uṭaiyar*)
    – `"priest"`
    – `"brahmin"`
    – `"monk"`
    – `"merchant"` (e.g. *nakarattār*)
    – `"artisan"`
    – `"brahminDelegate"` (e.g. *sabhaiyār*, *sabhaiyōm*)
    – `"regionalDelegate"` (e.g. *nāṭṭār*, *nāṭṭōm*)
    – `"officer"` (e.g. temple officer, royal officer)
    – `"dancer"`
    – `"singer"`
    – `"peasant"`
    – `"shepherd"` (*maṉṟāṭi*)
    – `"unknown"` (this value is to be used when you do not know the rank/status of the person but want to encode a value for `@subtype`)
  – propositions for the value of the attribute `@subtype`:
    – `"donor"`
    – `"donee"`
    – `"founder"` (of a temple or a monastery)
    – `"administrator"` (overseer of donation; e.g. the one who makes sure that the in-charge of a donation supplies what he has to supply).
    – `"inChargeDonation"` (e.g. the one who has to supply oil every day)
    – `"witness"`
    – `"orderIssuer"`
    – `"orderAddressee"`
    – `"auditor"` (controller of transaction)
    – `"beneficiaryMerit"` (e.g. transfer of merit; donation “on behalf of”, “in the name of”)
    – `"commemoratedPerson"` (e.g. “in the honour of (a deceased warrior)”)
    – `"scribe"` (exact role undetermined)
    – `"composer"` (i.e. author of the text or part of the text; e.g. poet of the Sanskrit eulogy).
    – `"handwriter"` (i.e. the one writing in chalk on the plate/stone for the engraver)
    – `"engraver"` (i.e. the artisan who engraved the text on the support)
    – `"sealer/solderer"` (i.e. the one who fabricated/sealed/soldered the seal)

Example 7.4.2.A: encoding ranks and roles

```
<persName type="human" subtype="coronation"><roleName type="king"
subtype="donor">Mahendravarman</roleName></persName> gave 25 gold coins to the <persName
type="divine" subtype ="standard"><roleName type="godTemple"
subtype="donee">Śiva</roleName></persName> of Tillaisthānam so that <roleName type="shepherds"
subtype="inChargeDonation">the shepherds</roleName> supply daily oil for a lamp for <roleName
type="king" subtype="beneficiaryMerit">his father</roleName> under the supervision of <roleName
type="priest" subtype="trustee">the priests</roleName> of the temple
```

7.4.3. Place names

– place names (including territorial and administrative divisions as well as built places) can be encoded using the element `<placeName>`
  – we recommend using the attribute `@type` using the values `"territorialDivision"` and `"builtPlace"`
– places can be described more precisely with the attribute `@subtype`, for which the following values have been proposed by the TF-A:
  – `@subtype` for territorial and administrative divisions:
    – `"province"` (*kōṭṭam*, *rāṣṭra*, *maṇḍala*, *vaḷanāṭu*, etc.)
    – `"district"` (*viṣaya*, *nāṭu*, *kūṟṟam*)
    – `"site"` (town, village)
    – `"sitePart"` (e.g. quarter, hamlet, *cēri*)
– `@subtype` for built places:
    – `"temple"`
    – `"shrine"` (e.g. for a secondary shrine in a temple complex)
    – `"monastery"` (e.g. *vihāra*, *maṭha*)
    – `"feedingHall"` (*cālai*, Skt. *śālā*, mess for devotee pilgrims)
    – `"tank"` (artificial)
    – `"pavillion"` (*maṇḍapa*)
    – `"garden"` (*nandavaṉam*)
– to indicate that the name in question is an alternative of some other name (perceived as a standard form), follow instructions in §10.6.3

Example 7.4.3.A: encoding place names

```
in <placeName type="territorialDivision" subtype="village">Cārukūr</placeName> in the <placeName
type="territorialDivision" subtype="district">Āṭaiyārunāṭu = province</placeName> and the
<placeName type="territorialDivision" subtype="province">Paṭuvūrkōṭṭam = district</placeName>
in <placeName type="builtPlace" subtype="shrine">the shrine of the Goddess</placeName> in
<placeName type="builtPlace" subtype="temple">the temple of Mahādeva</placeName> at <placeName
type="territorialDivision" subtype="village">Tillaisthānam</placeName>
```

7.4.4. Measurements

– when necessary, the tag `<measure>` allows encoding references of quantity
  – use the attribute `@type` to record the typology used to measure, e.g. volume, weight, currency…
  – usually, measure requires encoding the quantity, the unit used, and possibly the commodity measured
  – it can be done using the attributes: `@unit`, `@quantity` and `@commodity`
– `@unit` indicates the unit used for the measurement expressed by its standard symbol, e.g. cm, m, ml, km, in …
– `@quantity` records a numeric value
– `@commodity` for the measured substance
– the numeral values for measurement should be encoded as per §7.1

Example 7.4.4.A: encoding place names

```
<measure unit="kaḻañcu" quantity="100" commodity="gold">nūṟṟu-k kaḻañcu poṉṉum</measure> <measure
unit="kāṭi" quantity="28" commodity="paddy">Irupatteṇ kāṭi nellum</measure> kuṭuttēṉ</p>
```

➢ "I have given one hundred *kaḻañcu* of gold and twenty-eight *kāṭi* of paddy"

7.4.5. Tagged semantic features interacting with text or markup

– this subsection applies to each of the elements `<persName>`, `<placeName>`, `<roleName>` and `<measure>`
– for semantic features with **ends or beginnings merged in sandhi** to an adjacent word, tag the entire name including the character(s) partly belonging to adjacent words

– e.g. `<persName type="human" subtype="coronation">siṁhavarmmā</persName>dhipāt`
– **empty elements** (such as `<lb/>` and `<milestone/>`) may be freely included within tags for semantic features
– **phrase-level elements overlapping with a semantic tag** shall be split into two segments, prioritising the semantic tag
  – e.g. `<persName type="human" subtype="coronation">siṁhavar<unclear>mmā</unclear></persName><unclear>dhi</unclear>pāt`
– in the case of a **semantic tag interrupted by** the start or end of a **block-level element**, the semantic tag must be split into two segments, prioritising the block-level element
  – in this case, the two parts of the semantic tag will have to be linked as follows:
  – add an `@xml:id` (§10.6.4) to both parts, with a value composed of
    – the filename followed by an underscore
    – followed by "name", "place", "role" or "measure" as applicable
    – followed by the number "1" (or the next higher number, should a single document contain more than one instance of split tags of the same nature)
    – followed by an uppercase A for the first part and an uppercase B for the second part
  – to link the two parts, add `@next` to the first part and `@prev` to the second part,
    – with the XML ID of the *other* part (prefixed with a # character) as the value of these attributes

Example 7.4.5.A: personal name split across block-level containers

```
<l> ...
<persName type="human" subtype="sobriquet" xml:id="Pallava00001_name1A"
next="#Pallava00001_name1B">guṇa</persName></l>
<l><persName type="human" subtype="sobriquet" xml:id="Pallava00001_name1B"
prev="#Pallava0001_name1A">bharaḥ</persName>...</l>
```

Example 7.4.5.B: personal name split across block-level containers

```
<l> ...
<persName type="human" subtype="sobriquet" xml:id="Pallava00001_name1A"
next="#Pallava00001_name1B">guṇa</persName></l>
<l><persName type="human" subtype="sobriquet" xml:id="Pallava00001_name1B"
prev="#Pallava0001_name1A">bharaḥ</persName>...</l>
```

7.5. Visual Features

7.5.1. The scope of visual features encoded in attributes

– the attributes described in this section for encoding visual features may be used with the following XML elements as the situation demands:
  – structural containers, i.e. textpart divisions (§3.4) and forme work (§3.3.5)
  – block-level containers, i.e. units of prose (§2.2) and verse (§2.3)
  – `<lb/>` elements,[38] when a localised feature applies to an entire physical line
  – the dedicated phrase-level tag `<hi>`,[39] when the scope of a visual feature is not a pre-existing element
– it may occasionally be necessary to use **multiple values of the attribute** `@rend` on one element
  – attributes cannot be iterated, so in such cases, add both (or all) applicable values within the same set of quote marks, separated by a space

[38] Though this element is not a container, by EpiDoc convention "Any rend or numbering attributes on ... lb refer to all text between the current and the following line-break" (http://www.stoa.org/epidoc/gl/latest/trans-linebreak.html.)

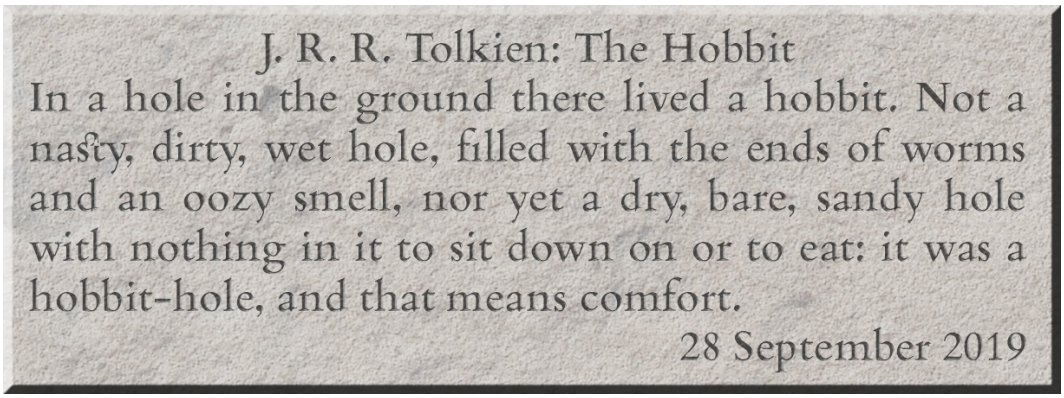
[39] This element is intended in TEI to encode highlighted text, defined as words or phrases graphically distinct from the surrounding text (https://www.tei-c.org/release/doc/tei-p5-doc/en/html/ref-hi.html). EpiDoc generalises the use of this element to characters with distinguishing graphical features regardless of whether or not these serve the purpose of highlighting (http://www.stoa.org/epidoc/gl/latest/trans-charactershighlighted.html).

– in order to reduce the complexity required for processing these, when using multiple values, please observe a strict order as follows
  – 1. directionality and orientation
  – 2. script
  – 3. lettering
  – e.g. `<lb n="01" rend="bt-rotated ornate"/>siddham`

7.5.2. Alignment

– we assume by default that the lines of an inscription are aligned to the left and more or less justified to the right margin
  – large-scale deviations from this pattern shall not be encoded in the markup but rather discussed in your description of the layout
– lines aligned differently than the majority of the lines in an inscription shall be encoded by adding the attribute `@style` to one of the following elements (but not to any other element):
  – `<div type="textpart">` to describe all lines in a textpart (§3.4)
  – `<lb/>` to describe individual lines (see also §7.5.1)
– **the permitted values of** `@style` for encoding alignment are as follows:
  – `"text-align: right"` for right-aligned text
  – `"text-align: center"` [note the mandatory US spelling] for centre-aligned text
  – `"text-align: left"` for left-aligned text
    – to be used only if the majority of the lines are aligned differently and this is mentioned in your layout description or encoded for the enclosing textpart division
  – `"text-align: justify"` for text conspicuously justified to both margins
    – to be used only if the content of a line would fit in a much narrower space and the creator of the inscription deliberately made it flush with both the left and right margin by increasing inter-word or inter-character spaces in a relatively regular manner (i.e. increased spaces are present between all or most separable words of the line, or between all or most characters as applicable)
    – spaces within a line that function as semantic segmentation are better encoded as such (§4.3.2) even if they also happen to serve the justification of a line

Example 7.5.2.A: encoding line alignment

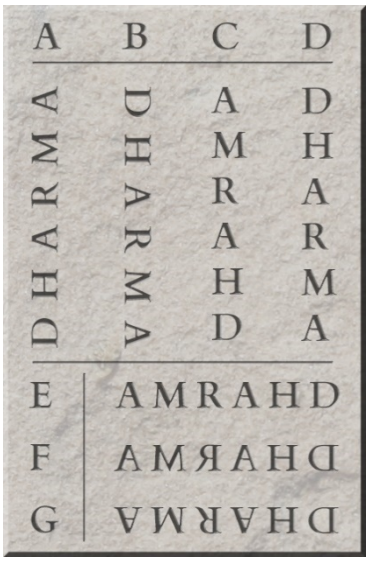



```
<lb n="1" style="text-align: center"/>J. R. R. Tolkien: The Hobbit
<lb n="2"/>In a hole in the ground...
...
<lb n="4"/>hobbit-hole...
<lb n="7" style="text-align: right"/>28 September 2019
```

7.5.3. Directionality and orientation

– the default writing mode for our inscriptions is horizontal top-to-bottom, which means that lines run horizontally (and left to right in the scripts we work with), while successive lines are placed one below the other

  - some inscriptions may be written throughout in a different writing more (e.g. vertically, with lines proceeding left to right; or horizontally, with lines proceeding left to right, but in a bottom to top order); if this is the case, it should be recorded in your metadata
- lines oriented differently than the majority of the lines in an inscription may be encoded by adding the attribute `@rend`[40] to one of the following elements (but not to any other element):
  - `<div type="textpart">` to describe all lines in a textpart (§3.4)
  - `<fw>` to describe the contents of a forme work item (§3.3.5)
  - `<lb/>` to describe individual lines (see also §7.5.1)
- **the permitted values of** `@rend` for encoding directionality and orientation are as follows:
  - `"bt-rotated"` – written vertically from bottom to top, with the tops of characters facing left (A in the illustration)
  - `"tb-rotated"` – written vertically from top to bottom, with the tops of characters facing right (B in the illustration)
  - `"bt-upright"` – written vertically from bottom to top, with the tops of characters facing upward (C in the illustration)
  - `"tb-upright"` – written vertically from top to bottom, with the tops of characters facing upward (D in the illustration)
  - `"rl-upright"` – written horizontally from right to left, with characters in their regular orientation (E in the illustration)
  - `"rl-flipped"` – written horizontally from right to left, with characters mirrored around their vertical axis as in true boustrophedon (F in the illustration)
  - `"rl-rotated"` – written horizontally from right to left, with the tops of characters facing downward (G in the illustration)



  - should you encounter any other combination of directionality and orientation (e.g. right to left), contact the authors and the XML-TEI Data Manager to agree on a new authorised value

7.5.4. Script

- the default script of an inscription must be specified in your metadata and described in the Hand Description (§11.2.1)
- the instructions in this subsection do not apply to inscriptions that employ a single script, nor to inscriptions in which script varies in one of the following ways:
  - changes in the style of lettering, are covered in §7.5.5 below
  - if script change is concomitant with a language change, then your metadata should specify the scripts used for each language in your inscription, and language changes must be encoded as per §7.2, without explicitly encoding script change
  - changes in the scribal hand do not qualify as script changes and should be encoded as per §4.4
- to tag a chunk of text as being written in a different script, add the attribute `@rend` to its containing element or, if not coterminous with an existing block-level container, use `<hi>` to wrap the chunk of text concerned and add `@rend` to this element
- **the permitted values of** `@rend` for script classification are at present limited to the following:
  - `"grantha"`
  - if you wish to encode a different script that alternates with the primary script of an inscription you are working on, please contact the authors of this Guide and the XML-TEI Data Manager to settle on an authorised value
- when encoding Grantha characters interspersed in Tamil or Vaṭṭeḻuttu script, note that only characters graphically distinct from the default script of the inscription should be marked up in this way

[40] Although the TEI guidelines (https://www.tei-c.org/release/doc/tei-p5-doc/en/html/WD.html#WDWMEG) recommend the use of CSS styling instructions to encode text directionality and orientation, these instructions provide no clear solution for handling case C in the illustration below and handle cases A and B very differently from D. To minimise complexity, we prefer to introduce custom values of `@rend` for this purpose, since our objective is to document the way the original text was written, and not to create machine-actionable code for reproducing the original orientation on screen or in print.

  - e.g. த்ரிஹுவன `<hi rend="grantha">tribhu</hi>vaṉa` if the default script is Tamil (where the *va* is considered Tamil, though it has the same form in Grantha);
  - but `<hi rend="grantha">tribhuva</hi>ṉa` for the same text if the default script is Vaṭṭeḻuttu (where the *va* is definitely not Vaṭṭeḻuttu and is thus classified as Grantha; though it could also be classified as Tamil)

7.5.5. Lettering

- this subsection concerns changes in lettering, i.e. the style in which the glyphs of a particular script are formed
- the following variations are not changes in lettering:
  - change to a different class of script, covered under §7.5.4 above
  - changes in scribal hand, covered under §4.4
- to tag a chunk of text as being written in different lettering, add the attribute `@rend` to its containing element or, if not coterminous with an existing block-level container, use `<hi>` to wrap the chunk of text concerned and add `@rend` to this element
- **the permitted values of** `@rend` for lettering are at present limited to the following:
  - `"ornate"`
  - `"large"`
  - `"small"`
  - `"tall"`
  - `"wide"`
  - `"expanded"` (for character spacing)
  - if more than one of the above is definitely relevant for a particular segment of text, encode them in the order of the list above
  - if you wish to encode a different style of lettering, please contact the authors of this Guide and the XML-TEI Data Manager to settle on an authorised value
- e.g. `<ab rend="ornate">svahasto mama mahārājādhijrāja-śrīharṣasya</ab>`

8. General Guidance for Tidy XML Code

8.1. Spaces and New Lines in the Code

8.1.1. White space

– in coding terminology, **white space** (or whitespace) means a blank space in a document, i.e. any combination of spaces, tabs and new line (carriage return) characters[41]
– white space that affects the processing and transformation of an XML document is called **significant**, while white space that does not is called **insignificant**
– **white space inside XML tags** is as a rule insignificant
  – thus, each of the following are perfectly equivalent:
    – `<lb n="1" break="no"/>`
    – `<lb n="1" break = "no" / >`
    –
```
<lb
n="1"
break="no"/>
```
  – some form of white space must, however, be present before all attribute names, so the above are not equivalent to the following:
    – `<lb n="1"break="no"/>`, which is incorrect XML
  – white space within attribute values is significant, so the above are also not equivalent to either of the following:
    – `<lb n=" 1" break="no"/>`, which is a different value than "1"
    – `<lb n="1" break="no "/>`, which is not meaningful in TEI
– when **an element contains only other elements and white space**, this space is as a rule insignificant; thus, `<p><lb n="1"><gap reason="lost" extent="unknown"/></p>` is perfectly equivalent to each of the following:
  – `<p> <lb n="1"/> <gap reason="lost" extent="unknown"/> </p>`
  –
```
<p>
<lb n="1"/>
<gap reason="lost" extent="unknown"/>
</p>
```
– the handling of **white space within text-containing elements** is a complex matter controlled by the software processing and transforming the XML document; for our purposes
  – in general, white space in content is significant, but in the course of processing it is *collapsed* and *trimmed*
    – **collapsing** means that any type and quantity of white space is reduced to a single space character
    – **trimming** means that space is stripped from the beginning and end of the text content of an element
  – thus, `<p>In a hole in the ground there lived a hobbit.</p>` will normally produce the same transformed text as any of the following:
    – `<p> In a hole in the ground there lived a hobbit. </p>`
    –
```
<p>
In a hole in the ground
there lived a hobbit.
</p>
```
  – moreover, the transformed text generated from `<p><lb n="1"/>In a <unclear>hole</unclear> <supplied reason="omitted">i</supplied>n the ground there lived a hobbit<g type="floret"/></p>` will normally not be affected by any of the following alterations:

[41] See https://wiki.tei-c.org/index.php/XML_Whitespace for a more detailed discussion of white space in XML.

  - `... <unclear>  hole </unclear> ...` (the white space at the beginning and end of the text content of the `<unclear>` element is trimmed)
  - `...<supplied reason="omitted"> i </supplied>n the...` (the white space at the beginning and end of the text content of the `<supplied>` element is trimmed)
- however, each of the following alterations **will** affect the output:
  - `<p><lb n="1"/> In a …` (the added space before "In" is not the first in the content of any element, so it will not be trimmed)
  - `<p><lb n="1"/>In a<unclear> hole</unclear>` (the space moved from a position after "a" to one before "hole" within `<unclear>` will be trimmed, since it is now the first in that element)
  - `...<unclear>hole</unclear><supplied reason="omitted">i</supplied>n the...` (the deleted space between `</unclear>` and `<supplied>` will not be automatically added in processing)
  - `... there lived a hobbit <g type="floret"/></p>` (the added space after "hobbit" is not the last in the content of any element, so it will not be trimmed)

8.1.2. Editorial spaces and markup

- see TG §2.6.1 for general guidance about where and how editorial spaces (for word separation) should be employed, and §8.1.1 above about how spaces in your XML document will be processed
- the above summary of white space in the processing of XML documents will not necessarily apply to the processing of our encoded files, chiefly for the following reasons
  - the attribute `@xml:space="preserve"` is added to the `<text>` element of our documents in order to tell processing algorithms not to trim white space, but the behaviour of various processing algorithms in complex situations is not entirely predictable
  - as we progress with the development of display transformations, white space may be deliberately added or removed in certain markup contexts
- therefore, do not bother memorising the subtleties of whitespace handling theory; instead, **as a rule of thumb**
  - avoid adding spaces to your text except where a space is required for word separation
  - but make sure to add all spaces required for word separation even if an XML element is also present at the same point
  - any kinks appearing due to the presence or absence of space at certain spots can be ironed out later on
  - but to be able to reduce the number of kinks that need to be ironed out, read the specific guidelines below
- with **block-level containers**, feel free to enter their contents in a new line after the start tag and/or to put the end tag in a new line if that makes your work easier for you
- **transition points** (`<lb/>`, `<pb/>` and `<milestone/>` of any kind) are not text containers, so white space after such an element will not be trimmed, therefore pay attention to the following:
  - never add white space after these elements
  - it is acceptable to add white space before these elements provided that they occur between words (and thus do not take `@break="no"`)
    - but it is not necessary to add white space before them; the applicable space or new line will be created in our transformation if `@break="no"` is not present
  - to prevent anomalies in processing, avoid adding white space before these elements if they occur within words (and thus take `@break="no"`)
    - however, to make your XML file easier to scan, you may use a carriage return *within* the `<lb>` tag (at a point where a space is present); this will not interfere with the processing of the code
    - thus both of the following arrangements are permitted and equivalent:
      - `catur-udadhi-salilāsvā<lb break="no" n="2"/>dita-yaśaso`
      - `catur-udadhi-salilāsvā<lb break="no" n="2"/>dita-yaśaso`
    - moreover, for transition points not interrupting words, all of the following arrangements are permitted and equivalent:

    - `saimhaḷakādibhiś ca<lb n="24"/>sarvva-dvīpa-vāsibhir`
    - `saimhaḷakādibhiś ca <lb n="24"/>sarvva-dvīpa-vāsibhir`
    - `saimhaḷakādibhiś ca`
      `<lb n="24"/>sarvva-dvīpa-vāsibhir`
    - `saimhaḷakādibhiś ca<lb`
      `n="24"/>sarvva-dvīpa-vāsibhir`
- **phrase-level elements enclosing text** (e.g. those encoding reading difficulties, editorial intervention and restoration) are text containers, therefore pay attention to the following:
- spaces outside such elements should be used wherever necessary, i.e. wherever the text within a phrase-level container belongs to a word separate from its neighbour outside the container, e.g.
  - `evam <unclear>etaT</unclear>` must have a space before the container
  - `evam e<unclear>taT</unclear>` must not have a space before the container
- white space at the inner edges of such containers may be trimmed from their content
  - therefore any necessary spaces must be placed outside these containers, e.g.
    - `evam<unclear> etaT</unclear>` is incorrectly spaced
    - while `evam <unclear> etaT</unclear>` is correct, though it contains a superfluous space within the `<unclear>` tag
- in elements that encode two (or more) alternative readings for a stretch of text, it may be necessary to add a space at the beginning or end of only one of these alternatives
  - to avoid anomalies in processing, in such cases you should increase the scope of the elements so that the space is not at the edge of the content, e.g.
    - in an ambiguous reading that may be *tathāpi* or *tathā hi*,
      - `tathā<choice><unclear>p</unclear><unclear> h</unclear></choice>i` is incorrectly spaced (since the space before *h* may be trimmed)
      - to eliminate the problem, use `<choice><unclear>tathāpi</unclear><unclear>tathā hi</unclear></choice>` or `tath<choice><unclear>āp</unclear><unclear>ā h</unclear></choice>i`
    - in an editorial correction of *gatakāle* to *gate kāle* (assuming that *gatakāle* is plausible in the context as a compound, but inferior to *gate kāle*)
      - `gat<choice><sic>a</sic><corr>e </corr></choice>kāle` is incorrectly spaced (since the space after *e* may be trimmed)
      - to eliminate the problem, use `<choice><sic>gatakāle</sic><corr>gate kāle</corr></choice>` or `gat<choice><sic>ak</sic><corr>e k</corr></choice>āle`
- **elements representing or enclosing non-alphabetic characters**, i.e. `<g>` and `<num>`
  - use spaces around (and outside) these elements as you would expect to see them in an edition, e.g.
    - between numerals and text
    - between symbols and following text, but not, as a rule, between symbols and preceding text
  - should such an element interrupt a word, do not add editorial spaces around it
  - the content of these elements, when they have any content at all, should not include spaces
    - except when a `<num>` element encloses `<g>` elements or vulgar fraction signs, which for the sake of consistency should be separated by a space from each other and from any Arabic digits within the same `<num>`
- **encoded space**, `<space>`
  - as a rule, add an editorial space on either or both sides of a `<space>` element where it meets text or numerals
    - but do not add a space when `<space>` interrupts a word, as can happen with `<space type="binding-hole">`
  - when one of these elements occurs at a point where an editorial space is already required for word spacing, add a space on either side or both sides of the element as you would expect to see spaces in your displayed edition

  - that is to say, add a space on any side where such elements meet text, but not where they meet other elements of this nature
  - the content of these elements, when they have any content at all, should not include spaces
    - except when a `<num>` element encloses `<g>` elements or vulgar fraction signs, which for the sake of consistency should be separated by a space from each other and from any Arabic digits within the same `<num>`
- **lacunae** (i.e. `<gap>` elements), where they are adjacent to extant or restored text, should normally be separated from text by a space
  - unless you are reasonably confident that the lacuna included part of a word whose other part is preserved outside the lacuna, in which case the text should precede or follow the `<gap>` element without an intervening space
  - by convention we shall take a lacuna encoded without an intervening space as a definite indicator of an editorial hypothesis that the word preserved adjacent to the lacuna is not complete
  - whereas a lacuna encoded with an intervening space will not be regarded to imply any hypothesis about the completeness or incompleteness of the preserved word
- **unintelligible text** (marked up as `<unclear>`, or as `<sic>` if they are clear but unintelligible), where adjacent to extant or restored text, should be spaced in the same way as lacunae
  - in any stretches of text where – due to damage, scribal errors or sub-standard language use – word spacing cannot be allocated definitively but becomes possible after editorial intervention (restoration and emendation), feel free to space the text only as required for your editorial text, i.e. do not feel bound to encode all your spaces as parts of your editorial alterations

8.1.3. Editorial hyphens and markup

- see TG §2.6.2 for general guidance about editorial hyphens (for compound segmentation)
- hyphens should never be used at the ends of epigraphic lines
  - if a line beginning interrupts a word, the element encoding the new line must take the attribute `@break="no"`
  - the same applies to `<pb/>` and `<milestone/>` elements
- when an editorial hyphen (marking an in-compound word boundary) coincides with one of the above elements or with a `<space/>`, the hyphen shall be placed at the beginning of the text after the intervening element(s)
- for other cases where editorial hyphenation interacts with markup, follow the directions for editorial spaces in §8.1.2 above, except that unlike space, hyphens may be used without worries at the beginning or end of text-containing elements

8.2. Top to Bottom Hierarchy

- while applying markup to encode various features, it is useful to keep in mind the rough hierarchy outlined below
  - any element of a particular level may contain elements of a lower level and in many cases also elements of the same level, but may not contain elements of a higher level
  - in case of overlap, elements lower in the hierarchy must give way to higher-level elements, i.e. the lower-level element must be split into two parts across the boundary of the higher-level element (examples below)

8.2.1. Block-level elements representing XML structure and extrinsic structure

- text divisions: the edition `<div>` wraps everything within your edition
- if textpart divisions (§3.4) are present, then these are the direct children of the edition `<div>` and wrap everything else

8.2.2. Block-level elements representing intrinsic structure

– all text in your edition must be contained in one of the block-level elements representing intrinsic structure: `<p>`, `<ab>`, or `<lg>` and `<l>`
– these elements must appear directly within the divisions, never outside them
– these elements must never be nested inside another element of this class, except that `<l>` elements must always be nested in `<lg>` elements
– as a special block-level container, `<fw>` will normally be nested within one of the above regular block-level containers, but may in some unlikely cases be outside them (§3.3.5)

8.2.3. Empty elements representing extrinsic structure

– elements marking transition points, namely line beginnings (§3.2.1), page beginnings (§3.5.2) and pagelike milestones (§3.5.3), must normally be placed within level-2 elements[42]
  – *except* special cases where such transition points cannot be integrated into the text following them, namely
    – page beginnings for the first and last blank pages of a set of copper plates (§3.5.2), which have no associated text
    – reconstructed transition points (of any kind) in massive initial or medial lacunae (§5.4.7)

8.2.4. Empty elements representing local features

– elements belonging to this level are spaces (§4.3) and unrestored lacunae (§5.4) including lacunae with known metre (§5.4.4)
– these empty elements with a spatial extent must by default be contained within level-2 elements and must not be placed before the first level-3 elements of a document
  – *except* that when necessary, lacuna markup may appear outside an element of intrinsic structure and before the first level-3 element (§5.4.7)
– these empty elements with a spatial extent must give way to elements of level 3 and above in case of conflict
  – thus, a space or lacuna interrupted by the start-tag or end-tag of any element of intrinsic structure (e.g. a verse line) or by an element of extrinsic structure (e.g. a line break) must be encoded as two separate instances

Example 8.2.4.A: lacuna split across line break

```
<lb n="1"/><gap reason="lost" extent="unknown"/>devyām utpanna<gap reason="lost"
extent="unknown"/><lb n="2"/><gap reason="lost" extent="unknown"/>ṭṭārikā-devyām utpannaḥ śrī-
mahārājeśvaravarmmā...
```

➢ here, the lacuna between *utpanna* and *ṭṭārikā* is encoded as two separate lacunae, one at the end of line 1, and another at the beginning of line 2; the line beginning element stands between these two

8.2.5. Phrase-level elements

– elements belonging to this level include:
  – feature markup, e.g. for reading difficulties (§5.3) and script (§7.5.4)
  – semantic markup, e.g. for numerals (§7.1), abbreviations (§7.3) and language (§7.2)
  – editorial intervention, e.g. restoration (§5.5), correction (§6.2) and normalisation (§6.3)
– such phrase-level markup encoding semantic or descriptive information about specific characters must
  – always be contained within level-2 elements
  – in case of conflict give way to elements of level 4 and above
– thus, a phrase-level element interrupted by the start-tag or end-tag of any element of intrinsic structure or by an element of extrinsic structure must be encoded as two separate instances

[42] This is not a technical requirement and its violation does not result in your code becoming invalid. However, since in many cases these elements will *have to* appear within block-level containers, we consider it better practice to always place them so, for consistency's sake.

Example 8.2.5.A: editorial restoration split across verse lines

```
<lg n="1" met="anuṣṭubh">
  <l n="a">svadattām para-dattām vā</l>
  <l n="b">yo hareta va<supplied reason="lost">sundharām</supplied></l>
  <l n="c"><supplied reason="lost">sa viṣṭhāyāṁ kṛmir bhūtvā</supplied></l>
  <l n="d"><supplied reason="lost">pitṛ</supplied>bhiḥ saha pacyate</l>
</lg>
```

- the lost and supplied text in the middle of the stanza is encoded as three separate instances of supplied text: the end of *pādas b*, all of *pāda c*, and the beginning of *pāda* d

– phrase-level elements may usually be nested inside others except for `<unclear>`, for which see below
  – otherwise, you may nest phrase-level markup as logic dictates, so long as you avoid overlaps, which must be eliminated by creating separate instances of interrupted markup elements

Example 8.2.5.B: overlapping phrase-level elements

uktañ ca bhagavatā ve~~da-vyāsena vyāsena vyā~~sena

- the stretch struck out in the text above represents unclear text in the original
- the last iteration of *vyāsena* is scribal dittography, which you as editor suppress

```
uktañ ca bhagavatā ve<unclear>da-vyāsena vyāsena</unclear>
<surplus><unclear>vyā</unclear>sena</surplus>
```

- the `<unclear>` element is split into two parts, one in the retained text and one in the suppressed segment

– the element `<unclear>` must, by EpiDoc rules, contain only text and may never contain any markup elements except `<g>`; thus,
  – when `<unclear>` overlaps with another markup element, it is always `<unclear>` that must be split into separate parts inside and outside the other element, as in Example 8.2.5.B above
  – when a different element (other than `<g>`) needs to be nested inside `<unclear>`, then again, `<unclear>` must be split into separate parts as in Example 8.2.5.C below, to allow the nesting of `<unclear>` within the other element instead of the other way round

Example 8.2.5.C: overlapping phrase-level elements

puṇyābhi~~vriddhaye~~

- the stretch struck out in the text above represents unclear text in the original
- the spelling *ri* is non-standard and you, as editor, flag it as such

```
puṇyābhi<unclear>v</unclear><orig><unclear>ri</unclear></orig><unclear>ddhaye</unclear>
```

- the `<unclear>` element is split into three parts, one before the flag, one for the flagged segment within the `<orig>` tag, and one after it

9. Additional Content Divisions

9.1. The Critical Apparatus

9.1.1. Overview

– the primary purpose of an apparatus in our project's diplomatic editions is to record significant alternative readings, restorations and emendations by previous editors
  – unless a specific sub-corpus or a particular text requires this, a complete record of minor differences (e.g. typing and printing errors, orthographic normalisation or whether a reading is shown as unclear or tentatively read/reconstructed) is not desirable
  – the apparatus is not normally the place to record your own editorial choices such as preferred readings, restorations, emendations, and flagging of any original text as unexpected or inappropriate
    – all of these must be encoded in the inline markup within your `<div type="edition">` wherever possible
    – however, apparatus notes may be used to add details or reasoning to such choices, and to record highly tentative proposals you are not confident enough to include in the edition itself
– TEI and EpiDoc allow a variety of ways to encode a critical apparatus for a scholarly edition; within the DHARMA project we shall limit ourselves for epigraphical work to an **external apparatus criticus**
  – this means that the apparatus entries are encoded in a section separate from the edition (namely, within the `<div type="apparatus">`) and referenced to locations within the text
– for the sake of project-wide consistency and ease of processing, the external apparatus shall be encoded as follows
  – after your `<div type="edition">` element, your document shall include a `<div type="apparatus">` containing the element `<listApp>`
  – within `<listApp>`, each apparatus entry must be individually wrapped in the element `<app>`, which
    – must **mandatorily** have the attribute `@loc` to indicate the location (epigraphic line) to which your entry pertains, as per §9.1.2 below
    – must **mandatorily** contain `<lem>` as its first child element, containing a lemma as per §9.1.3 below
    – may contain one or more `<rdg>` elements (one for each alternative reading/restoration/emendation), as per §9.1.4 below
    – may contain one or more `<note>` elements containing a human-readable note in freeform text pertaining to that particular lemma, as per §9.1.7 below
– if your edition division includes **boxlike partitions** (§3.4), then mandatorily **replicate** those **textpart divisions** within the `<div type="apparatus">`
  – see §9.1.8 below for details

Example 9.1.1.A: critical apparatus

➢ a snippet from the edition `<div>` of the Allahabad *praśasti* of Samudragupta

```
<div type="edition" xml:lang="san-Latn">
  [...]
  <lg n="2" met="śārdūlavikrīḍita">
    <l n="a"><lb n="7"/><supplied reason="lost"
cert="low">ā</supplied><unclear>ry</unclear>y<supplied reason="lost"
cert="low">ai</supplied><unclear>h<unclear>īty upaguhya bhāva-piśunair utkarṇṇitai romabhiḥ</l>
    [...]
  </lg>
  [...]
</div>
```

➢ and from the corresponding apparatus

```
<div type="apparatus">
  [...]
  <app loc="7">
    <lem source="bib:Bhandarkar1981_01"><supplied reason="lost"
cert="low">ā</supplied><unclear>ry</unclear>y<supplied reason="lost"
cert="low">ai</supplied><unclear>h<unclear>īty
    </lem>
    <rdg source="bib:Fleet1888_01"><supplied
reason="lost">ā</supplied><unclear>ry</unclear>y<supplied reason="lost">o</supplied>
<unclear>h<unclear>īty
    </rdg>
    <rdg source="bib:Goyal1967_01"><supplied
reason="lost">a</supplied><unclear>rh</unclear>y<supplied reason="lost">o</supplied>
<unclear>h<unclear>īty
    </rdg>
    <rdg source="bib:Agrawala1983_01"><supplied reason="lost">e</supplied><unclear>h</unclear>y
<supplied reason="lost">e</supplied><unclear>h<unclear>īty
    </rdg>
  </app>
  [...]
</div>
```

9.1.2. Indicating location

– the value of the attribute `@loc` must unambiguously specify the location to which the apparatus entry pertains
  – the primary purpose of this value is to be intelligible to a human reader, but we may wish to make it machine-actionable in the future and therefore adopt rigorous rules from the beginning
– locations shall normally be identified as epigraphic lines, using line numbers as reference, because they are (almost) ubiquitous and unique in our editions
  – therefore, the value of `@loc` shall normally be the line number, i.e. the `@n` attribute of the `<lb/>` element representing the beginning of the target line
    – if the entry refers to a segment of text that extends across a line break, then `@loc` shall be the number of the first line where the segment begins; see also §9.1.6 below about the inclusion of line (and other) beginning tags in a lemma
– the sole exception to the ubiquitous presence of line numbers in our editions is forme work (§3.3.5)
  – it is therefore not possible to refer to the contents of forme work using line numbers
  – should you need to add an apparatus entry for a forme work item, the value of `@loc` shall be the number of the forme work item prefixed with the letters “fw”, e.g. `<app loc="fw2r">` to refer to the forme work element with the number 2r
– the sole exceptions to the uniqueness of line numbers are **editions comprised of textpart divisions** (§3.4), where line numbers will be restarted in each division
  – however, since the apparatus division of the document must replicate the textpart divisions of the edition division, `@loc` references in the apparatus still only need to include the `@n` of the target line and will remain unambiguous

– although some of us may be used to referring to stanza numbers in apparatus entries, we have chosen not to allow this in order to reduce the complexity of referencing
  – apparatus entries for lemmas in verse shall be referred to by line number like those in prose
  – notes concerning entire stanzas shall be added to the commentary, not to the apparatus

9.1.3. Specifying a precise spot by a lemma

– the exact spot (locus) to which an apparatus entry pertains is specified by a lemma, tagged with the XML element `<lem>`
– there are no strict rules for the **extent of your lemmas**; as with any critical apparatus, lemmas should be large enough to make them unambiguous within the line referred to in the `@loc` attribute and small enough to remain concise
  – lemmas should preferably consist of whole words, which may be members of compounds in the text
  – when the lemma boundary does not coincide with a word boundary (i.e. an editorial space) in the text, indicate this in the lemma as follows:
    – when a lemma cuts a non-compound word, use the character ° (but preferably use a whole word as a lemma), e.g.
      – text *puṇyābhivṛddhaye*; lemma *°vṛddhaye* (where *abhivṛddhaye* is not a compound word)
    – when a boundary between independent words or compound members is obscured by sandhi, use the character °, e.g.
      – text *puṇyābhivṛddhaye*; lemmas *puṇyā°* and *°ābhivṛddhaye* (**not** *puṇya°* and *°abhivṛddhaye*)
      – text *yathāsmābhiḥ*; lemmas *yathā°* or *°āsmābhiḥ* (**not** *°asmābhiḥ*)
      – text *maharṣi*; lemmas *maha°* or *°ṛṣi*
    – when a boundary between compound members is not obscured by sandhi, then depending on whether or not you hyphenate that word in your edition, use
      – the character ° if you do not hyphenate, e.g.
        – text *śrīpolekeśivallabhasya*; lemmas *śrī°* or *°polekeśi°* or *°vallabhasya*
      – a hyphen if you do hyphenate, e.g.
        – text *śrī-polekeśi-vallabhasya*; lemmas *śrī-* or *-polekeśi-* or *-vallabhasya*
  – avoid very long lemmas, if possible, by breaking them up into several smaller ones
  – long lemmas that cannot be split up in this way may be shortened by replacing a section of them with `<gap reason="ellipsis"/>` (which will be displayed as "…")
– the lemma should appear **exactly as it appears in your digital edition**,[43] including any markup that encodes information about reading difficulties and editorial intervention
  – see §9.1.6 for some concerns pertaining to the use of markup in lemmas
– `<lem>` may take the attribute `@source` (§10.6.2) to show that **a previous edition supports the reading adopted in your edition**
  – see §9.1.5 below for guidance on deciding whether a reading supports yours
  – if your apparatus entry consists only of a lemma (without either `<rdg>` nor `<note>`), then `@source` must be present on the lemma, since the sole purpose of such an entry is to credit a previous editor for a difficult reading or ingenuous restoration
    – such apparatus entries will be rare; only create them if you feel they are necessary
  – if your apparatus entry contains a `<note>` but no alternative readings, then there is no need to credit the lemma to a source unless credit is particularly due to a previous editor for an ingenious reading adopted in your edition
  – if your apparatus entry contains alternative readings, then (whether or not it also contains a `<note>`), any previous editors who agree with the lemma must be credited with `@source`
  – keep in mind that whenever you use `@source` on a lemma, the bibliographic citation of that publication in bibliography division must include an encoded siglum for use in the apparatus, as per §9.4.3

[43] That is to say, in addition to preserving the markup in your edition, you must also not add any further markup such as italicisation.

9.1.4. Alternative readings, restorations and emendations

– alternatives to your edited text should be recorded as the contents of an `<rdg>` element
  – text within this element should be marked up with XML tags to clearly indicate what the cited editor deemed unclear, emended or supplied
  – that is to say, convert the original editor's markup and/or additional explanation into XML tags endorsed by this guide as best possible
    – since the markup found in many printed editions is less expressive and/or less rigorously consistent than our EpiDoc conventions, you may need to interpret the intention of the original editor and mark up alternatives accordingly
    – we deem this method to be preferable to the disadvantages inherent in the alternative, namely recreating all brackets etc. precisely as observed in the previous edition
  – never retain any traditional editorial markup (such as brackets or asterisks)
  – see §9.1.6 for some concerns pertaining to the use of markup in readings, in particular about the encoding of line breaks within a reading
– the **extent of an alternative text segment** should always correspond exactly to the extent of its lemma
  – as in lemmas, use ° or a hyphen at the beginning or end of an alternative if its boundary does not coincide with the boundary of an independent word of the text
    – see §9.1.3 for details and examples
  – if a printed edition shows nothing (i.e. not even a lacuna) at a locus where your edition has content (including a tentative reading or a lacuna), the reading of the edition in question may be represented by an empty element, e.g.
    – `<rdg source="bib:VenkatasubbaAyyar1943_01"/>`
– alternatives **must always be credited** to the editor(s) who proposed or endorsed them, using the attribute `@source` in `<rdg>`; see §10.6.2 for details
  – see §9.1.5 below for guidance on deciding whether two editors' readings may be deemed identical
  – if your apparatus includes at least one lemma with alternative readings *and* you cite more than one previous editor, your apparatus entries should always be formulated in a **"positive"** manner: for any lemma with one or more alternative readings, clearly indicate (either under the lemma or under one of the alternative readings) what the readings of **all** previous editors were
    – however, keep in mind that your apparatus does not have to list every minor divergence from previous editions (see §9.1.1)
  – also keep in mind that whenever you cite a reading from a publication, the bibliographic citation of that publication in your bibliography division must include an encoded siglum for use in the apparatus, as per §9.4.3

9.1.5. Identical lemmas, identical readings

– when **deciding whether two readings may be deemed identical**, i.e. whether a certain previous edition's reading agrees with your lemma or with the reading cited from another previous editor, you should normally consider only the actual received text shown in each edition
  – ignore differences that consist only in the presence or absence of markup for unclear or restored characters
    – e.g. if one reading is `ya<unclear>thā</unclear><supplied reason="lost">smābhiḥ</supplied>` and another is *yathāsmābhiḥ* without any markup, then the two readings are to be deemed identical
  – ignore previous editors' emendations or normalisations if they do not affect the interpretation of the text, i.e. if you (or another editor) have chosen only to flag a phenomenon as erroneous or non-standard, or chosen to ignore one, whereas a previous editor reads the same but emends or normalises it, then their reading is still to be deemed as identical to your lemma
  – if a previous edition contains a minor orthographic mistake that does not affect the meaning and that may well be a typographic error in that edition, feel free to ignore it if their reading is otherwise identical to yours (or to another previous edition's)

  - if the original editor uses *ś* to transliterate both Grantha *ś* and Tamil *c*, and you are unable to determine which is meant, choose in `<lem>` the interpretation you favour and add a `<note>` to the `<app>`, such as "The original editor's reading could also be interpreted as …" (specifying the original editor by name if the `<app>` contains several `<rdg>`)
  - in all of the above cases, the recommendation of ignoring such differences may be overridden for highly problematic spots of text, where you may find it best to faithfully reproduce each previous editor's reading down to the last detail
- whenever **multiple editions** are **cited** for a lemma or reading, remember that their citations must be listed **within a single** `@source` **attribute** in chronological order (§10.6.2)
- when more than one previous editor supports a reading, but the **readings of these editors differ in minor details** you ignore as per the above guidelines, by preference show the reading as featured in the first of the cited editions

9.1.6. XML tags in lemmas and readings

- pay attention to the following, especially when you copy and paste the marked-up text of a lemma, but also when adding markup to a reading:
- **tags for block-level containers** (`<p>`, `<ab>`, `<lg>` and `<l>`) must not be included in lemmas or readings
  - for a problematic locus extending across a boundary between such containers, preferably create separate lemmas
  - if separate lemmas do not seem appropriate, you may simply delete the (start or end) tag belonging to such an element from your lemma
- **empty elements** representing transition points (`<lb/>`, `<pb/>` and `<milestone/>` of any kind) shall, however, be included in both lemmas and readings
  - to reduce code clutter, feel free to use these elements without any attributes, since the purpose of including them in lemmas and readings is only to show the fact that such a transition is present (or was indicated as present, not necessarily always in the right place, by a previous edition)
  - we foresee that all of these elements will be displayed as a simple / character when they appear as a lemma (thus, since a pagelike partition will always be followed by a line beginning, these together will display as //)
  - remember that when a lemma extends across such a boundary, the `@loc` of the apparatus entry must be the number of the line where the lemma begins
- when a lemma or reading includes **phrase-level markup** (e.g. `<unclear>`), pay attention to start-tags and end-tags, which may be outside the lemma in the edition, so within your lemma,
  - add the start-tag for retained markup commencing before and ending inside your lemma
  - add the end-tag for retained markup commencing inside your lemma and ending after it
  - add start and end-tags for a lemma snipped from within a longer stretch of phrase-level markup

9.1.7. Freeform apparatus notes

- if you find the encoding within `<lem>` and/or `<rdg>` insufficient for recording certain details about your base reading or a cited alternative, add a `<note>` element within the relevant `<app>` entry
  - see §10.4.1 for general guidance on notes
- any editorial notes concerning a segment of text that cannot be conveniently identified by a line number and lemma should be placed in the commentary, not the apparatus

9.1.8. Textpart divisions in the apparatus

- as stated in the Overview above, if your edition includes boxlike partitions (§3.4), then these divisions must be replicated within the apparatus
- in this case, inside `<div type="apparatus">`, create as many `<div type="textpart">` elements as there are in the `<div type="edition">`
  - in the start-tag of these replicated divisions, include all attributes (`@subtype` and `@n`) with the same value that they have in the edition division

  – if your edition's textpart divisions have `<head>` elements, these should likewise be replicated in the apparatus, immediately after the start-tag of each textpart division
– the apparatus container `<listApp>` must in this case be enclosed within `<div type="textpart">`
  – if you have apparatus entries for more than one textpart, then create a `<listApp>` within each textpart to wrap the `<app>` elements belonging to that textpart
  – if one or more textparts have no pertaining apparatus entries, then the textpart division (with attributes) must still be created for these in the apparatus, but that division shall not contain anything (except, if applicable, `<head>`), i.e. there should not be a `<listApp>` wrapper, nor any `<app>` items there

Example 9.1.8.A: critical apparatus with more than one textpart, each with content

```
<div type="apparatus">
 <div type="textpart" n="A">
  <head xml:lang="eng">Seal</head>
  <listApp>
   <app loc="1">
    <lem>ku<unclear>mā</unclear>rāmātyādhikaraṇasya</lem>
    <note>Our restitution ... </note>
   </app>
  </listApp>
 </div>
 <div type="textpart" n="B">
  <head xml:lang="eng">Plate</head>
  <listApp>
   <app loc="2">
    <lem>°bhi<supplied reason="omitted">ḥ</supplied> mekhalayā</lem>
    <note>Absence of doubling ... </note>
   </app>
   ...
  </listApp>
 </div>
</div>
```

Example 9.1.8.B: critical apparatus with more than one textpart, one without content

```
<div type="apparatus">
 <div type="textpart" n="A">
  <head xml:lang="eng">Seal</head><!--Empty textpart.-->
 </div>
 <div type="textpart" n="B"><head xml:lang="eng">Plates</head>
  <listApp>
   <app loc="6">
    <lem>-pu<unclear reason="eccentric_ductus">ñja</unclear></lem>
    <note>While <foreign>puñja</foreign> must have been intended ... </note>
   </app>
   ...
  </listApp>
 </div>
</div>
```

9.2. The Translation

9.2.1. Overview

– whenever possible, a translation should be included in your XML document along with your edition
– the translation must be wrapped in `<div type="translation">`
  – this division follows the edition division and the apparatus division
– use the following attributes for `<div type="translation">`
  – mandatorily, `@xml:lang` to encode the target language (see Appendix D for a list of language codes permitted as values for this attribute)
  – generally, one of the following:

  - `@source` (§10.6.2), if a published translation is adopted verbatim
  - `@resp` (§10.6.1), if the translation is by you and/or another project member
  - in more complicated situations, such as a collaborative translation or the use of an unpublished translation by another person, finer details of authorship may be recorded in a credit note (§9.2.2), which may be used in addition to or instead of `@source` or `@resp`
- your translation should be a convenient representation of the intent of the original, hence it should be as literal as seems useful, but as free as seems necessary
- translations of your own should correspond to the text as you have edited it, including restorations and emendations
- by contrast with our epigraphic editions, where the spelling of the original is always retained in xml and viewable in display (notwithstanding any editorial interventions that may be marked up), in translation your are advised to ignore the original's superficial irregularities/oddities of punctuation and spelling
  - in particular, normalise the spelling of original personal names, toponyms and terms retained from the original, as suggested for "loose transliteration" in TG §2.2.2

9.2.2. Front matter in a translation

- we foresee that **headings for translations** will be generated automatically on the basis of the `@xml:lang` and `@source` or `@resp` attributes of the translation division
  - e.g. `<div type="translation" xml:lang="eng" resp="part:daba">` displayed as a heading "Translation into English by Dániel Balogh"
- to create a **custom header** for a translation where the above is insufficient,
  - include the element `<head>` as the first item within `<div type="translation">`
    - containing free text that is to be displayed as a heading above the translation
  - such headers, if used, will replace the auto-generated header, so it is recommended that you include the word "Translation" and a specification of the target language
- **a credit note** may be added **after the header** where necessary, e.g. to give credit to someone
  - who has helped with the translation as a whole, or
  - whose published translation you have reworked, or
  - whose unpublished translation you are adopting in whole or in part
- to create a credit note, include the element `<note type="credit">` as the first item within `<div type="translation">` (or as the second item, immediately after the custom `<head>` if one is used)
  - containing free text consisting of one or more complete sentences (with a capital initial and final punctuation) and, if applicable, including a citation (§10.4.5) of the published translation credited
  - this `@type` of `<note>` will only be used in the translation division, for this particular purpose
- keep in mind that a credit note should not be used in translations adopted verbatim from a publication, nor for translations created by project members
  - in both these cases, authorship must be encoded in the attributes of the translation division (using `@source` or `@resp` as applicable, see §9.2.1 above)

9.2.3. Attaching multiple translations

- if you wish to include more than one translation, simply repeat `<div type="translation">` for each of them, with attributes applied as above
- there is no requirement to include multiple translations just because they are available
  - new translations created by you, the encoder of the inscription, are preferred
  - pre-existing translations may be added
    - when a new translation is not feasible or a previous translation is so satisfactory that a new translation is unnecessary
    - when you deem them good enough and are in a major modern language other than the one you are translating to
    - when you deem them relevant to the history of the understanding of the inscription

– when you include more than one translation in your XML document, these should be presented in an order of decreasing usefulness, prioritising those which are more recent, more accurate, and in more widely spoken languages

Example 9.2.3.A: multiple translations

```
<div type="edition" xml:lang="x-oldcham-Latn">
  ...
  <ab>pu vyā</ab>
  ...
</div>
...
<div type="translation" xml:lang="eng">
  ...
  <p>Her majesty the queen</p>
  ...
</div>
<div type="translation" xml:lang="fra">
  ...
  <p>Sa majesté la reine</p>
  ...
</div>
```

9.2.4. Reproducing a published translation

– when encoding a previously published translation without changes, the **author** of that translation **must be credited** using the attribute `@source` (§10.6.2)
  – to credit a contributor other than the author of a published translation, or to indicate the author of a published translation that you have improved on, attach a credit note (§9.2.2) to the beginning of the translation
– as far as feasible, **convert** any character-based **markup** used by the original translator to XML-based markup
  – translator's marks that cannot be converted to XML equivalents may be retained (as an exception to the rule of not using non-XML markup)
    – clarify any such markup within the credit note
– any **words transliterated** from an Indic script appearing in a translation you adopt, or the notes attached to it, should be silently converted to our transliteration system whenever feasible
  – however, when the element `<quote>` (§10.4.4) is used to cite parts of a translation, the original transliteration should be preserved
– **notes attached to a published translation** do not have to be reproduced verbatim or in their entirety
  – however, any notes attached to a translation will by default be assumed to be by the author credited in the attributes of the translation as a whole (as per §9.2.1), therefore any notes that are not reproduced verbatim from the published translation must be attributed explicitly as follows:
    – when adding notes of your own, use `@resp` (§10.6.1) on each such note to encode your authorship
    – when supplementing a published translation with notes from another source, use `@resp` (§10.6.1) or `@source` (§10.6.2) to assign credit to a project member or to a publication, as applicable
    – when paraphrasing notes that are not your own, likewise use `@resp` (§10.6.1) to encode your authorship, and include in your paraphrase an attribution to the original author of the note
      – if that original author is the person to whom the translation as a whole is credited, then this attribution need not include an encoded reference, e.g. "Fleet observes that..."
– if a published translation is **based on a reading**, restoration or emendation **other than what you adopt in your edition**, it is highly recommended that you point this out in a note of your own, attached to the spot where the translation is based on divergent text
– when a published translation you are encoding **omits a stretch of the text** (for example because it is unintelligible or because the original publisher considered a part of the text not worthy of attention), this shall be indicated in your encoding as per §9.2.12)
– handling **mistakes in a published translation**

- it is recommended that you silently correct any obvious typographic errors in a published translation you are reproducing
- unusual or incorrect interpretation and unexpected transliteration/normalisation found in published translations may be flagged with `<sic>` as per §9.2.11

9.2.5. Structural markup in translation

- the overall structure of a translation should, as far as practicable, imitate the structure of the original inscription, in the following manner
- if your edition includes boxlike partitions (§3.4), the textpart divisions of your edition must be mandatorily replicated in your translation (with the same attributes and, if applicable, `<head>` elements)
- pagelike partitions (§3.5), if present, may be replicated or omitted from the translation as you see fit
  - for replication, use the same element with the same attributes as that used in your edition, and include the `<head>` elements if applicable
- gridlike partitions (§3.6) and quasi-partitions (§3.3) including forme work (§3.3.5) shall not be replicated in the translation
- line beginnings shall not be replicated, but line numbers may be indicated as per §9.2.6 below
- block-level containers in the original (i.e. `<p>`, `<ab>` and `<lg>`) shall be normally replicated as a corresponding paragraph (`<p>` element) in the translation, but feel free to use a smaller or larger number of `<p>` elements at your discretion
  - for paragraphs translating verse, add the attribute `@rend` with the value `"stanza"` to the `<p>` element
    - should your translation of a stanza consist of verselike lines that will need to be displayed as separate typographic lines, you may wrap each of these in an `<l>` element within the `<p>` element corresponding to a stanza

9.2.6. Indicating correspondence to the original

- to indicate how block-level translation elements correspond to parts of the original text, you should normally add the attribute `@n` to each `<p>` element in the translation as reference to either a line number or a stanza number in the original
  - for short inscriptions best translated as a single paragraph of prose, such referencing may be omitted
  - to indicate a line or stanza in the original, simply use the value of `@n` from the appropriate `<lb/>` or `<lg>` element of the original
    - in `<p>` elements translating verse (and thus carrying the attribute `@rend="stanza"`), this `@n` will be interpreted as a stanza number, while in `<p>` elements translating prose (and thus without the `@rend` attribute) it will be interpreted as a line number
  - to refer to a range of lines or stanzas, use a hyphen between two values, e.g.
    - `<p n="1-3">` a paragraph translating prose in lines 1 to 3
    - `<p rend="stanza" n="8-9">` a single paragraph translating stanzas 8 and 9 together
  - to refer to non-contiguous lines or stanzas, use a comma and a space between two values, e.g.
    - `<p n="1, 3">` a paragraph translating prose from lines 1 and 3 (but not 2)
    - `<p rend="stanza" n="8, 10">` a paragraph translating (parts of) stanzas 8 and 10 together
- the above indicators are for human readers and are not meant to be machine-actionable, therefore
  - feel free to refer to larger ranges of lines or to several stanzas for passages best translated in larger chunks
  - feel free to refer to the same line or stanza number in several translation elements, each of which includes parts of a single line or stanza of the original
  - should the intelligibility of the translation demand it, feel free to translate items in a different order than that in which they appear in the original

Example 9.2.6.A: numbering in translation to indicate correspondence to the original

```
<div type="translation" xml:lang="eng">
  <p n="1-9">Hail! From the victorious [...] </p>
  <p n="9-15">There is this village [...] </p>
  <p n="15-17">In the first year [...] </p>
  <p rend="stanza" n="1">By numerous kings, many times land has been given. Whoever holds land at
a given moment, to him does the fruit then belong.</p>
</div>
```

9.2.7. Phrase-level markup in translations

– in addition to plain English (or other modern-language) text, the translation division may contain phrase-level markup of the following kinds
  – globally permitted miscellaneous markup as per §10
  – some additional encoding solutions available only in translations, as outlined in the subsections following this one (§9.2.12 to §9.2.11)
– no other markup should appear in translations,[44] and this applies also to the use of non-XML markup such as brackets, asterisks and other signs

Example 9.2.7.A: phrase-level markup in translation

```
<div type="translation" xml:lang="eng">
  <p>Indeed <supplied reason="explanation"><foreign>asti</foreign></supplied>, with respect to
vendible properties in this division, the sale of <supplied reason="explanation">a
<foreign>kulyavāpa</foreign> of</supplied> waste land that is without revenue charges and yields
no tax, to be enjoyed in perpetuity in accordance with the law on permanent endowments, is
customary for one hundred <foreign>kārṣāpaṇa</foreign>s. And no conflict of interest <supplied
reason="explanation"><foreign>virodha</foreign></supplied> whatsoever <supplied
reason="subaudible">will result</supplied> through its sale: <supplied reason="subaudible">on the
contrary,</supplied> for His Majesty <supplied reason="subaudible">there will be</supplied>
increase of wealth and attainment of one sixth of the merit.</p>
</div>
```

9.2.8. Foreign words

– words in a language other than the language of the translation must be tagged as `<foreign>` as per §10.3.3
  – there are no special rules or methods applicable to translations, and this subsection only exists to make it explicit that this encoding can and must be used in translations
– as per §10.3.3, words in the inscription's language or Sanskrit do not require the attribute `@xml:lang`, including
  – such words appearing in the text without any other markup, e.g.
    – technical terms (e.g. one `<foreign>kulyavāpa</foreign> of land`)
    – unintelligible text that is not translated as per §9.2.12
  – such words inserted into the translation as explanation, as per §9.2.9
– as per §10.3.3, when words in another modern language are encoded as `<foreign>`, the attribute `@xml:lang` must always be present

9.2.9. Additions to the translation

– words in the translation that do not correspond to anything in the extant original text shall be tagged as `<supplied>`, as outlined below
  – in each of these cases, `@cert="low"` may be added to this element to indicate tentativeness; see §9.2.10 below
– words **added to the translation for the sake of target language syntax** shall be marked up as `<supplied reason="subaudible">`

[44] If you feel your translation needs any further markup, please consult the authors of the Guide and the project's XML-TEI Data Manager.

  - this markup method is to be used for words that, though not explicitly present in the original, need to be read to get a proper translated sentence; see the next point about additions that are not required for completing the syntax in the target language
  - e.g. `<p>He was generous to his subjects and <supplied reason="subaudible">therefore</supplied> loved <supplied reason="subaudible">by them</supplied> … </p>`
  - we foresee that this markup will be displayed as square brackets, e.g. "He was generous to his subjects and [therefore] loved [by them]."
  - do not clutter a translation with such tags unless you find that such accuracy is essential: depending on how free or literal your translation is, you may prefer to avoid the use of this element
- segments of translation corresponding to text **restored by the editor in the original** shall be indicated in the translation using the same elements as in the edition, namely
  - lost text, e.g. `The truest of <supplied reason="lost">kings</supplied>...` corresponding to `<supplied reason="lost">nṛpa</supplied>-sattamaḥ` in the edition
  - text omitted by the scribe, e.g. `The truest of <supplied reason="omitted">kings</supplied>...` corresponding to `<supplied reason="omitted">nṛpa</supplied>-sattamaḥ` in the edition
  - use the same encoding for concepts presumed to have been present in a larger lacuna of the text, even if they have not been restored in the edition (because there is no way to know what synonym was used or where a word was located within a lacuna)
    - e.g. `The village named <supplied reason="lost">was granted</supplied>...` for `...nāmo grāmaḥ <gap reason="lost" quantity="12" unit="character"/>`
    - note that in this case you will have to use `@reason="lost"` in the `<supplied>` element even if the corresponding `<gap>` element in the edition has `@reason="illegible"` (§5.1)
    - such restorations will usually not cover the whole of a lacuna and will thus need to be used in conjunction with lacuna markup as per §9.2.12
  - we foresee that this markup will be displayed as square brackets, without distinction from words added for the sake of target language syntax, e.g. "The truest of [kings]..."
  - do not clutter a translation with such tags unless you find that such accuracy is essential: as a rule, lost or omitted text shorter than a full word and confidently restored by you in the edition should not be marked up as supplied in the translation
- words implied by the context and **added to the translation for the sake of clarification or disambiguation** shall be marked up as `<supplied reason="explanation">`
  - this markup method is to be used for supplementary words that are not required for completing the syntax in the target language or may even interrupt the sentence; see the previous point about words that need to be read to get a proper translated sentence
    - e.g. `<p>... devotion to <supplied reason="explanation">Viṣṇu</supplied> the bearer of the discus and the mace ... </p>`
  - we foresee that this markup will be displayed as parentheses, e.g. "... devotion to (Viṣṇu) the bearer of the discus and the mace ..."
  - to **add words of the original** (or equivalents in Sanskrit or another applicable major language) next to translated words, in order to make your translation more transparent, enclose these in `<supplied reason="explanation">` as above, and within that tag, use `<foreign>`
    - e.g. `<p>Homage to that thousand-headed Person <supplied reason="explanation"><foreign>puruṣa</foreign></supplied> ... </p>`
    - foreseeably displayed as "Homage to that thousand-headed Person (*puruṣa*)"
    - as per §10.3.3, the attribute `@xml:lang` need not be used in this case

9.2.10. Indicating uncertainty

- to indicate the uncertainty or tentativeness **of a translated word or phrase**, wrap a segment of translation in `<seg cert="low">`

  - e.g. `<p>Out of the hundred and <seg cert="low">twenty</seg> shares comprising this village ... </p>`
  - use this method regardless of whether the tentativeness of your translation stems from the condition of the original (e.g. partly or wholly unclear, illegible or restored) or from the obscurity of its language
  - this will probably be displayed wrapped in a pair of question marks (inverted and regular), e.g. In the great and renowned city ¿named? two ¿times?> five ...
- to indicate the uncertainty or tentativeness **of a word added to the translation**, add `@cert="low"` to any `<supplied>` element used as per §9.2.9 above
  - e.g. `<p>On this day he <supplied reason="explanation" cert="low">Viṣṇuvardhana</supplied> donated … </p>`
  - this will be displayed with a ? added inside the brackets corresponding to `@reason` as above, e.g. On this day he (Viṣṇuvardhana?) donated ...

9.2.11. Indicating incorrect or unexpected text

- the element `<sic>` may be used in translations in either of the following circumstances
- in your own translation, you may deploy `<sic>` to highlight words or stretches of translation corresponding to an original that seems inappropriate in context
- when reproducing a published translation, `<sic>` may be used to highlight points where the original translator's usage or transliteration practice is wrong or unexpected

9.2.12. Gaps in the translation

- **lacunae** in the original shall be indicated in the translation using the same elements as in the edition, namely, `<gap reason="lost"/>` or `<gap reason="illegible"/>`, distinguished as per §5.4.2
  - normally, these elements may be used in a translation without any further attributes
    - gaps encoded in this way will probably be displayed as *[…]* (regardless of the value of `@reason`) and will be sufficient for most lacunae in most translations
  - however, when you deem it essential to present accurate details of a lacuna in the translated text, you may optionally use the attributes `@unit`, `@quantity` and `@precision` as set out in §5.4.3 and §5.4.6, with the following additional options:
    - fractional numbers (decimal fractions) for `@quantity` are permitted in this case, even though they cannot be used in the edition
    - `@unit` may be `"line"` even if a multiline lacuna is represented in the edition as a series of inline lacunae
  - such gaps will probably be displayed as text, e.g.
    - `<gap reason="lost" quantity="3.5" unit="line"/>` displayed as *[3.5 lines lost]*
    - `<gap reason="illegible" quantity="10" unit="character" precision="low"/>` displayed as *[ca. 10 characters illegible]*
  - no other methods of lacuna markup shall be used in translations, i.e. avoid the use of `@extent` and the encoding of sub-*akṣara* lacunae
- when **a segment of extant text is not translated because it is not intelligible**, this shall be indicated in the following way
  - mandatorily create `<gap reason="ellipsis"/>` without any further attributes to indicate the place in the translation where text is skipped
    - such a gap element in a translation will be displayed as …
  - after `<gap reason="ellipsis"/>`, as a rule replicate the unintelligible text wrapped in the element `<foreign>` which is in turn wrapped in `<supplied reason="explanation">`
    - markup pertaining to the replicated text may be used as per §10.1
    - if the unintelligible text is very long (e.g. an entire paragraph), you may optionally forego replicating it; in this case the `<note>` element mentioned in the next point is mandatory
  - optionally, after the `<supplied>` element, add a `<note>` with any explanation you deem appropriate
- when **a segment of extant text is not translated for any other reason** (for instance because it is considered too trivial to translate), this shall be indicated in the following way

– mandatorily create `<gap reason="ellipsis"/>` without any further attributes to indicate the place in the translation where text is skipped
  – such a gap element in a translation will be displayed as … (in the same way as for unintelligible text, but not followed by the text in the original language)
– preferably, after the `<supplied>` element, add a `<note>` with an explanation of why the text has not been translated

9.2.13. Blank space in the translation

– in general, spaces encoded in the edition (§4.3) should not be preserved in the translation
– however, spaces left blank in the original with the intent of subsequent filling (*vacat*, §4.3.3) may be preserved in the translation if you feel that this serves a useful purpose
– in this case, use the `<space/>` element exactly as in the edition division, e.g. `<space type="vacat" quantity="3" unit="character"/>`
  – this will probably be displayed as text in square brackets, e.g. *[space of ca. 3 characters left blank]*

9.2.14. Indicating bitextuality

– to **indicate bitextuality** (*śleṣa*) in your translation, select one translation of the double entendre as the primary or more literal one (to leave without markup), and wrap the translation of the secondary or less literal meaning in `<seg rend="pun">`
  – this will be displayed as {} curly braces around the segment thus tagged
  – this encoding will not be machine-actionable and will in many cases leave some ambiguity that will have to be resolved by the reader, but we do not perceive a need for a more rigorous (and thus more complex) encoding scheme

Example 9.2.14.A: translation with bitextual words dispersed across a stretch of text

```
... who make the ocean heave <seg rend="pun">the treasured water burst forth</seg> with the powerful wind <seg rend="pun">vital breath</seg> arising from the lute <seg rend="pun">ritual procedure</seg> ...
```

➢ display: … who make the ocean heave {the treasured water burst forth} with the powerful wind {vital breath} arising from the lute {ritual procedure} …

Example 9.2.14.B: translation with bitextual interpretation integrated into one sentence

```
May this reservoir of water, which is eternally festive because it is surrounded by the bodies of many riverlike women, never become exhausted just as the great ocean <seg rend="pun">which eternally revels in bodily union with many women who are rivers, yet never contracts the clap</seg>.
```

➢ display: May this reservoir of water, which is eternally festive because it is surrounded by the bodies of many riverlike women, never become exhausted just as the great ocean {which eternally revels in bodily union with many women who are rivers, yet never contracts the clap}.

9.3. The Commentary

9.3.1. Overview

– the commentary to your text shall be wrapped in `<div type="commentary">`
  – this division follows the edition division containing the translation(s)
– the contents of the commentary division shall be freeform discursive English text, wrapped in one or more `<p>` elements, which may include globally permitted miscellaneous markup (§10), but no other markup including non-XML markup such as brackets, asterisks and other signs
– possible topics of the commentary include:
  – discussion of the readings chosen for your edition, along with any details that could not be encoded within the edition or the apparatus, including
    – alternative readings too nebulous to encode in the edition
    – uncertainty about the location of a line break with respect to restored text (§3.2.3)

- discussion of metrical phenomena and uncertainty about verse metres
- discussion of the interpretation as reflected in your translation and any alternatives
- literal translations of phrases more elegantly translated in your translation, but for this reason, possibly obscure
- pointing out parallel passages in other sources, especially if these are used as the basis of restoration in the present text
- note that palaeographic observations should not go into the commentary; rather,
  - those pertaining to the inscription as a whole belong in the `<handDesc>` section of the TEI header, see §11.2.1
  - those pertaining to specific loci should be recorded in the apparatus as notes (§9.1.7)

9.3.2. Structure of the commentary and correspondence to the text

- commentarial **paragraphs will not be linked** in a machine-actionable way **to the text**
  - as in any written commentary, refer to lines, stanzas, *pāda*s or particular words/phrases as and when necessary, spelling out such references in a clear human-readable manner
  - however, should you wish to create a structured commentary with entries referred to particular sections of the text, you may employ the human-readable linking method described for translations in §9.2.6 above, i.e.
    - add to any `<p>` element in the commentary the attribute `@n`, corresponding to a line number, a range of line numbers, or a set of non-contiguous line numbers in the text
    - add to any `<p>` element in the commentary the attribute `@rend` with the value `"stanza"` and the attribute `@n`, corresponding to a stanza number, a range of stanza numbers, or a set of non-contiguous stanza numbers in the text
    - see §9.2.6 for details and examples of this method
- **textpart divisions** (`<div type="textpart">`) may be created to break up a long commentary into **sections** (chapters), but there is no obligation to reproduce the textparts of the edition in the commentary and no facility of linking such divisions to any divisions of the text
  - use any arbitrary number of textpart divs
    - keeping in mind the requirement of tessellation (§3.4.2), i.e. that if a textpart division is present within your commentary, then all your commentarial text must be contained within textpart divisions
  - always add the attribute `@n` to number commentarial textparts, using plain Arabic numerals as values, which shall only be used for internal reference if at all
  - always add a header to commentarial textparts, by creating the element `<head>` (containing free text to be displayed as a heading) as the first item within each textpart division

9.4. The Bibliography

9.4.1. Overview

- the project will maintain a master bibliography in Zotero
  - consult the ZG about adding entries to this bibliography, and if you do not yet have access to the group Library DHARMA, ask for it
  - in this system, each reference will be known by a unique internal identifier in the form of the Short Title assigned to any Zotero item
- the bibliography division of an XML document in our project serves a twofold purpose
  - to present what specialists of Greek and Roman epigraphy call the “epigraphic lemma”, i.e. a paragraph which explains the history of research leading up the edition encoded in your file
  - to collect all bibliographic references pertaining to the inscription edited in your document and the artefact bearing it
- the XML shall be created within `<div type="bibliography">`
  - this division is the last element within the `<body>` of your document, appearing after the commentary

9.4.2. The structured bibliography

– the structured bibliography will be divided in our editions into two sections, a primary and a secondary bibliography
– the primary bibliography shall include only independent integral editions
– the secondary bibliography shall contain all other publications, such as
  – re-publications of previous editions (without new insights)
  – reports (ARIE, etc.)
  – partial readings or translations
  – descriptions/discussions of the inscription's content, the support, specific readings, etc.
– the encoding of these two bibliographies consists of the containers `<listBibl type="primary">` and `<listBibl type="secondary">`,
  – and within each of these, one `<bibl>` element for each bibliographic entry, encoding a regular citation as per §10.4.5
– these lists will be populated automatically from your metadata spreadsheets, but will require your attention on the following points:
  – check them and correct where necessary
  – as far as possible, arrange them in an alphabetical order

9.4.3. Bibliographic sigla

– if your edition includes an apparatus that cites readings from previous editors, the citations of those editions in your bibliography must include a manually encoded siglum (an abbreviation by which they will be shown in the apparatus)
  – this applies regardless of whether the citation is in your primary bibliography or the secondary one
  – if the citation appears in more than one place within your file, it is sufficient to encode a siglum on one of these appearances
– encode the siglum as the attribute `@n` on the relevant `<bibl>` item
– as a rule, the siglum shall consist of all the initials of the author of the edition (in the order applicable to the language of the publication), but you may deviate from this in the following cases
  – should there happen to be two cited editions by the same author or by two authors with identical initials, in order to eliminate ambiguity you may extend the siglum with a number (e.g., referring to the respective years of publication) or with a full name
  – should the author have a multipartite name that would lead to a siglum of more than three letters, or should a cited edition be the product of several authors, you may reduce the siglum to the initials of the surnames

Example 9.4.3.A: bibliographic citation with a siglum, within the primary bibliography

```
<bibl n="EH">
  <ptr target="bib:Hultzsch1913-1914_01"/>
    <citedRange unit="page">225-226</citedRange>
  <citedRange unit="item">B</citedRange>
</bibl>
```

9.4.4. The epigraphic lemma

– in addition to the structured bibliographies, you will need to create an epigraphic lemma
– this shall take the form of a single `<p>` element located within the bibliography division and before the primary bibliographic list, containing freeform discursive English text
  – your freeform text should contain bibliographic citations (encoded as per §10.4.5) for entries pertaining to the study of the text and its translation
    – these can be copied and pasted from the structured bibliography, then edited as needed and expanded with explanatory text
– the epigraphical lemma should mostly consist of items of the primary bibliography, but early reports, facsimiles and translations without an accompanying edition may also be mentioned here

– it is recommended that you present the principal publications in ascending chronological order up to the one that most immediately precedes the present edition
– each item referred to should be accompanied by brief information on why each bibliographic item is relevant and on whether it includes an edition, a facsimile or a translation of the text
– these citations should include page ranges only if the publication is not entirely or mostly about the text being edited
  – for instance, when citing a journal article primarily concerned with the text, refer to the article as a whole and not specifically to the page range containing the edited text
– if it is known from a publication, or if you are the author of the present edition, include a brief statement of the principal visual documentation that has been used
– note that you must identify the author of the present edition in the epigraphic lemma, even if no previous publication exists
  – the same author must also be identified in `<respStmt>`, see §11.1.2

9.4.5. Full markup example for the bibliography

Example 9.4.5.A: the bibliography division

```
<div type="bibliography">
  <p>First edited by Cohen Stuart <bibl rend="omitname"><ptr
target="bib:CohenStuart1875_01"/><citedRange unit="page">23</citedRange><citedRange
unit="item">XIII</citedRange></bibl> with a lithographic reproduction in the accompanying volume
of plates <bibl><ptr target="bib:Huart+Hooiberg1875_01"/></bibl>; edited again by Boechari <bibl
rend="omitname"><ptr target="bib:Boechari1985-1986_01"/><citedRange unit="page">53</citedRange>
<citedRange unit="item">E.16</citedRange></bibl>; re-edited here by Arlo Griffiths from the
Leiden estampage of the plate.
  </p>
  <listBibl type="primary">
    <bibl n="B">
      <ptr target="bib:Boechari1985-1986_01"/>
      <citedRange unit="page">53</citedRange>
      <citedRange unit="item">E.16</citedRange>
    </bibl>
    <bibl n="CS">
      <ptr target="bib:CohenStuart1875_01"/>
      <citedRange unit="page">23</citedRange>
      <citedRange unit="item">XIII</citedRange>
    </bibl>
    <bibl n="HH">
      <ptr target="bib:Huart+Hooiberg1875_01"/>
    </bibl>
  </listBibl>
  <listBibl type="secondary">
    <bibl><ptr target="bib:NBG"/><date>1870</date><citedRange
unit="volume">8</citedRange><citedRange unit="page">72, 78</citedRange></bibl>
    <bibl><ptr target="bib:Verbeek1891_01"/><citedRange unit="page">149-150</citedRange>
<citedRange unit="item">276</citedRange></bibl>
    <bibl><ptr target="bib:Damais1970_01"/><citedRange unit="page">48</citedRange><citedRange
unit="item">86</citedRange><citedRange unit="note">13</citedRange></bibl>
  </listBibl>
</div>
```

10. Globally Available Markup Outside the Edition

10.1. Editorial Markup Outside the Edition

– as a rule, the XML elements pertaining to the text edition (§2 to §7) should be avoided outside the edition division of your XML file, except as explicitly endorsed in the following sections of §10 and, for specific divisions of the XML file (viz. the apparatus and the translation), in the relevant sections of §9
– when citing something from the text edited in your file or from another text, it is generally recommended that you omit editorial markup from that citation
– however, sometimes you may deem it essential to cite a diplomatic reading with all its intricacies, particularly
  – in a note or commentary section discussing the editorial difficulties connected to a reading, or
  – in a translation when replicating an unintelligible stretch of text (§9.2.12),
– in such cases, you may use the following elements of editorial markup outside the edition division, limited to features whose replication is deemed essential in the context (i.e. without an obligation to replicate all editorial markup)
  – empty elements representing transition points (`<lb/>`, `<pb/>` and `<milestone/>` of any kind; §3.2, §3.5 and §3.6)
    – to reduce code clutter, feel free to remove all attributes from these elements
  – any markup pertaining to the originally inscribed text (§4)
  – any markup pertaining to physical condition and reading difficulties (§5)
  – any markup pertaining to modern editorial intervention (§5.5)
  – any markup pertaining to visual features (§7.5)
– when citing primary text with editorial markup as above, keep in mind that
  – tags for block-level containers (`<p>`, `<ab>`, `<lg>` and `<l>`) must not be included in citations
  – foreign-language citations within a stretch of modern-language text must always be tagged as `<foreign>` (§10.3.3), so all markup pertaining to the citation must be within these tags
  – XML elements must always have a start-tag and an end-tag, so when copying and pasting from your edition, make sure that these tags are present in your citation even if one end of the segment tagged in your edition is outside the copied string, i.e.
    – add the start-tag for retained markup commencing before and ending inside your citation
    – add the end-tag for retained markup commencing inside your citation and ending after it
    – add start and end-tags for a citation snipped from within a longer stretch of phrase-level markup

10.2. Formatting

10.2.1. Character formatting

– given the principle of conceptual markup (§1.3.4), it will not normally be necessary for you to apply character formatting as such: all essential formatting will be handled globally through external stylesheets and governed by the XML tags you create,
  – so for instance instead of italicising foreign words and titles, you tag them as `<foreign>`[45] or `<title>` respectively
– that said, you may occasionally find it useful to encode simple formatting instructions without any specific semantic classification
– for this purpose, only outside the edition[46] and mainly within the commentary division, you may use the element `<hi>` (signifying typographic highlighting) with the attribute `@rend` taking on one of the following values:
  – `"italic"`
  – `"bold"`

[45] Note that this tag is also applicable to short segments of text meaningless in and of themselves, see §10.3.3.
[46] See §7.5.1 about the use of `<hi>` in the edition.

  - `"subscript"`
  - `"superscript"`
- in addition to the above, the use of `<hi rend="grantha">` is permitted outside the edition in text cited from the inscription (or another primary text), for the purpose of highlighting characters originally written in Grantha
  - note that this does not replace the `<foreign>` tag (§10.3.3) required for such text outside the edition

10.2.2. Lists

- should you need to format some text as a structured list, the following markup may be used in a commentary or in a translation (but not in any other section of your document)
- within the `<p>` element enclosing your text, create the container `<list>`
  - without any attributes to produce a plain list (displayed with each item in a new line and indented)
  - or with the attribute `@rend` with the values `"bulleted"` or `"numbered"` for these list styles
- within `<list>`, create an `<item>` element as a container for each list item
- to create a multi-level list, `<list>` elements may be nested in one another, but this is not recommended (especially not for numbered lists); please contact the authors and the TEI manager if you feel this is essential for you

10.3. Encoding Language

10.3.1. Tagging language with `@xml:lang`

- whenever encoding a language with the attribute `@xml:lang`, we shall use the **language codes** defined by the ISO standard 639-3[47]
  - the codes relevant to our project are listed in Appendix D of this guide
- **script codes** defined by ISO 15924[48] are often conjoined to language codes (using a hyphen between the two), but in our practice, this shall be limited to the following
  - the language code of text originally written in an Indic script and edited in Romanised transliteration shall mandatorily be suffixed with "-Latn"
    - since the languages we work with are always transliterated throughout our editions, apparatuses and commentaries, you will thus in most cases have to suffix "-Latn" to your language codes
- language tags without a script code will by default be assumed to be in a native script associated with the language in a given region and time period
  - this is applicable to modern languages as well as to the languages of our corpus
  - specific details about the native script(s) used in an inscription shall be recorded in our TEI headers
  - bilingual (or multilingual) inscriptions using different scripts for different languages require no explicit indication of the scripts in the edition, provided that the TEI headers clearly state what script is used for which language
  - should an inscription use more than one script for one language, script changes (§7.5.4) shall be encoded as applicable
- the subsections below concern language encoding outside the edition division
  - see §7.2 for specific instructions on language encoding within the edition

10.3.2. Tagging language in pre-existing containers

- the `<TEI>` container (§1.4) of the entire XML document specifies English as the language of the document, and this is understood to apply to all child elements unless otherwise specified
- when **a structural unit** of the document is in another language, the attribute `@xml:lang` must be added to the start-tag of the corresponding structural element to specify the language:
  - the edition division must always be set to the language of the original text (§1.4)
  - translation divisions must explicitly indicate their language (§9.2.1) even if it is English

[47] https://iso639-3.sil.org/
[48] https://en.wikipedia.org/wiki/ISO_15924

– similarly, smaller structural units (e.g. `<note>` or `<p>`) may be set to a particular language as and when applicable

10.3.3. Tagging foreign languages outside the edition

– this subsection concerns short stretches of a language different from that of the surrounding English (or other modern-language) text (for the sake of simplicity referred to here as a "foreign" language)
  – see §7.2 about tagging language within the edition
  – stretches of foreign language that coincide with an already existing XML container should be handled as per §10.3.2 above
– text in a foreign language shall be wrapped in the element `<foreign>`, and will be displayed in italics
– text tagged as `<foreign>` outside an edition need not always carry the attribute `@xml:lang`
  – `@xml:lang` **is not required** (and, to reduce code clutter, counter-recommended) for text in the language of a monolingual inscription (i.e. that encoded for the edition division), including
    – citations from the inscription in a translation, commentary or note (including apparatus notes), e.g. `Two <foreign>kulyavāpa</foreign>s of land`
    – strings of text that are not meaningful in and of themselves, such as
      – single (transliterated) characters mentioned in a palaeographic description, e.g. `The scribe tends to use <foreign>ṅh</foreign> instead of <foreign>ṁh</foreign>`
      – morphological components or unintelligible segments mentioned in a discussion, e.g. `The suffix <foreign>-vat</foreign>`
  – `@xml:lang` **is optional** (to be used or avoided on a case-by-case basis)
    – for technical terms in Sanskrit or another major language applicable to the inscription's context, e.g. `a <foreign>bahuvrīhi</foreign> compound`
    – for text in any language of a multilingual inscription (i.e. those encoded for certain textpart divisions or smaller sections of the edition)
  – `@xml:lang` (with the appropriate language and script code) **is mandatory**
    – for words or phrases in a modern language other than that of the surrounding text (e.g. a French quotation in an English commentary)
      – however, do not overdo the tagging of modern foreign words: e.g. French or Latin terms commonly used in English should not be tagged at all

10.4. Notes, Quotations and References

10.4.1. Encoding notes

– notes may be used in the apparatus, the translation and the commentary
  – it is strongly recommended that you add no notes to any other part of your document; if you feel an overwhelming need to do so, please first discuss this with the authors of this Guide and the XML-TEI data manager
– to create a note, add the element `<note>` at the point where the note should be anchored
  – notes may be rendered as footnotes, endnotes or tooltips (attached to a note anchor on the spot), depending on display decisions which will be made later
  – `<note>` elements must always be within the structural containers applicable to the division, i.e. within `<p>` in a translation or commentary, and within `<app>` in an apparatus
  – in freeform text (i.e. in a translation or commentary), place the `<note>` element after any adjacent punctuation mark, not before it
– if the author of a note is not the same as the author encoded in the `@resp` or `@source` attribute of the note's ancestor element (e.g. the translation division), then authorship must be encoded for the note as follows:
  – using `@resp` (§10.6.1) if the note is by you and/or another project member
  – using `@source` (§10.6.2) if the note is adopted verbatim from a publication

  - paraphrased notes are in this respect regarded as the product of the person doing the paraphrasing; the author of the original note shall be credited by including a regular citation (§10.4.5) in the paraphrase or, if the reference is obvious from the context, simply by referring to the original author by name
- the **content of notes** shall be freeform text, preferably consisting of complete sentences in English, starting with a capital letter and ending with punctuation
  - notes may contain any phrase-level markup permitted in freeform text, including in particular bibliographic citations, which should be added wherever you refer to a published opinion
  - it is recommended that you keep your notes short, but should you find it absolutely necessary, the contents of notes may be structured into paragraphs by creating `<p>` elements within `<note>`
    - in this case, the entirety of your note text should be contained in `<p>` elements

10.4.2. Encoding titles

- outside the edition division (e.g. in notes, commentary, etc.), any titles you mention shall be tagged with the element `<title>`
- titles to be tagged in this way
  - include non-epigraphic primary sources (literary texts), e.g. `<title>Harivaṁśa</title>`
  - include secondary sources (technical literature, where you mention a title outside a citation encoded as per §10.4.5), regardless of whether the title is cited
    - in full, e.g. `<title>Early History of the Deccan</title>`,
    - in abbreviated form, e.g. `<title>Mahâbodhi</title>` for the title *Mahâbodhi, or the great Buddhist temple under the Bodhi tree at Buddha-Gaya*, or
    - as a widely known acronym, e.g. `<title>OJED</title>` for the title *Old Javanese-English dictionary*
  - do not include the titles of inscriptions (in secondary literature or the DHARMABase), which are to be encoded as per §10.4.6
- by default, all titles tagged in this way will be displayed in italics; when this is not desired, do the following
  - for titles of chapters (e.g. in a multi-author book) and articles (e.g. in a journal)
    - add the attribute `@level` with the value `"a"` (for “analytic”)
      - e.g. `<title level="a">Grants from Valabhî</title>`
    - titles tagged in this way will be displayed in regular type, but with quote marks added around them
  - for any titles that you wish to display without italics and that are not chapters or articles
    - add the attribute `@rend` with the value `"plain"`
    - titles tagged in this way will be displayed without any typographic distinction from the surrounding text

10.4.3. Quotations without an encoded reference

- **quoted text not attributed to a published source** shall be wrapped in the element `<q>`
  - do not add quotation marks to the text, as these will be automatically produced in display in the correct form
  - however, to exercise more control over the type of quote marks displayed, you may choose to omit the tag and add marks manually
    - in this case take care to use the desired characters („…“ “…” ‘…’ «…») rather than generic typewriter quotes or apostrophes
- this encoding method applies primarily to quotations in the same language as the surrounding text, such as
  - translations of phrases or sentences of the edited text or another text (appearing in a commentary or apparatus)
  - your translation of text quoted as direct speech (e.g. with *iti*) in an inscription (within the translation)
- if you use `<q>` to quote text in a modern language other than that of the surrounding text, use `@xml:lang` on the `<q>` element, as per §10.3.2

– text quoted from a primary source in the original language, to be displayed in italics without quotation marks, shall not be marked up in this way, but must be tagged as `<foreign>` as per §10.3.3

10.4.4. Quoting published material

– to encode **direct quotations from a published source**, proceed as follows
  – within the `<p>` element in which you quote some text, create the wrapper `<cit>`
  – within `<cit>`, wrap the quoted text in the element `<quote>`
  – also within `<cit>`, but outside (and immediately after) `<quote>`, add a bibliographic citation as per §10.4.5 to specify the source of the quotation[49]
– do not add quotation marks to the text, as these will be automatically produced in display in the correct form
– to create a **block quote**, proceed as above but add `@rend="block"` to the `<quote>` element
  – text quoted in this way will be displayed as a separate indented paragraph without quotation marks
  – when citing text from a publication with `<quote>`, any **transliterated words** in the cited text shall be retained **in their original form** rather than being converted to our transliteration system
  – but any text adopted without `<quote>`, for example in apparatus readings, translations and paraphrased/summarised opinions, should be converted to our transliteration system

10.4.5. Bibliographic citations

– bibliographic citations may be used in any part of your XML document where modern-language text is permitted, and must be mandatorily listed in the bibliography as discussed under §9.4
– a citation is encoded in the form of the element `<bibl>`
– the empty element `<ptr/>` (pointer) must mandatorily appear as the first element within `<bibl>`
  – with the mandatory attribute `@target`, whose value shall be the Zotero Short Title of the cited publication, prefixed with the string “bib:”
    – e.g. `<bibl><ptr target="bib:Agrawala1983_01"/></bibl>`
– **to limit the citation** to a specific part of the publication, add the element `<citedRange>` after the `<ptr/>` element but within the `<bibl>` element, wrapping the details of the citation in the following manner
  – by default, `<citedRange>` will be understood to refer to page numbers, so references to pages without additional detail can simply be added as the content of this element, e.g. `<citedRange>12</citedRange>`
    – use a hyphen to record a range of pages, e.g. `<citedRange>12-21</citedRange>`
      – the number of the last page should always be recorded in full, e.g. `<citedRange>123-124</citedRange>` (not 123-4 or 123-24)
    – use a comma (followed by a space) to list non-adjacent pages, e.g. `<citedRange>12, 24</citedRange>`
  – to refer to an entity other than a page, add the attribute `@unit` to `<citedRange>` with one of the following values
    – `"page"` for page numbers where this needs to be made explicit (see below)
    – `"part"` to distinguish sections of publications where identical page numbers re-occur within a single volume (such as Epigraphia Carnatica)
    – `"volume"` for multi-volume publications
    – `"note"` for a numbered (foot or end) note
    – `"item"` for a number in an anthology of editions, or an item in a numbered list (to be displayed as №)
    – `"entry"` for an entry in a dictionary or encyclopaedia (to be displayed as *s.v.*)
    – `"figure"`, `"plate"`, `"table"`, `"appendix"` etc. as applicable, for visual material

[49] It is, in principle, possible to add `@source` (§10.6.2) to a `<quote>` element, but this would not be sufficient in many cases, because it does not allow referring to a page. We shall therefore refrain from doing so and always use a full citation.

    - note that the values of `@unit` should always be English regardless of the language of the publication, e.g. use `"appendix"` for an original *bijlage*
    - should you feel the need to use a different value, please contact the authors and the XML-TEI Data Manager to discuss the matter
    - as per ZG §4.4 and §4.6, numerals other than Arabic ones (e.g. Roman and Devanagari) should be converted in your citations to Arabic numbers unless this results in an ambiguity (because Arabic and non-Arabic numerals are both used within a publication, for the same unit of citation, e.g. Roman page numbers in the front matter and Arabic page numbers in the main text)
  - where necessary, it is possible to add more than one `<citedRange>` element (e.g. to encode a reference to a certain figure on a certain page)
    - in this case, page references must be explicitly encoded with `@unit="page"`
    - the `<citedRange>` elements must appear in the order in which they are eventually to be displayed, e.g. a volume number must precede a page number and a page number must precede a note number
  - see the examples below for the use of `<citedRange>` in various combinations
- **to add a year to a citation** where the corresponding Zotero entry does not include one (in serial publications recorded as a single Zotero entry, ZG §4.4)
  - include a `<date>` element containing the year, at the point where it should appear in the citation (within `<bibl>`, but not within `<citedRange>`)
  - see Example 10.4.5.D and Example 10.4.5.E below
  - note that not all serial publications will be recorded as a single entry in our Zotero library; see Example 10.4.5.H for an illustration involving ARIE, where each volume is recorded separately
- a citation encoded in this way will be ultimately **displayed** as a human-readable author-date citation
  - the internal details of citations will be automatically styled according to project conventions (with some details yet to be finalised)
  - parentheses will not be automatically produced around citations and will have to be added in the surrounding text wherever you need them
  - if **the name of the author(s)** is an integral part of your text and must thus appear **independently of the citation**:
    - add the attribute `@rend` with the value `"omitname"` to the `<bibl>` element, which will cause the pointer to display without the name
    - add the author's name wherever you require in the text outside the `<bibl>` element
    - see Example 10.4.5.F below for an illustration
  - if you wish to **show "ibid." instead of the name of the author(s)**:
    - add the attribute `@rend` with the value `"ibid"` to the `<bibl>` element, which will display "ibid." instead of the name and will not use parentheses
    - depending on your context, parentheses may be avoided altogether or used further away from the citation
    - see Example 10.4.5.G below for an illustration

Example 10.4.5.A: encoding a basic citation

➢ Majumdar 1943: 23–28

```
<bibl>
  <ptr target="bib:Majumdar1943_01"/>
  <citedRange>23-28</citedRange>
</bibl>
```

Example 10.4.5.B: encoding a citation with a page and a figure number

➢ Majumdar 1943: 23–28, fig. 12

```
<bibl>
  <ptr target="bib:Majumdar1943_01"/>
  <citedRange unit="page">23-28</citedRange>
  <citedRange unit="figure">12</citedRange>
</bibl>
```

Example 10.4.5.C: encoding a citation with a volume and page number

➢ Majumdar 1943, vol. 1: 23–28

```
<bibl>
  <ptr target="bib:Majumdar1943_01"/>
  <citedRange unit="page">23-28</citedRange>
  <citedRange unit="figure">12</citedRange>
</bibl>
```

Example 10.4.5.D: encoding a citation of a serial publication without a named author

➢ NBG 8, 1870: 72, 78

```
<bibl>
  <ptr target="bib:NBG"/>
  <citedRange unit="volume">8</citedRange>
  <date>1870</date>
  <citedRange unit="page">72, 78</citedRange>
</bibl>
```

Example 10.4.5.E: encoding a citation of an appendix of a serial publication without a named author

➢ OV 1918: Q, 169

```
<bibl>
  <ptr target="bib:OV"/>
  <date>1918</date>
  <citedRange unit="appendix">Q</citedRange>
  <citedRange unit="page">169</citedRange>
</bibl>
```

Example 10.4.5.F: encoding a citation with parentheses only around the year and pages

➢ Majumdar (1943: 23–28)

```
Majumdar <bibl rend="omitname">
  <ptr target="bib:Majumdar1943_01"/>
  <citedRange>23-28</citedRange>
</bibl>
```

Example 10.4.5.G: encoding a citation with *ibid.*

➢ ibid.: 23–28

```
<bibl rend="ibid">
  <ptr target="bib:Majumdar1943_01"/>
  <citedRange>23-28</citedRange>
</bibl>
```

Example 10.4.5.H: encoding a citation of the Annual Report on Indian Epigraphy

```
<bibl>
  <ptr target="bib:ARIE1962-1963"/>
  <citedRange unit="page">157</citedRange>
  <citedRange unit="appendix">C/1945-1946</citedRange>
  <citedRange unit="item">1</citedRange>
</bibl>
```

➢ please always follow this example when citing the ARIE appendices, i.e. always include the year (or range of years written out in full) mentioned in the title of the appendix, separated with a slash from the letter of the appendix

10.4.6. Referring to inscriptions in the DHARMABase

– to refer to another inscription in the DHARMABase, use the element `<ref>`
– such references may be included in any freeform text, but will at the present stage most likely to be used in an apparatus note or inside a `<p>` element in the `<div type="commentary">`
– unlike the `<ptr/>` element used in bibliographic citations (§10.4.5), `<ref>` is not an empty element: it must contain a human-readable reference

  - we recommend keeping the contents limited to the identifier of the inscription you want to quote
  - any specification of the line (or other details) to which you are referring may be added after the `<ref>` element, e.g. `<ref>C. 7</ref>, line 5`
- in order for this markup to allow us to generate a hyperlink to the intended edition in the online presentation of the DHARMABase, the attribute `@target` has to be used to establish the link to the relevant xml file
  - for files kept in the same repository, the value of this attribute shall be the filename of the inscription, e.g. `<ref target="C00007.xml">C. 7</ref>`
  - for files kept in different repositories, add a further attribute `@n`, containing the name of the GitHub repository where the file is located, e.g. `<ref n="tfa-pallava-epigraphy" target="Pallava00001.xml">Pallava 1</ref>`
- the same method can also be used to create a link toward an external database
  - in this case, the value of the `@target` element should contain a permanent URL

10.5. Encoding Names

- see §7.4 about the optional encoding of names within the edition
- names (contemporary or pre-modern) may, in principle, be tagged anywhere in a document, but we do not recommend doing so in any content except where explicitly called for in another section of this guide
  - at present the Responsibility Statement (§11.1.2) is the only section which calls for name markup outside the edition

10.5.1. Tagging contemporary names

- when a contemporary name requires a tag, wrap the entire name in the element `<persName>`
  - if the name is that of a DHARMA project participant (such as your own name), add the attribute `@key`
    - the value of this attribute shall be the personal identifier[50] of the participant, with the prefix "part:" (as an abbreviated reference to the file listing participants of the project)
- within this element
  - either apply the tags `<forename>` and `<surname>` to the respective components of the name
  - or apply `<name>` to the whole of a name if it cannot be broken down in this way
    - note that in this case `<persName>` must still wrap `<name>`

Example 10.5.1.A: encoding the name of a project participant

```
<persName key="part:argr">
  <forename>Arlo</forename>
  <surname>Griffiths</surname>
</persName>
```

10.6. Attributes as Referencing Systems

10.6.1. Encoding authorship with `@resp`

- the attribute `@resp` (for "responsibility") may be added to any XML element to encode the fact that a particular project participant is the author of that particular item, without explicitly writing their name in the text (for which see §10.5.1)
- in the initial stages of our project, most of our XML documents will be the products of a single individual or a small number of people, who will be recorded as authors in the TEI header (§11.1.2) of each file
  - therefore, this attribute will only be necessary where specifically called for in this guide, namely
    - to explicitly encode your authorship for translations (§9.2.1)

[50] Our personal identifiers are available at https://github.com/erc-dharma/project-documentation/blob/master/DHARMA_IdListMembers_v01.xml

  - to encode the authorship of individual notes in a translation by someone other than the author of the note (§9.2.4)
- later on, however, many of our documents will probably be revised and improved by other project members
  - to facilitate the tracking of such revisions and to have a record of credit, `@resp` may be added to any element
- the value of `@resp` shall be the personal identifier of a project participant[51] with the prefix "part:"
  - to credit more than one participant, simply include several personal identifiers (each prefixed as above) within a single `@resp` attribute, separating them with nothing but a space

10.6.2. Crediting publications with `@source`

- the contents of certain markup elements as a whole may need to be credited to a publication, including in particular
  - lemmas and/or readings in a critical apparatus (§9.1.3, §9.1.4)
  - notes (§10.4.1)
  - translations as a whole (§9.2.1)
- to credit a previous publication, the attribute `@source` must be added to the XML entity representing the item you wish to credit
- the value of `@source` shall be the Zotero Short Title of the publication containing the reading you are crediting, prefixed with the string "bib:"
  - do not include additional reference details such as a page number or an item number in a compilation: this cannot be done in this referencing system
  - wherever a page or item number is essential (because your XML document does not include a full citation of the publication concerned), you will need to use a full citation (§10.4.5) in your text instead of the attribute `@source`
- to credit more than one publication (e.g. because more than one scholar has suggested or endorsed a certain reading, or because a note was published in several publications)
  - the relevant prefixed short titles must appear, in chronological sequence (earlier publications precede later ones), within a single `@source` attribute, separated by nothing but a space, e.g. `<rdg source="bib:Devadatta1863_01 bib:Doe2019_01">`

10.6.3. Identifying persons and places with `@key`

- to encode the common (standard) name of a person or place designated in your text by an alternative name, add the optional attribute `@key` to the `<persName>` or `<placeName>` element, recording the standard name as the value of this attribute, e.g.
  - `<persName key="Śiva">Pināk in</persName>`
  - `<placeName key="Pāṭaliputra">Kusumapura</placeName>`
- this attribute may be used whenever you feel that a name needs identification or disambiguation,
  - in conjunction with the attributes for classifying names discussed above; or
  - with a `<persName>` or `<placeName>` element created solely for the purpose of adding this attribute
- the use of this attribute does not produce a fully machine-actionable encoding and is intended as a first step toward the possible eventual creation of a prosopography and gazetteer
  - to this end, the values used in the corpus may be harvested at a later time for standardisation and verification, which may be followed by replacing this attribute with a fully machine-actionable linking mechanism

10.6.4. Identifying elements with `@xml:id`

- the attribute `@xml:id` may be used to assign a unique identifier to any XML element

[51] Our personal identifiers are available at https://github.com/erc-dharma/project-documentation/blob/master/DHARMA_IdListMembers_v01.xml

– at the present stage, the use of `@xml:id` is prescribed by this guide only for a few situations, but we shall probably use this attribute more extensively in the future
– with this in mind, we prefer to make all our XML identifiers unique *across the project's corpus*, even though many practical applications of this identifier require only that it be unique within a particular document
– to achieve this, an `@xml:id` shall in our project always begin with the **filename** (without extension) of the document, and be followed by the specific identifier of the item in question, with an underscore character separating the two
– for example, to create XML identifiers for hands numbered "hand1" and "hand2" in the file Pallava00001.xml, use "Pallava00001_hand1" and "Pallava00001_hand2"

10.7. Punctuation and Style in Modern Languages

– in general, observe the conventions of whichever modern language you are writing in and avoid imposing the conventions of another language
– when writing in French, it is not necessary to create the *espace insécable devant ponctuation*, which can be added automatically later on

11. The TEI Header

– the TEI header presents marked-up metadata about the XML document and about the inscription and artefact(s) it concerns
– the header may be composed of several high-level elements, the most prominent of which is the File Description
– the sections below outline the header elements used in our project and their contents
– at the present stage (as of May 2020), we encode only a bare minimum of data directly within the TEI header
– the guidelines below are intended to help you understand the functions and structure of the TEI header, but you need not be able to create such a header from scratch
– instead, rely on the most recent version of the project's EpiDoc template[52] and add data to the header only where comments in the template instruct you to do so

11.1. Describing the XML Document

– the mandatory File Description is enclosed in the element `<fileDesc>`, which precedes a description of the original document
– in our practice, the mandatory contents of the File Description shall be as follows
  – a Title Statement, wrapped in the element `<titleStmt>`, with the following items
    – information about the title of the digital document
    – information about the persons responsible for its content
  – a Publication Statement, tagged as `<publicationStmt>` and serving to group together information concerning the publication of the digital document

11.1.1. The title

– the contents of the `<title>` element shall be plain text in English, without any additional markup
– this title will also be used in the web publication of the digital edition
– use a title that clearly and unambiguously identifies the inscription
  – see Appendix E for guidance concerning title creation
– variant names applied to the inscription in question in previous publications shall be recorded in the metadata spreadsheet (and will be made searchable once imported from there into our TEI headers)

11.1.2. The responsibility statement

– after the title but still within `<titleStmt>` the wrapper `<respStmt>` is used for crediting contributors
– short descriptions of the principal roles that we wish to record are wrapped in the tag `<resp>`
– the names of the contributors are encoded with the markup introduced in §10.5.1
– follow the instructions found in the current template to fill out the contents of this statement[53]

Example 11.1.2.A: the responsibility statement

```
<respStmt>
  <resp>EpiDoc encoding</resp>
  <persName ref="part:jodo">
    <forename>John</forename>
    <surname>Doe</surname>
</persName>
</respStmt>
```

11.1.3. The publication statement

– the structure and most of the contents of this statement will be provided in our template, but you will have to add the following data as instructed by comments in the template

[52] Available under https://github.com/erc-dharma/project-documentation/tree/master/templates
[53] The precise details to be recorded here are still in flux at the time of finalising this version of the EGD. You can expect more detailed instructions either here in a future version or in the template itself.

– the place where you work in `<pubPlace>`
– the name of the file itself, encoded as an identification number in the element `<idno>` with the attribute `@type` bearing the value `"filename"`
– the name of the copyright holder

Example 11.1.3.A: the publication statement

```
<publicationStmt>
  <authority>DHARMA
       <note>This project has received funding from the European Research Council ERC under the
European Union's Horizon 2020 research and innovation programme grant agreement no 809994.
       </note>
  </authority>
  <pubPlace>Paris</pubPlace>
  <idno type="filename">Pallava00001</idno>
  <availability>
    <licence target="https://creativecommons.org/licenses/by/4.0/">
     <p>This work is licensed under the Creative Commons Attribution 4.0 Unported Licence. To
view a copy of the licence, visit https://creativecommons.org/licenses/by/4.0/ or send a letter
to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.</p>
    <p>Copyright c 2019-2025 by Emmanuel Francis.</p>
    </licence>
  </availability>
 <date from="2019" to="2025">2019-2025</date>
</publicationStmt>
```

11.2. Describing the Original Document

– the final element of the `<fileDesc>` is the source description, `<sourceDesc>`
  – this mandatory element records details of the original from which the digital text is derived
– TEI permits the use of various elements in a source description, but in the case of epigraphic documents its only child element is `<msDesc>`, signifying "manuscript description" and applicable to any text-bearing object besides manuscripts in the strict sense
– at present, you only need to encode the data explicitly called for in the subsections below
  – other metadata shall be recorded in spreadsheets for the time being and they will, at a later stage, be integrated with the TEI header through a largely automated process
– however, this Guide does frequently recommend that you discuss this or that matter 'in your metadata'
  – whenever there is no evident way to do so in a metadata spreadsheet, or if you do not have access to such a spreadsheet, or if for any other reason it is more convenient, we suggest that you insert a `<!--comment-->` at the very bottom of your file and fill it with any content that will eventually need to find its place in the TEI header of that file
  – please be aware that data stored in such comments will not be automatically moved to a spreadsheet or to a TEI header: at some point they will have to be moved manually to their proper place

11.2.1. The hand description

– basic designations of script names (Gupta Brahmi, Tamil, Grantha, Khmer, Kawi, etc.) will be recorded in our metadata spreadsheets and imported from there into our TEI Headers in due course
– if you wish to record any further **palaeographic observations** pertaining to the inscription as a whole (rather than to a specific locus), you may do so in the `<handDesc>` section of the TEI header
  – to record your observations, use the element `<p>` within `<handDesc>`, filing it with free prose
    – you may create additional `<p>` elements for a longer description
  – it is not mandatory to create content in `<handDesc>`, but if you have such information to record, do it here
  – it will normally be necessary to discuss only those elements that seem uncommon/exceptional given the general knowledge that the informed reader may be assumed to have of the script(s) in question
    – subjects that are of projectwide interest and should in general be recorded include:

- the use of any other type of vowel killer than the 'normal' *virāma/puḷḷi* (e.g., miniature/subscript consonants, see TG §3.3.1)
- the use of ornamental lettering in whole or part of the text
- you may cite examples of every phenomenon with free-text reference to the line or lines where they are found, e.g. `<p> ... Final consonants <foreign>K</foreign> and <foreign>T</foreign> are found in lines 3 and 8. ...</p>`
- also mention here any perceived similarity to hands seen in other inscriptions (using the mode of reference to other inscriptions prescribed in §10.4.6)

- in addition to a `<p>` element with free text, the `<handDesc>` element may include a **structured description of multiple hands**
  - we shall only use this method when more than one hand can be clearly identified within a single document
  - in this case, you will need to take the following steps:
    - within `<handDesc>`, wrap the `<p>` element pre-built into your template in the element `<summary>`, whether or not you have added any content inside this `<p>`
      - if you have created more than one `<p>` element here, wrap all of them together in a single `<summary>`
    - after the `<summary>` element, create one `<handNote>` element for each hand, with the mandatory attribute `@xml:id` to serve as a unique identifier for each hand (see also §10.6.4 about XML identifiers)
    - the values of this attribute shall be `"hand1"`, `"hand2"` and so on for the required number of hands, prefixed with the filename (without extension) and an underscore _ character
  - in the contents of the `<handNote>` element, write a concise, freeform description of the hand
  - e.g. `<handNote xml:id="Pallava00001_hand1">A neat hand with a tendency to use northern character forms.</handNote>`
  - once the hands have been encoded in the header as above, use `<handShift/>` within your edition to indicate hands, as described in §4.4

11.3. Keeping Track of File History

- from the moment it is created, the life-cycle of any xml file is liable to include any number of events, such as additions, updates, corrections, or transformations
  - the history of the file is to be recorded in the Revision Description, encoded in `<revisionDesc>` as the final high-level element in the TEI header
- once basic encoding has reached the first significant milestone, no further significant changes should be made in the file without a notification in the revision description
- recording changes at a manageable yet still meaningful level of detail can become an asset for the management and control of the files, for instance by helping
  - to resolve issues regarding the encoding choices that can arise when files are being edited by multiple team members
  - to gain a quick overview of the latest changes made when you return to work on a file after some time
- we therefore recommend that you always check this part of the file before resuming your work to be sure that you have a clear understanding of the state of the encoding and avoid deleting changes made by others
- within `<revisionDesc>`, create one `<change>` element for each significant change, with the following attributes:
  - mandatorily, `@who`, the value of which shall be the personal identifier[54] of the person(s) making the change, i.e. normally yours, with the prefix "part:" (as an abbreviated reference to the file listing participants of the project)

[54] Our personal identifiers are available at https://github.com/erc-dharma/project-documentation/blob/master/DHARMA_IdListMembers_v01.xml

  - to record multiple identifiers, prefix each as above and separate them by a space
- mandatorily, `@when`, the value of which shall be the date of the change in ISO format, i.e. YYYY-MM-DD
- optionally as needed, `@status`, to help keep track of significant milestones in the history of the file, with one of the following values
  - `"draft"`
  - `"candidate"`
  - `"approved"`
  - `"published"`
  - `"withdrawn"`
- the contents of `<change>` shall be a freeform description of the modification
  - please be concise, but avoid generic formulations and favour precise ones
- keep in mind that changes should be logged in reverse order, i.e. the most recent change should appear at the top of the list

Example 11.2.1.A: the revision description

```
<revisionDesc>
  <change who="part:daba" when="2019-12-10">Encoding of the translation</change>
  <change who="part:daba" when="2019-12-01" status="draft">Creation of the file and basic
structural encoding of the inscription</change>
</revisionDesc>
```

Appendices

Appendix A. Converting CII/EI Markup Conventions to EpiDoc

– word segmentation
  – **spaces** (indicating non-compound word separation) remain spaces
  – hyphens
    – used for compound segmentation between words fused in vowel sandhi (e.g. *parākkram-āṅka* = *parākkrama+aṅka*) are discarded (*parākkramāṅka*)
    – used for compound segmentation and not affected by sandhi fusion are optionally retained as per TG §2.6.2
    – inserted at the ends of printed lines (when an epigraphic line is too long to fit in one printed line) are normally discarded
      – if they also serve for compound segmentation, they may be optionally retained as above
    – inserted at the ends of epigraphic lines (when a line end is not the end of a word) are to be converted into markup by adding `@break="no"` to the following `<lb>` element (see §3.2.4/)
      – if such a hyphen also serves for compound segmentation, optionally retain the hyphen, but move it after the line beginning tag
  – **double hyphens** (or equal signs)[55] normally become spaces
    – but when used between words fused in vowel *saṁdhi* (e.g. *c=āpi*), they are discarded
– **round parentheses ()** are used in two ways:
  – with text inside, e.g. *sa* to mark an editorial correction of the text preceding the parenthetical text
    – the scope is normally the same number of *akṣara*s as there are within the parentheses (in most cases, exactly one *akṣara*)
    – the corresponding markup in EpiDoc is correction or normalisation by substitution (§6.2.2, §6.3.2), which we can apply to any segment, from a single phoneme to a longer string
  – with a question mark inside, (?) to flag the preceding text (usually one *akṣara*) as very uncertainly read
    – in our terms, this is "unclear" with low certainty (§5.3.2)
  – with text and a question mark inside, used in some publications as follows:
    – (?abc), with the question mark before the text, to indicate a possible alternative reading (§5.3.3)
    – (abc?), with the question mark after the text, to indicate a tentative emendation (which we do not encode as such; the tentativeness of an emendation encoded as per §5.5 may be indicated in an apparatus note)
– **square brackets []** are used for no less than four functions
  – 1. to wrap "letters which are much damaged and nearly illegible in the original"
    – this normally corresponds to our "unclear" category (§5.3.1), in some cases with low certainty
  – 2. to wrap "letters … which, being wholly illegible, can be supplied with certainty"
    – i.e. normally "supplied" in our terminology (§5.5)
    – it is usually not possible to distinguish 2 from 1 without studying a facsimile of the inscription; if you cannot do this and you are only transcribing a printed edition to EpiDoc, use "unclear" for both
  – 3. text followed by a question mark in square brackets, [abc?] is used by some editors to indicate tentatively or conjecturally read text
    – this usually corresponds to "unclear" with low certainty (§5.3.2)
  – 4. text followed by an asterisk in square brackets, [abc*] in principle means editorial restoration of characters omitted by the original scribe, but in the actual practice of some editors it seems to be used for function 2 above
    – for editorial correction of omissions, see §6.2.4

[55] Double hyphens are used in CII convention to indicate non-compound word separation where the end of the first and the beginning of the second word belong to the same akṣara of the original. Their primary function is to distinguish regular *akṣara*s from *halanta* consonants and initial vowels (e.g. तदाहुः > *tad=āhuḥ*; तद् आहुः > *tad āhuḥ*). We achieve this distinction by means of uppercase characters (TG §3.3.1 and §3.3.3).

  - note that earlier editors usually supply punctuation marks and numbers in stanzas, which you should not do (see §6.1.3)
  - if possible, look at a facsimile to check whether this editorial markup stands for a scribal omission or for lost and supplied text; if this is not possible, assume that square brackets with an asterisk stand for scribal omission
- 5. prosodic notation in square brackets indicates a lacuna of known metre, to be marked up as per §5.4.4
- plain transliterated text
  - bear in mind that unclear markup in EpiDoc should be used when the interpretation of a character would be ambiguous without its context, i.e. more frequently than in most earlier printed editions
  - the same applies, to a lesser degree, to restorations: if a character (or component) is completely indistinct in a good facsimile or the original support, feel free to mark it up as supplied even if the print edition shows it as merely unclear or even as clearly read
- **dots** or other signs indicating lacunae
  - CII normally uses dots for lacunae, roughly indicating their size by the rule of "two dots correspond to one *akṣara*" (single dots may mean a lost vowel or a lost consonant)
  - other editors may use asterisks or underscores, each corresponding to an *akṣara*, though this correspondence may be extremely inaccurate in some editions
  - see §5.4 about handling lacunae in EpiDoc

Appendix B. Metre (Prosody)

Looking up Sanskrit metres

- to identify the metre of a Sanskrit stanza, check the lists of syllabic and moraic metres below and use Apte's (1957) Appendix A to identify metres not listed here
- to accelerate your identification, you can try one of these online tools:
  - https://sanskritmetres.appspot.com/ requires a full stanza as input but will tolerate mistakes and lacunae and produce an approximate match
  - http://sanskritlibrary.org:8080/MeterIdentification/ recognises a very large number of metres and accepts half or quarter stanzas as input (but does not tolerate errors); it accepts several transliteration schemes but these need to be selected manually instead of being recognised automatically
- if you have identified a metre not already listed in the Table 3 below, please get in touch with the authors of this guide to add its name and pattern to the table

Syllable length

- quantitative and syllabo-quantitative verse in the languages we work with relies on the distinction of short and long syllables for producing rhythm in verse, therefore we prefer the terms "length", "long" and "short" to the corresponding triad "weight" (or "quantity"), "heavy" and "light" often used in discussions of verse in other languages
- a *mora* is defined as the duration of a short syllable, and a long syllable is always equivalent to two morae
- as a reminder, syllable length is essentially determined as follows:
  - a **short syllable** is one whose vowel is short (in Sanskrit: *a*, *i*, *u*, *ṛ*, *ḷ*) and is followed by no more than one consonant (without regard to whether a word boundary is also present)
  - a **long syllable** is one that does not meet both of the above conditions for a short syllable; specifically, a syllable is called
    - long by nature, if its vowel is long (in Sanskrit: *ā*, *ī*, *ū*, *ṝ*, *e*, *ai*, *o*, *au*)
    - long by position, if its vowel is short but is followed by two or more consonants
- *anusvāra* and *visarga* normally count as consonants in determining syllable length, but
  - in Prakrit languages, *anunāsika* (usually not distinguished in writing from *anusvāra*) indicates the nasalisation of a vowel rather than a nasal consonant following the vowel, and a nasal vowel followed by a single consonant may still count as short

  - should you encounter this phenomenon, accept the metre as legitimate, and preferably encode a normalisation (§6.3.2) of *anusvāra* to *anunāsika*
- in Sanskrit verse (though generally only in pre-classical Sanskrit), *anusvāra* may cause the preceding vowel to be long even when followed by another vowel
  - should you encounter this phenomenon in a classical metre, treat it as a metrical anomaly and add `@real` to the encoding of verse lines that exhibit it (§2.3.5)
- certain schools of versification permit some kinds of licence in determining syllable length, the most common licence being that a short vowel followed by a voiceless stop and an *r* or *l* may count as a short syllable
  - our strategy is to encode the use of such licence as a metrical anomaly in order to facilitate research
    - thus, `@real` must be added to the encoding of verse lines that employ it (§2.3.5)

Prosodic code

- the signs set out below are to be used in values of XML attributes that require prosodic notation, namely in the following contexts
  - 1, `@met` used in `<lg>` to encode the metre for which a conventional name is not available (see §2.3.4)
  - 2, `@met` used in `<seg>` to encode the prosody of a lacuna (see §5.4.4 and §5.4.5)
  - 3, `@real` used in `<l>` to encode the actual prosody of a metrically deviant verse line (see §2.3.5)
  - the final column of the table shows which of these contexts permit the use of each particular sign; the general rules are as follows
    - in the attribute `@real`, use only the signs + and - to encode the exact prosody of a metrical realisation
      - exceptionally, anceps or moraic notation is permitted in `@real` when necessitated by partial lacunae for which only the template, not the actual realisation, is known
    - in `@met`, use the notation for anceps (rather than for a long syllable) at the end of each line of syllabic verse
    - caesurae and odd/even quarter boundaries shall only be noted in context 1, and the latter only when the metre has a different template for odd and even lines (*ardhasamavṛtta*)
- prosodic code must not contain spaces
- the table also shows the equivalent conventional signs (where available), which will be used for displaying metrical notation
- when using **numbers to encode moraic feet or cola**, be aware of the following
  - numbers used in prosodic code (for moraic metre) must always be delimited by the foot boundary sign |
    - this allows multi-digit numbers to be used when necessary; however, consider whether large moraic units can be analysed into combinations of smaller feet
  - moraic feet may be constrained (e.g. the pattern ⏑–⏑ is prohibited in many tetramoraic feet), but this depth of prosodic analysis is not desirable in our encoding of metre: simply encode all tetramoraic feet as 4 regardless of whether or not they exclude certain patterns
  - for partially lacunose feet, show only the number of lost morae
    - e.g. to encode the prosody of a partially lost tetramoraic foot of which one light syllable is extant at the end, use "3-"

Table 2. Prosodic notation

Description	Code	Conventional notation	Context
one short/light syllable	-	⏑	1, 2, 3
one long/heavy syllable	+	–	1, 2, 3
one syllable of indeterminate length (anceps)	=	⏓	1, 2, (3)

two morae (one long or two short syllables)	2	⏑⏑	1, 2, (3)
larger moraic foot or colon	numeral(s)		1, 2, (3)
foot boundary	|	|	1, 2, (3)
caesura	||[56]	||	1
boundary of odd and even quarter	/		1[57]

Sanskrit syllabic metres

– the names listed below are to be used as values of `@met` in `<lg>`
  – always use metre names exactly in the form shown there (rather than legitimate variant or alternative names)
– the XML notation shown below uses the prosodic code introduced on page 133 above
  – caesurae are indicated in conventional notation for the sake of accuracy and to help you in metre identification, but are not shown in the XML notation, so if you wish, you can copy and paste segments of this notation for use in the `@met` attribute of lost text (encoded as per §5.4.4)

Table 3. Sanskrit syllabic metres

Syllables	Name	XML notation	Conventional notation
7/7	sumānikā[58]	+–+–+–=	–⏑–⏑–⏑⏓
8/8	anuṣṭubh[59]	====–++=/ ====–+–=	⏓⏓⏓⏓⏑––⏓/ ⏓⏓⏓⏓⏑–⏑⏓
10/11	vegavatī	––+––+––+=/ +––+––+––+=	⏑⏑–⏑⏑–⏑⏑–⏓/ –⏑⏑–⏑⏑–⏑⏑–⏓
10/11	viyoginī[60]	––+––+–+–=/ ––++––+–+–=	⏑⏑–⏑⏑–⏑–⏑⏓/ ⏑⏑––⏑⏑–⏑–⏑⏓
11	triṣṭubh[61]		
11	dodhaka	+––+––+––+=	–⏑⏑–⏑⏑–⏑⏑–⏓
11	indravajrā	++–++––+–+=	––⏑––⏑⏑–⏑–⏓
11	rathoddhatā	+–+–––+–+–=	–⏑–⏑⏑⏑–⏑–⏑⏓
11	śālinī	+++++–++–+=	––––|–⏑––⏑–⏓
11	svāgatā	+–+–––+––+=	–⏑–⏑⏑⏑–⏑⏑–⏓
11	upajāti[62]	=+–++––+–+=	⏓–⏑––⏑⏑–⏑–⏓

[56] Two iterations of | [U+007C Vertical Line], not a || double vertical bar character.
[57] Use only in `@met` for stanzas where a conventional metre name is not available and the metre has a different template for odd and even lines (*ardhasamavṛtta*).
[58] In assigning a name to this very rare metre, we follow Damais (1952: 25) who in turn relies on an editorial (correction) to a list of metres in Colebrooke (1873: 141, n.1). It appears from Velankar (1949: १२१) that no two traditional authorities agree on a name for this metre. The names cited there from treaties on poetics are: *uṣṇih* (which is also the class name for 7-syllable *samavṛtta*s), *kāminī*, *kheṭaka*, *gominī*, *raktā*, *śikhā* and *samānikā*.
[59] Also known as *śloka*, *vaktra*. See also the *Notes on* anuṣṭubh on page 136 below.
[60] If a verse matches this template, do not classify it as *vaitālīya*; see the *Notes on the* vaitālīya *family* on page 137 below.
[61] Used as an umbrella term for 11-syllable metres not conforming to one of the specific schemes listed here; see *Vedic trimeter* on page 138 below.
[62] Also known as *triṣṭubh upajāti*; see the *Notes on the* upajāti *family* on page 137 below.

Syllables	Name	XML notation	Conventional notation
11	upendravajrā	−+−++−−+−+=	⏑−⏑−−⏑⏑−⏑−⏓
11	toṭaka	−−+−−+−−+−−=	⏑⏑−⏑⏑−⏑⏑−⏑⏑⏓
11	vātormī	++++−−++−+=	−−−−⏑⏑−−⏑−⏓
11/11	upacitra[63]	−−+−−+−−+−=/ +−−+−−+−−+=	⏑⏑−⏑⏑−⏑⏑−⏑⏓/ −⏑⏑−⏑⏑−⏑⏑−⏓
11/12	aparavaktra	−−−−−−+−+−= −−−−+−−+−+−=	⏑⏑⏑⏑⏑⏑−⏑−⏑⏓ ⏑⏑⏑⏑−⏑⏑−⏑−⏑⏓
11/12	hariṇaplutā[64]	−−+−−+−−+−=/ −−−+−−+−−+−=	⏑⏑−⏑⏑−⏑⏑−⏑⏓/ ⏑⏑⏑−⏑⏑−⏑⏑−⏑⏓
11/12	mālabhāriṇī[65]	−−+−−+−+−+=/ −−++−−+−+−+=	⏑⏑−⏑⏑−⏑−⏑−⏓/ ⏑⏑−−⏑⏑−⏑−⏑−⏓
12	jagatī[66]		
12	bhujaṅgaprayāta	−++−++−++−+=	⏑−−⏑−−⏑−−⏑−⏓
12	maṇimālā	++−−++++−−+=	−−⏑⏑−−|−−⏑⏑−⏓
12	drutavilambita	−−−+−−+−−+−=	⏑⏑⏑−⏑⏑−⏑⏑−⏑⏓
12	pramitākṣarā	−−+−+−−−+−−=	⏑⏑−⏑−⏑⏑⏑−⏑⏑⏓
12	vaṁśamālā[67]	=+−++−−+−+−=	⏓−⏑−−⏑⏑−⏑−⏑⏓
12	vaṁśastha[68]	−+−++−−+−+−=	⏑−⏑−−⏑⏑−⏑−⏑⏓
12	vaiśvadevī	++++++−++−+=	−−−−−|−⏑−−⏑−⏓
12/13	puṣpitāgrā	−−−−−−+−+−+=/ −−−−+−−+−+−+=	⏑⏑⏑⏑⏑⏑−⏑−⏑−⏓/ ⏑⏑⏑⏑−⏑⏑−⏑−⏑−⏓
13	mañjubhāṣiṇī	−−+−+−−−+−+−=	⏑⏑−⏑−⏑⏑⏑−⏑−⏑⏓
13	mattamayūra	+++++−−++−−+=	−−−−|−⏑⏑−−⏑⏑−⏓
13	praharṣiṇī	+++−−−−+−+−+=	−−−|⏑⏑⏑⏑−⏑−⏑−⏓
13	rucirā	−+−+−−−−+−+−=	⏑−⏑−|⏑⏑⏑⏑−⏑−⏑⏓
14	asaṁbādhā	+++++−−−−−−++=	−−−−−|⏑⏑⏑⏑⏑⏑−−⏓
14	praharaṇakalikā	−−−−−−+−−−−−−=	⏑⏑⏑⏑⏑⏑−|⏑⏑⏑⏑⏑⏑⏓
14	vasantatilakā[69]	++−+−−−+−−+−+=	−−⏑−⏑⏑⏑−⏑⏑−⏑−⏓

[63] All lines contain 11 syllables, but the rhythm of the odd lines is different from the rhythm of the even lines.
[64] The rhythm of the first line of the *hariṇaplutā* is the same as that of the *upacitra.*
[65] If a verse matches this template, do not classify it as *aupacchandasika.*
[66] Used as an umbrella term for 12-syllable metres not conforming to one of the specific schemes listed here; see *Vedic trimeter* on page 138 below.
[67] Also known as *jagatī upajāti*; see the *Notes on the* upajāti *family* on page 137 below.
[68] Also known as *vaṁśasthavila.*
[69] Also known as *vasantatilaka*, *uddharṣiṇī*, *siṁhonnatā.* Though not explicitly prescribed in any extant metrical treatise, poets

Syllables	Name	XML notation	Conventional notation
15	mālinī	------+++-+++-+=	⏑⏑⏑⏑⏑⏑––|–⏑––⏑–⏒
17	hariṇī	-----+++++-+--+-=	⏑⏑⏑⏑⏑–|––––|⏑–⏑⏑–⏑⏒
17	mandākrāntā	++++-----++-++-+=	––––|⏑⏑⏑⏑⏑–|–⏑––⏑–⏒
17	nardaṭaka	----+-+---+--+--=	⏑⏑⏑⏑–⏑–⏑⏑⏑–⏑⏑–⏑⏑⏒
17	pṛthvī	-+---+-+---+-++-=	⏑–⏑⏑⏑–⏑–|⏑⏑⏑–⏑––⏑⏒[70]
17	śikharinī	-+++++-----++---=	⏑––––––|⏑⏑⏑⏑⏑––⏑⏑⏑⏒
19	śārdūlavikrīḍita	+++--+-+---+++-++-=	–––⏑⏑–⏑–⏑⏑⏑–|––⏑––⏑⏒
19	sumadhurā	++++-++-----+++---=	––––⏑––|⏑⏑⏑⏑⏑–|––⏑⏑⏑⏒
20	mattebhavikrīḍita	--+++--+-+---+++-++-=	⏑⏑––⏑⏑–⏑–⏑⏑⏑–|––⏑––⏑⏒
20	suvadanā	++++-++------+++---=	––––⏑––|⏑⏑⏑⏑⏑–|––⏑⏑⏑⏒
21	sragdharā	++++-++------++-++-+=	––––⏑––|⏑⏑⏑⏑⏑–|–⏑––⏑–⏒

Notes on *anuṣṭubh*

– we use *anuṣṭubh* as an umbrella term and shall not make a distinction between Vedic *anuṣṭubh*, and classical *vaktra*/*śloka* in our classification
– when encoding the prosody of lost text (§5.4.4), the first four syllables of each *anuṣṭubh* quarter shall be marked up as indeterminate, as shown in the list of syllabic metres above
– there are in fact some additional restrictions for these syllables, set out in Table 4 below
– if you wish, feel free to mark up any non-conformant lines as metrically deviant (§2.3.5)
– note that *vipulā* variations (alternative cadences in the odd quarters) do not require any explicit markup but may be encoded as metrically deviant lines (§2.3.5)

Table 4. Permitted patterns in *pathyā anuṣṭubh*

	1	2–4	5	6	7	8
odd	⏒	––⏒ ⏑–⏒ –⏑⏒	⏑	–	–	⏒
even	⏒	––⏒ ⏑–⏒ –⏑⏑	⏑	–	⏑	⏒

often observe a caesura after the 8th syllable of a *vasantatilakā* line. (For further discussion see Pollock 1977, 73-74.)

[70] The caesura in *pṛthvī* is not observed by all poets.

Table 5. Recognised *vipulā anuṣṭubh* patterns (even lines only)

	1	2–4	5	6	7	8
na-vipulā	⏑̲	– – – ⏑ – – – ⏑ –	⏑	⏑	⏑	⏑̲
bha-vipulā	⏑̲	– ⏑ –	–	⏑	⏑	⏑̲
ma-vipulā	⏑̲	– ⏑ –	– |	–	–	⏑̲
ra-vipulā	⏑̲	– – – ⏑ – – – ⏑ –	| –	⏑	–	⏑̲

Notes on the *upajāti* family

– this family of metres includes 11 and 12-syllable metres which vary in the length of the first syllable and thus give rise to ambiguities concerning classification
  – *upajāti* or *triṣṭubh upajāti*, a free mix of *indravajrā* and *upendravajrā*
  – *vaṁśamālā* or *jagatī upajāti*, a free mix of *indravaṁśa* and *vaṁśastha*
– when every line of a stanza is in one of the "pure" metres (e.g. *indravajrā*), that stanza should normally be classified as that pure metre, whereas stanzas with one or more lines in the other child metre should be classified as the "mixed metre" (e.g. *upajāti*)
– however, the mixed metres are more widely used than the pure ones, therefore
  – if an inscription includes several successive stanzas of a mixed metre among which one or a few stanzas are in a pure metre, then it makes better sense to classify the pure stanza(s) as being also of the mixed metre (assuming that the poet was composing in the mixed metre and by chance all lines of that particular stanza turned out in one of the pure metres)
  – if an inscription includes a stanza in one of these metres with at least one line-initial syllable lost, then it is better to assume the stanza to be in the mixed metre even if all the fully extant lines are in one of the pure metres
  – there may always be cases where the above considerations do not apply; for example when a composer shows off his skill by employing a wide variety of metres

Notes on the *vaitālīya* family

– this family of *ardhasama* metres also gives rise to ambiguities of classification because it uses a loose moraic template for the first part of each line and a syllabic template for the cadence (final part) of each line:
  – *vaitālīya*, with the pattern ⏑̲⏑̲ – ⏑ – ⏑ ⏑̲ / ⏑̲⏑̲⏑̲⏑̲ – ⏑ – ⏑ ⏑̲
  – *aupacchandasika*, with the pattern ⏑̲⏑̲⏑̲ – ⏑ – ⏑ – ⏑̲ / ⏑̲⏑̲⏑̲⏑̲ – ⏑ – ⏑ – ⏑̲
  – and the much rarer *āpātalikā*, with the pattern ⏑̲⏑̲⏑̲ – ⏑ ⏑ – ⏑̲ / ⏑̲⏑̲⏑̲⏑̲ – ⏑ ⏑ – ⏑̲
– in addition, there exist a small number of fully syllabic templates which are specific, constrained instantiations of the above, partly moraic templates:
  – *vaitālīya* may be realised as *viyoginī* or *aparavaktra* (see Table 3 for the patterns)
  – *aupacchandasika* may be realised as *mālabhāriṇī* or *puṣpitāgrā*
  – *āpātalikā* may be realised as *vegavatī*
– in actual poetic practice, these fully syllabic instantiations are much more common than the less constrained moraic templates
– nonetheless, many editors of Indic texts prefer to classify such stanzas by the generic metre and not by the specific instantiation
– you should avoid this and, if a previous edition identifies a stanza as one of these generic metres or if you are editing a previously unedited text, check whether the stanza in fact conforms to one of the specific metres, and if it does, mark it up as such

Vedic trimeter

– though rare in our epigraphic corpus, some stanzas may be composed in lines of 11 or 12 syllables that do not observe any of the strict schemes named in Table 3 above; instead,
  – lines consist of 11±1 or 12±1 syllables, with varying line number permitted within a stanza
  – the initial colon (the "opening", before a more or less clear caesura) is relatively free, but predominantly trochaic
  – the caesura is generally followed by a pair of short syllables ("break")
  – the final colon (cadence) of each line is relatively fixed in a trochaic pattern
– such metres shall be collectively referred to as trimeter, following Arnold (1905:7, 11-14)
– we judge that a rough typology of metrical patterns serves our needs better than a detailed encoding that could give due consideration to the intricacies of these metres
– therefore, use the following values of `@met` for stanzas in such metres
  – `"triṣṭubh"` for stanzas of predominantly 11-syllable lines which predominantly conform to either of the following patterns
    – ⏑ – ⏑ – || ⏑ ⏑ – | – ⏑ – ⏑
    – ⏑ – ⏑ – ⏑ || ⏑ ⏑ | – ⏑ – ⏑
  – `"jagatī"` for stanzas of predominantly 12-syllable lines which predominantly conform to either of the following patterns
    – ⏑ – ⏑ – || ⏑ ⏑ – | – ⏑ – ⏑ ⏑
    – ⏑ – ⏑ – ⏑ || ⏑ ⏑ | – ⏑ – ⏑ ⏑
  – `"trimeter"` as a general token for stanzas where the metrical pattern and/or length of the lines varies more than in the more specific metres named above
– depending on the level of your interest in metrical studies, feel free to encode the actual prosody of each line in `@real`

Sanskrit/Prakrit moraic metres

– the metres of the *āryā* or *gāthā* family consist of two hemistichs, each comprised of eight feet which are tetramoraic except for the 6th (which is a single mora in some cases) and the 8th (which is usually bimoraic), with a caesura after the 3rd foot

each hemistich follows an exact template as listed in
  – Table 7 below
  – keep in mind that by our encoding convention, a hemistich in such a metre is encoded as one `<l>` element (definition of "line" in §2.3.1), with the number `"ab"` or `"cd"` (§2.3.2)
– the metre names listed below are to be used as values of `@met` in `<lg>`
– if you are encoding verse of this type with lacunae, it is not necessary to encode the prosody of lacunae more accurately than the generic template shown in the list below (but feel free to do so where you can)

the full detail of permitted metrical variation in the regular (*pathyā*) form is shown in
– Table 7 below, including the following variations:
  – a hemistich in which the caesura after the third foot is ignored or displaced is called a *vipulā*, which may be marked up as an unobserved caesura (§2.3.2)
  – a hemistich with a special constraint applied to the first 5 feet is called a *capalā*
  – metrically anomalous and *capalā* lines may optionally be marked up using the attribute `@real` (§2.3.5)

Table 6. Names and general pattern of moraic metres

	Moraic feet	Template (see next table)
āryā	4|4|4||4|4|4|4|2/ 4|4|4||4|4|1|4|2	A B
gīti	4|4|4||4|4|4|4|2/ 4|4|4||4|4|4|4|2	A A
upagīti	4|4|4||4|4|1|4|2/ 4|4|4||4|4|1|4|2	B B
udgīti	4|4|4||4|4|1|4|2/ 4|4|4||4|4|1|4|2	B A
āryāgīti	4|4|4||4|4|4|4|4/ 4|4|4||4|4|4|4|4	C C

Table 7. Specifics of moraic metres

	1	2	3	4	5	6	7	8
A	⏑⏑⏑⏑	⏑–⏑ ⏑⏑⏑⏑	⏑⏑⏑⏑	|⏑–⏑ |⏑⏑⏑⏑	⏑⏑⏑⏑	⏑–⏑ ⏑|⏑⏑⏑	|⏑⏑⏑⏑ ⏑⏑– –⏑⏑ ––	⏑⏑
B	⏑⏑⏑⏑	⏑–⏑ ⏑⏑⏑⏑	⏑⏑⏑⏑	|⏑–⏑ |⏑⏑⏑⏑	|⏑⏑⏑⏑ ⏑⏑– –⏑⏑ ––	⏑	⏑⏑⏑⏑	⏑⏑
C	⏑⏑⏑⏑	⏑–⏑ ⏑⏑⏑⏑	⏑⏑⏑⏑	|⏑–⏑ |⏑⏑⏑⏑	⏑⏑⏑⏑	⏑–⏑ ⏑|⏑⏑⏑	|⏑⏑⏑⏑ ⏑⏑– –⏑⏑ ––	⏑⏑⏑⏑
capalā	⏑⏑–	⏑–⏑	––	|⏑–⏑	–⏑⏑			
vipulā				⏑–⏑ ⏑|⏑⏑⏑				

Tamil metres

– due to practical considerations, Tamil metre shall be classified only by major types (*pā*), as shown in Table 8 below

– the names in the first column are to be used as values of `@met` in `<lg>`

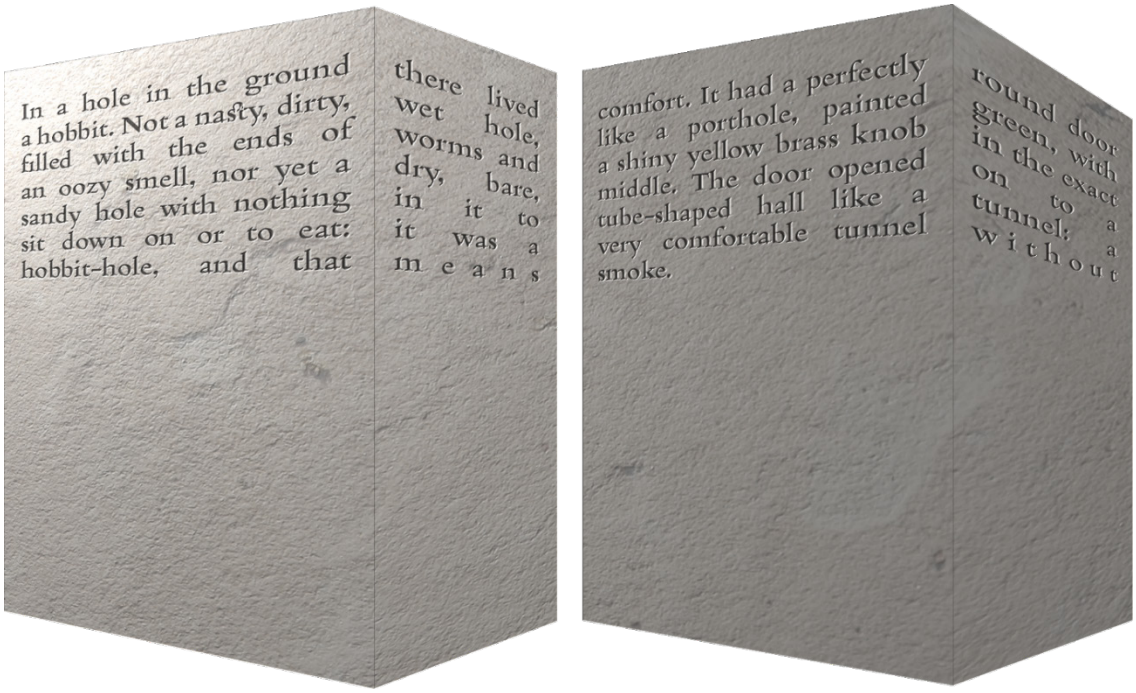
Table 8. Tamil metres

Major type	Included forms	Included subtypes (*pāviṉam*)
veṇpā	kuṟaḷ-veṇpā (2 *aṭi*s) nēricai-veṇpā (4 *aṭi*s) iṉṉicai-veṇpā (4 *aṭi*s) cintiyal-veṇpā (3 *aṭi*s) paḵṟoṭai-veṇpā (5 to 12 *aṭi*s)	kuṟaḷ-veṇcenturai kuṟaḷ-tāḻicai veṇ-tāḻicai veṇ-turai veḷi-viruttam
āciriyappā (3 to 1000 *aṭi*s)	nēricai-āciriyappā iṇaikkuṟal-āciriyappā	āciriya-tāḻicai āciriya-turai

	nilaimaṇṭila-āciriyappā aṭimaṟimaṇṭila-āciriyappā	āciriya-viruttam
kalippā[71]	ottāḻicai-kalippā (with 3 subforms) veṇ-kalippā koccaka (with 5 subforms, the 5th one having 3 subsubforms)	kali-tāḻicai kali-tuṟai kali-viruttam kaṭṭalai-kalittuṟai kaṭṭalai-kalippā
vañcippā[72]	(no forms)	vañci-tāḻicai vañci-tuṟai vañci-viruttam
maruṭpā	composed of elements of veṇpā and āciriyappā	

Appendix C. "Case Studies" in Encoding Complex Layout

Case study 1: four-faced stele



– this imaginary stele is an oblong quadrangle in cross-section
– all four sides are inscribed, with text starting on one of the broad faces
  – each line of the text runs across one edge, onto the adjacent narrow face
  – subsequent lines fill up the inscribed field of this pair of faces
  – the text then flows on to the top of the next broad face and proceeds to fill up that face and the adjacent narrow face in the same way as the first pair of faces
– thus, the partition between the two pairs is pagelike (§3.5), therefore the virtual zones must be marked up using pagelike milestone elements (§3.5.3)
– the boundary between a wide face and the adjacent narrow face comprises a gridlike partition, since the text disregards the physical transition to a new surface
  – such a pair of faces thus constitutes a single virtual zone, whose partition into two physical surfaces may be ignored in the markup or may be encoded by means of a gridlike milestone element (§3.6)

[71] Contains up to 6 different types of elements which are: *taravu*, *tāḻicai*, *arākam*, *ampōtarańkam*, *taṉiccol*, *curitakam*.
[72] Contains elements which are: *taṉiccol*, *akaval-curitakam*.

```
<p><!--All milestones come within paragraphs.-->
<milestone type="pagelike" unit="faces" n="Ab"/><!--An optional <label> could have been added at
this point and for the second facepair, as per §3.5.4. -->
<lb n="Ab1"/><milestone unit="face" n="A"/>In a hole in the ground <milestone unit="face"
n="b"/>there lived
<lb n="Ab2"/><milestone unit="face" n="A"/>a hobbit. Not a nasty, dirty, <milestone unit="face"
n="b"/>wet hole,
<lb n="Ab3"/><milestone unit="face" n="A"/>filled with the ends of <milestone unit="face"
n="b"/>worms and
<lb n="Ab4"/><milestone unit="face" n="A"/>an oozy smell, nor yet a <milestone unit="face"
n="b"/>dry, bare,
<lb n="Ab5"/><milestone unit="face" n="A"/>sandy hole with nothing <milestone unit="face"
n="b"/>in it to
<lb n="Ab6"/><milestone unit="face" n="A"/>sit down on or to eat: <milestone unit="face"
n="b"/>it was a
<lb n="Ab7"/><milestone unit="face" n="A"/>hobbit-hole, and that <milestone unit="face"
n="b"/>means
<milestone type="pagelike" unit="faces" n="Cd"/>
<lb n="Cd1"/><!-- This encoding example uses the repetitive scheme of line numbering §3.2.2, in
accordance with SE Asian epigraphic conventions. Alternatively, lines could have been numbered
starting from 1 and continuing from 8 on face Cd.-->
<milestone unit="face" n="C"/>comfort.
</p><!-- The text has been broken up into two semantic paragraphs §2.2.1 for the sake of this
illustration.-->
<p>It had a perfectly <milestone unit="face" n="d"/>round door
<lb n="Cd2"/><milestone unit="face" n="C"/>like a porthole, painted <milestone unit="face"
n="d"/>green, with
<lb n="Cd3"/><milestone unit="face" n="C"/>a shiny yellow brass knob <milestone unit="face"
n="d"/>in the exact
<lb n="Cd4"/><milestone unit="face" n="C"/>middle. The door opened <milestone unit="face"
n="d"/>on to a
<lb n="Cd5"/><milestone unit="face" n="C"/>tube-shaped hall like a <milestone unit="face"
n="d"/>tunnel: a
<lb n="Cd6"/><milestone unit="face" n="C"/>very comfortable tunnel <milestone unit="face"
n="d"/>without
<lb n="Cd7"/><milestone unit="face" n="C"/>smoke.
</p>
```

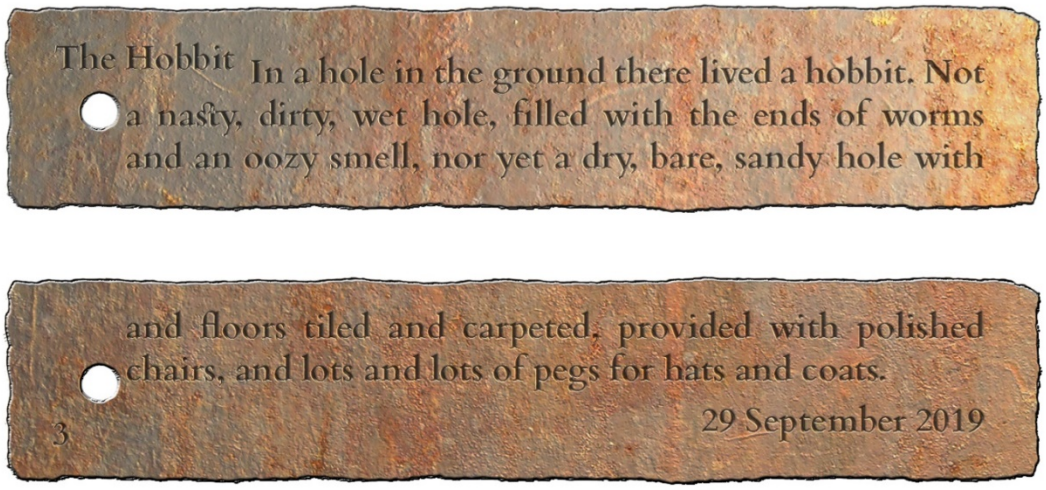
Case study 2A: copperplate charter with seal and other goodies

– this imaginary set of copper plates consists of
  – an inscribed seal
  – three plates, the first and the last inscribed only on their inner faces
    – with foliation marks on the recto faces of the second and third plate
    – with an inset initial text on the first page
    – with a visually separated colophon on the last page
– the seal and the plates comprise a boxlike partition that must be encoded as two textpart divisions (§3.4)
– the second division is a virtual zone subdivided into pagelike partitions (§3.5), which are actual pages and must thus be encoded as `<pb>` elements
  – the blank outer pages must also be encoded (§3.5.2)
– within the second division,
  – text begins with a floating incipit, which must be marked up as a special line (§3.3.3)
  – the foliation numbers comprise forme work (§3.3.5), each attached to the relevant page
  – the special alignment of the colophon may optionally be encoded (§7.5.2)

```
<div type="textpart" n="A"><head xml:lang="eng">Seal</head><!--Seal as a division. Since the two
divisions are different in nature, @subtype is not used, but a <head> is added for identification
§3.4.3.--><!--By project convention, the seal is encoded before the text of the plates.-->
  <ab><!--Seal text wrapped in a block-level element, in this case <ab> because it does not
```

```
qualify as a paragraph §2.2.2.-->
  <lb n="1"/>J R R
  <lb n="2"/>Tolkien
  </ab>
</div>
<div type="textpart" n="B"><head xml:lang="eng">Plates</head>
  <pb n="1r"/><!--Blank outer page; the page beginning is not inside a block-level element
§3.5.2.-->
  <ab>
    <pb n="1v"/><!--This page beginning is inside the first block-level container of the text,
which happens to be the <ab> wrapper for the incipit. This does not imply that the page break is
part of that <ab>.-->
    <lb n="01"/>The Hobbit<!--Line numbers are mandatorily restarted in the second textpart
§3.4.4. The specially positioned incipit has the line number 01 §3.3.3.-->
  </ab>
  <p><!--First semantic paragraph of the text.-->
  <lb n="1"/>In a hole in the ground there lived a hobbit. Not
  <lb n="2"/>a nasty, dirty, wet hole, filled with the ends of worm
  <lb n="3"/>and an oozy smell, nor yet a dry, bare, sandy hole with
  <pb n="2r"/>
  <!--Within a textpart, line numbers are continued on subsequent pages as recommended under
§3.2.2. Alternatively, they could be reset to 1 on each page, provided that the number of the
current page is incorporated into each line number to maintain uniqueness.-->
  <fw place="bot-left" n="2r">
    <!--Foliation encoded right after the page beginning, §3.3.5.-->
    <num value="2">2</num>
  </fw>
  <lb n="4"/>nothing in it to sit down on or to eat: it was a
  <lb n="5"/>hobbit-hole, and that means comfort.
  </p><!--Ending a semantic paragraph here and starting a new one.-->
  <p>It had a perfectly
  <lb n="6"/>round door like a porthole, painted green, with a shiny
  <pb n="2v"/>
  <lb n="7"/>yellow brass knob in the exact middle. The door opened
  <lb n="8"/>on to a tube-shaped hall like a tunnel: a very
  <lb n="9"/>comfortable tunnel without smoke, with panelled walls,
  <pb n="3r"/>
  <fw place="bot-left" n="3r">
    <!--Foliation right after the page beginning.-->
    <num value="3">3</num>
  </fw>
  <lb n="10"/>and floors tiled and carpeted, provided with polished
  <lb n="11"/>chairs, and lots and lots of pegs for hats and coats.
  </p>
  <ab><!--The colophon is an incomplete sentence, so it gets an <ab> wrapper.-->
  <lb n="12" style="text-align: right"/><!--Optionally marking up right-aligned line.-->
  <num value="29">29</num> September <num value="2019">2019</num>
  </ab>
  <pb n="3v"/><!--Blank outer page; the <pb/> element is not inside a block-level container.-->
</div>
```

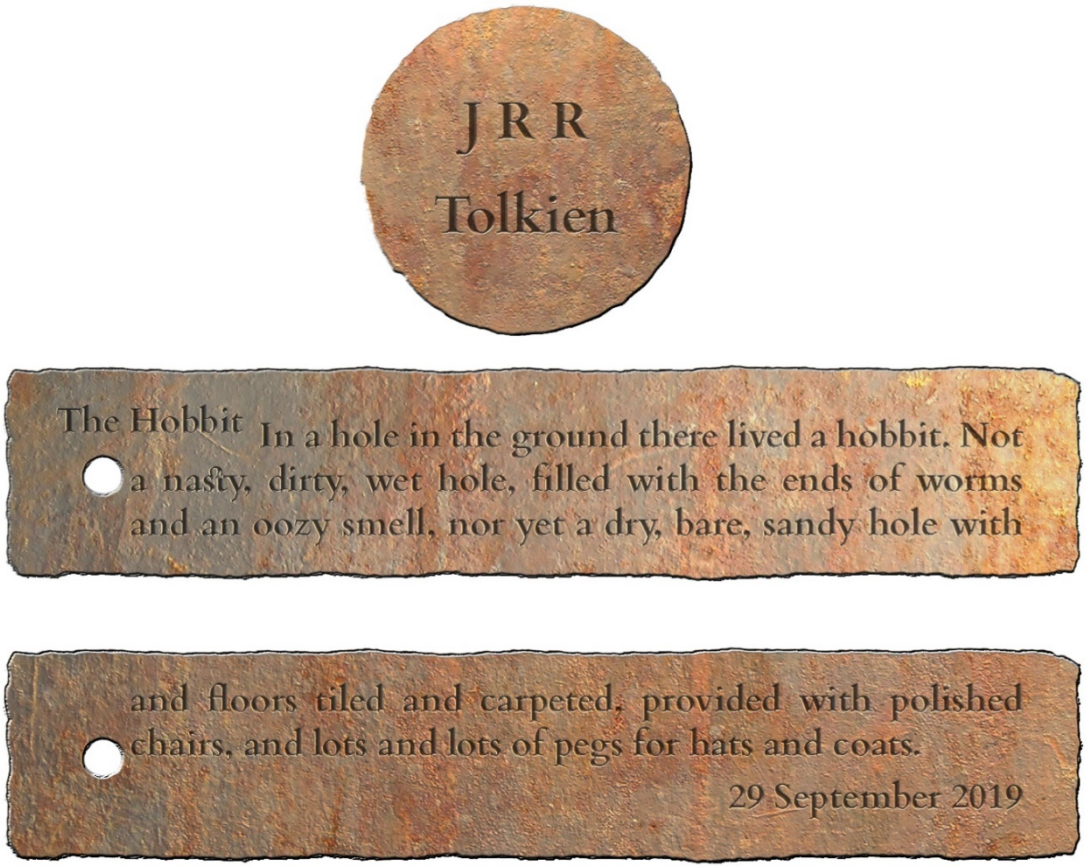
Case study 2B: copperplate charter with a lost plate reconstructed



– as a variation on Case Study 2A, we now have a partial set of plates where the middle plate is missing along with the seal
– from the presence of foliation marks (and from our intimate knowledge of the Middle Earth copper plate corpus) we can infer that exactly one plate was lost, therefore we include the lost plate in our edition instead of creating textpart divisions (§5.4.8)
  – we can also infer that the lost plate would have been inscribed with exactly three lines on both faces, so this is also encoded in the edition
– extant details are encoded as in Case Study 2A above

```
<pb n="1r"/>
<ab>
  <pb n="1v"/>
  <lb n="1v0"/>The Hobbit
</ab>
<!--In this edition, line numbering is restarted on each page, and page numbers are incorporated
in line numbers as per §3.2.2. Since the number of lines per page is known, we could have opted
to number lines consecutively, logically continuing line numbers after the lacuna §5.4.8.-->
<p part="I"><!--The incomplete paragraph is marked as an initial part.-->
  <lb n="1v1"/>In a hole in the ground there lived a hobbit. Not
  <lb n="1v2"/>a nasty, dirty, wet hole, filled with the ends of worm
  <lb n="1v3"/>and an oozy smell, nor yet a dry, bare, sandy hole with
</p><!--The open block-level container is closed before the lacuna.-->
<pb n="2r"/><!--Although a foliation mark was in all probability present on the lost plate, we do
not restore one.-->
<gap reason="lost" quantity="3" unit="line"/>
<!--Individual line beginnings are not reconstructed on a lost page, only recorded as a lacuna of
known size. -->
<pb n="2v"/>
<gap reason="lost" quantity="3" unit="line"/>
<p part="F"><!--A new block-level container, marked as a final part, is opened after the lacuna,
before the next extant page beginning.-->
  <pb n="3r"/>
  <fw place="bot-left" n="3r">
    <num value="3">3</num>
  </fw>
  <lb n="3r1"/>and floors tiled and carpeted, provided with polished
  <lb n="3r2"/>chairs, and lots and lots of pegs for hats and coats.
</p>
<ab>
  <lb n="3r3" style="text-align: right"/>
  <num value="29">29</num> September <num value="2019">2019</num>
</ab>
<pb n="3v"/>
```

Case study 2C: copperplate charter with a lost plate not reconstructed



– as another variation on Case Study 2A, we again have a partial set of plates where the middle plate is missing
– this time, however, the seal is extant and there are no foliation marks, nor are we sufficiently familiar with any other Middle Earth plates, so we cannot confidently reconstruct the structure of the lost section
– our edition must therefore be divided into three textparts: one for the seal, one for the initial plate, and one for the final plate (§5.4.8)
– extant details are encoded as in Case Study 2A above

```
<div type="textpart" n="A"><head xml:lang="eng">Seal</head>
  <ab>
  <lb n="1"/>J R R
  <lb n="2"/>Tolkien
  </ab>
</div>
<div type="textpart" n="B"><head xml:lang="eng">Initial plate</head>
  <pb n="1r"/>
  <ab>
    <pb n="1v"/>
    <lb n="01"/>The Hobbit
  </ab>
  <p part="I"><!--Incomplete paragraph marked as an initial part. -->
  <lb n="1"/>In a hole in the ground there lived a hobbit. Not
  <lb n="2"/>a nasty, dirty, wet hole, filled with the ends of worm
  <lb n="3"/>and an oozy smell, nor yet a dry, bare, sandy hole with
  </p>
</div>
<!--Nothing is encoded for the lacuna itself, since the use of textparts makes it clear that text
is lost between the two; details go in the layout description and the commentary. -->
<div type="textpart" n="C"><head xml:lang="eng">Final plate</head>
  <p part="F"><!--Incomplete paragraph marked as a final part. -->
```

```
  <pb n="1r"/>
  <!--Page and line numbering are reset to 1 in the second textpart §3.4.4. -->
  <lb n="1"/>and floors tiled and carpeted, provided with polished
  <lb n="2"/>chairs, and lots and lots of pegs for hats and coats.
  </p>
  <ab>
  <lb n="3" style="text-align: right"/>
  <num value="29">29</num> September <num value="2019">2019</num>
  </ab>
  <pb n="1v"/>
</div>
```

Appendix D. Language Codes

– some of the languages that concern us do not yet have a language codes
  – for these we must use provisional codes (starting with x) until our code requests (submitted on December 15 2019) yield the desired result.
– note that for Dravidian languages, we do not make a distinction between “Old” and “Modern” varieties
  – by contrast, in the case of all vernacular languages of Southeast Asia, you must make this distinction, although we expect you will only very rarely have the need to use codes for modern Burmese, modern Cam, modern Javanese modern Khmer, or Malay(sian)/Indonesian.

Table 9. ISO 639-3 language codes

Language (and script)	code
Undetermined language	unknown
Burmese, modern	mya
Burmese, old	obr
Cham, modern (of Phanrang)	cjm
Cham, old (generally known as “Cham”)	x-oldcham
Dutch	ndl
English	eng
French	fra
Indonesian	ind
Japanese	jpn
Javanese, modern	jav
Javanese, old	kaw
Kannada (modern or old)	kan
Khmer, modern	khm
Khmer, old	x-oldkhmer
Malay, modern (Bahasa Malaysia)	zlm
Malay, old	x-oldmalay
Mon, old	omx
Pali	pli
Prakrit	pra
Pyu	pyx
Sanskrit	san
Sundanese, old	x-oldsun
Tamil (modern or old)	tam
Telugu (modern or old)	tel
Vietnamese	vie

Appendix E. Titling Conventions

– when assigning a title to the inscription you are encoding, it is in general recommended that you try to remain faithful to established names for inscriptions that have been edited before
  – in cases where established names in a corpus follow significantly varying models, so that the need for some harmonization is felt, the team member(s) responsible for the corpus in question has/have the option to design a new system and apply it rigorously to all members of the corpus in question
– new inscriptions should, as a rule, be named on the analogy of established names within the same corpus
– in all cases, variant names applied to the inscription in question in previous publications shall be recorded in the metadata spreadsheet (and will be made searchable once imported from there into our TEI headers)
– when creating new titles, it is recommended (but not mandatory) that you compose your title by combining the following elements, in this order

1. **place**: start your title with a place name, using (if possible) one of the following options
   a. internal: attempt to identify the name of the place(s) that is (or are) most fundamentally concerned by the transaction recorded in the inscription
      – resort to a less important but more distinctive internal toponym if the toponym resulting from the above test is insufficiently distinctive
      – if you are dealing with Indonesian inscriptions, represent the internal toponym free of diacritics, in EYD spelling (e.g., Sobhamerta for *śobhāmṛta*), and for further guidance, see Damais 1952: 6-9
   b. based on provenance: use the name of the findspot, e.g. "Nalanda"
      – provenance-based places may, if necessary, include the specification of a topographic feature or a monument, e.g. "rock", "cliff", "victory pillar", "Cave 16", "Vedānteśvara temple pillars", etc.
   c. based on custody: a reference to the place or institution where the support is currently kept. E.g. BBRAS plates of Dhruvasena I, year 210
      – place names based on custody should only be employed if neither a provenance-based, nor an internal place name is available
2. **artifact type or document type**: add a word or two describing the nature of the support or the text, using the cover term "inscription" only if no more satisfactory term can be found.
   – use document type (e.g., "grant", "dedication" or "dedicatory inscription", "foundation", "label", "graffito", "eulogy") if the place name used in the title is an internal toponym
      – e.g. "Raktamālā grant"
   – use artifact type (e.g., "plate", "stele", "slab", "pillar", etc.) if the place name used in the title is based on provenance or custody
      – e.g. "Nalanda plates", "British Museum pillar"
3. **the principal protagonist** of the inscription, if named in the text
   – this is usually the person who issued/commissioned the inscription and/or the work commemorated in it, but may be another named person, such as the reigning ruler if the commissioner is another person who is not named
   – preferably, use "of" before the name of the issuer
   – preferably, use "of the time of" before the name of a reigning ruler who is not the protagonist
4. **supplementary details**: after the above elements, add further information whenever necessary for disambiguation, potentially including:
   – calendrical or regnal year, if mentioned within the text
      – optionally including further dating specifics noted in the text, if necessary for disambiguation, e.g. month and day in the case of several charters of the same provenance, same king, same year, and where a title based on the content does not work
      – optionally using "undetermined year" if a year is mentioned in the text, but is lost or illegible
   – any additional distinction, e.g.
      – "Set 1" and "Set 2" to distinguish two sets of copper plates with the same provenance, issued by the same ruler on the same date

  - "north column" and "south column" to distinguish two copies of an inscription engraved in duplicate
- the above items may be listed with commas or connected to one another by short English phrases as seems most suitable
- some examples of titles composed or elaborated along the above lines:
  - Nalanda plate of Samudragupta (provenance, type and issuing ruler)
  - Raktamālā grant of the time of Budhagupta, year 159 (internal place, type, reigning ruler, date)
  - Vallam, Vedānteśvara temple pillars, foundation by Kantacēn̲an̲ of the time of Mahendravarman I (provenance with monument details, type of inscription, issuing person, reigning ruler)
  - Uttiramērūr, Sundaravaradaperumāḷ temple, southern wall of *vimāna*, donation of the time of Nandivarman II, year 16 (provenance with monument details, additional specification of location, type, reigning ruler, regnal year)
  - Uttiramērūr, Vaikuṇṭhaperumāḷ temple, larger platform, southern base, inscription of the time of Dantivarman, year 8, larger platform, southern base (provenance with monument details, additional specification of location, no clear text type, reigning ruler, regnal year)
  - Gunung Wukir stele of Sañjaya
  - Hampran dedication of Bhānu

Appendix F. Normalisation Suggestions

- this appendix contains some specific suggestions for encoding non-standard usage in various languages
- all of these are to be understood as no more than suggestions: when encoding any particular text, feel free to apply a stricter or more lenient approach to any phenomenon depending on what seems to be normal in that given text and in related texts
- if the prosody of metrical verse is affected (either spoiled or corrected) by normalisation, then the considerations outlined in §6.1.4 overrule the suggestions listed below

Table 10. Normalisation suggestions

language	phenomenon	action
Sanskrit in Cambodia	mismatch of dentals with retroflexes in conjuncts (e.g., *ṇd*, *ṣth*)	flag or normalise
Old Javanese	spelling of long *pepet* with *ə* plus length mark[73]	ignore
Old Javanese	use of *r̥* or *l̥* (and *r̥̄* or *l̥̄*) in words whose dictionary spelling has *rə* or *lə* (or *rə* or *lə*)	ignore
Old Khmer	absence of *virāma*	ignore
Old Khmer	representation of final consonant *C* by the spelling *CCa*, including final /h/ represented as *ṇha*	ignore
Sundanese and Javanese	expected nasal+stop cluster spelt with only the stop	normalise
Sanskrit	sibilant doubled after *r*	ignore or flag
Sanskrit	any consonant doubled before *r*	ignore or flag
Sanskrit	use of *tv* where *ttv* is expected	ignore or flag
any	use of *b* where *v* is expected, or vice versa	flag
any	infidelity to the correct length of vowels in words borrowed from	ignore or flag

[73] To be represented as *ə:* as per TG §3.3.6.

language	phenomenon	action
	Sanskrit[74]	
any	use of *anusvāra* in place of a final *m·* or *M* or vice versa	ignore or flag
any	use of a nasal consonant in place of an *anusvāra* before a sibilant or *h*	ignore or flag
any	use of one nasal instead of another	ignore or flag
any	use of an aspirated consonant instead of its unaspirated counterpart or vice versa	ignore or flag
any	use of a dental consonant instead of its retroflex counterpart or vice versa	ignore or flag
any	use of one sibilant instead of another	ignore or flag

[74] See also TG §3.3.7 and EGD §6.3.7.

References

Apte, Vaman Shivaram. 1957. *Revised and enlarged edition of Prin. V. S. Apte's The practical Sanskrit-English dictionary*. Edited by P. K. Gode and C. G. Karve. Poona: Prasad Prakashan.

Arnold, Edward Vernon. 1905. *Vedic Metre in Its Historical Development*. Cambridge: Cambridge University Press.

Birnbaum, David J. 2015. *What is XML and why should humanists care? An even gentler introduction to XML*. http://dh.obdurodon.org/what-is-xml.xhtml

Bodard, Gabriel. 2010. 'EpiDoc: Epigraphic Documents in XML for Publication and Interchange'. In *Latin on Stone: Epigraphic Research and Electronic Archives*, edited by Francisca Feraudi-Gruénais, 101–18. Lanham: Lexington Books. http://www.stoa.org/wordpress/wp-content/uploads/2010/09/Chapter05_EpiDoc_Bodard.pdf

Colebrooke, Henry Thomas. 1873. *Miscellaneous Essays*. Vol. 3. London: Trübner.

Damais, Louis-Charles. 1952. 'Études d'épigraphie indonésienne, III: liste des principales inscriptions datées de l'Indonésie'. Bulletin de l'École française d'Extrême-Orient 46 (1): 1–105. https://doi.org/10.3406/befeo.1952.5158

Fleet, John Faithfull. 1888. *Inscriptions of the Early Gupta Kings and Their Successors*. Corpus Inscriptionum Indicarum, III. Calcutta: Superintendent of Government Printing.

Pollock, Sheldon Ivan. 1977. *Aspects of Versification in Sanskrit Lyric Poetry*. American Oriental Society.

Roueché, Charlotte and Julia Flanders. *The Gentle Introduction to Mark-up for Epigraphers*, http://www.stoa.org/epidoc/gl/latest/intro-eps.html

Velankar, Hari Damodar. 1949. *Jayadāman A Collection of Ancient Texts on Sanskrit Prosody and a Classified List of Sanskrit Metres with an Alphabetical Index*. Haritoṣamālā 1. Bombay: Haritosha Samiti.