

Manual for the quetta (ᑭᑭᑦ) module

Florent Michel

version 0.2.0

Contents

1. Introduction	1
1.1. ‘Quetta’?	1
1.2. The Tengwar script	1
2. How to use	1
2.1. Requirements	1
2.2. Importing the module	2
2.3. Design principles	2
2.4. Quenya (ᑭᑭᑦ)	3
2.5. Sindarin (ᑭᑭᑦᑭᑦ)—Mode of Gondor	7
2.6. The One Ring inscription	9
3. Math mode?	10
4. How to contribute	11
Bibliography	11

1. Introduction

1.1. ‘Quetta’?

‘*Quetta*’ (ᑭᑭᑦ) means ‘word’ in Quenya [1]¹, one of the fictional languages invented by British writer and philologist J. R. R. Tolkien. It thus seemed fitting for a module aimed at making the process of typing these languages easier.

Words are also, loosely speaking, the base units this module works on, as we shall see in more details below. While its general philosophy is to map each symbol used in Tolkien’s elvish languages to letters from the Latin alphabet, a few word-wise substitution rules were implemented so that, in *most* (but probably not all) cases the correct spelling can be obtained by typing the word phonetically. For the same reason, the mapping generally works on groups of letters rather than individual ones, at least when there is no natural one-to-one mapping between individual symbols.

1.2. The Tengwar script

A proper introduction to Tengwar is way beyond the scope of this document. We refer interested readers to Appendix E of the book [2] and online references such as [Wikipedia](#), [tolkiengateway.net](#), [omniglot.com](#), or [tecendil.com](#).

In short, Tengwar (ᑭᑭᑦᑭᑦ in Quenya mode) is one of the scripts invented by Tolkien, primarily consisting of 36 letters (called *tengwar*; singular: *tengwa* (ᑭᑭᑦᑭᑦ)) and diacritics (*tehtar* (ᑭᑭᑦᑭᑦᑭᑦ ; singular: *tehta* (ᑭᑭᑦᑭᑦᑭᑦ))). There are several ways to relate tengwar to sounds, called *modes*. This module primarily focuses on the Quenya (ᑭᑭᑦ), or ‘classical’, mode (in universe, it is the original way to write tengwar). Support for the other modes described by Tolkien is planned for a future version.

2. How to use

2.1. Requirements

- [Typst](#) version 1.11.0 or up
- The [Tengwar Annatar](#) fonts version 1.20 (support for other Tengwar fonts is not currently planned).

¹<https://www.elfdict.com/w/quetta>

2.2. Importing the module






To import the module, simply add

```
#import "<path>/quetta.typ": *
```

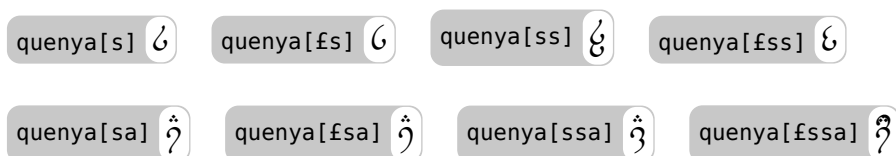
at the top of your .typ file, where <path> is the path to the quetta module.



2.3. Design principles

This module provides one main command for each supported mode—at the moment, only *quenya* is implemented. This command takes text (possibly including formatting) as input and performs the following sequence of operations (not necessarily in this order):

1. Phonetic translation into tengwar and tehtar—for instance, converting *quenya* to .
2. Application of spelling rules—for instance, converting  to .
3. Conversion of numbers in base 12 and conversion to the tengwar number system (see below)—for instance, 144 becomes .
4. Conversion of punctuation symbols—for instance, ? becomes .
5. Adjustments to the position of tehtar and to the kerning between some symbols.

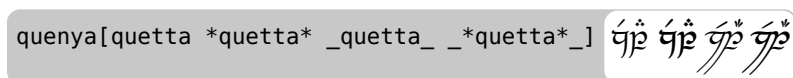
Alternative glyphs, when available, can be obtained with the symbol \mathfrak{f} :



For tengwar associated with a sound starting with ‘k’, the standard glyphs are obtained using the spelling ‘c’ for *calma* () or ‘qu’ for *quessë* (), and the alternatives glyphs with a ‘k’ or ‘kw’:

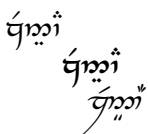


Formatted text is supported, although it is still somewhat experimental:



For a larger amount of text or more involved formatting, it can be easier to use a show rule as follows:

```
#[#show: quenya
  quenya
  #h(1em) *quenya*
  #h(2em) _quenya_
]
```



One limitation of the current implementation is that functions changing other style properties such as text color must be called *after* the conversion function. For instance, a centred 16-points italic version of the Quenya word ‘tengwar’ with a blue-green linear gradient may be obtained as follows:

```
#set align(center)
#text(size: 16pt,
      fill: gradient.linear(blue, green)
) [#box(quenya[_tengwar_])]
]
```



2.4. Quenya (ᑭᑦᑎᑦᑦ)

2.4.1. Generalities

The implementation of the Quenya mode mostly follows Reference [3], summarizing information available in Appendix E of [2] and examples provided in other parts of the book. In universe, the Quenya language and pronunciation evolved somewhat between its origins and events described in [2]; in this implementation, we aim to stay close to the Second Age conventions. Here are a few basic examples:

quenya[quenya] ᑭᑦᑎᑦᑦ

quenya[quetta] ᑭᑭᑦ

quenya[tengwar] ᑭᑦᑎᑦᑦ

quenya[namárië] ᑭᑦᑎᑦᑦᑭᑦᑦ

A full description of the Quenya mode is beyond the scope of this document. As a first approximation, consonant sounds are represented by *tengwar* as follows²:

consonant	tengwa	consonant	tengwa	consonant	tengwa	consonant	tengwa
t	ᑭ	p	ᑭ	c	ᑭ	qu	ᑭ
nd	ᑭᑦ	mb	ᑭᑦ	ng	ᑭᑦ	ngw	ᑭᑦ
th	ᑭ	f	ᑭ	h	ᑭ	hw	ᑭ
nt	ᑭᑦ	mp	ᑭᑦ	nc	ᑭᑦ	nqu	ᑭᑦ
n	ᑭ	m	ᑭ	n	ᑭ	ñw	ᑭ
r	ᑭ	v	ᑭ	y	ᑭ	w	ᑭ

consonant	tengwa
ñ	ᑭᑦ
rd	ᑭᑦ
l	ᑭ
ld	ᑭ
s	ᑭ
ss	ᑭ

Different tengwar are used for the same sounds in different situations; see Section 2.4.2. Vowel sounds are generally represented by a *tehta*, placed either on the previous consonant or a short carrier for a short

²In these tables, the letter ‘u’ represents the consonant sound ‘w’.

input	,	.	-	—	!	?	()	/
output	.	:	~	~	!	?	()	ſ

Note: Generally, parentheses in Quenya are denoted by the single symbol \P —there is no distinction between opening and closing parentheses. We deviate from this convention by mapping the symbol ‘(’ to (and ‘)’ to). The proper Tengwar parenthesis is mapped to ‘/’.

End-of-paragraph symbols can be obtained by combining commas and periods:

quenya[.-] :~ quenya[.,] :: quenya[.] :: quenya[,.,] ::

The decorations \approx and \preceq are obtained using ‘>’ and ‘<’ (or the French quotation marks ‘»’ and ‘«’):

quenya[>quenya<] \approx \preceq

The symbol ‘|’ can be used to prevent glyph combination:

quenya[nn n|n] \approx \preceq quenya[na n|a] \approx \preceq

2.4.5. Number system

Quenya uses a base-12 system, with 12 digits listed in the following table:

0	1	2	3	4	5	6	7	8	9	10	11
๐	๑	๒	๓	๔	๕	๖	๗	๘	๙	๑๐	๑๑

In contrast with the usual system, multi-digit numbers are written (from left to right) from least to most significant digit.

Examples:

quenya[1] ๑ quenya[11] ๑๑ quenya[12] ๑๒ quenya[100] ๑๐๐ quenya[123] ๑๒๓
quenya[144] ๑๔๔ quenya[1000] ๑๐๐๐ quenya[10000] ๑๐๐๐๐ quenya[65537] ๖๕๕๓๗

2.5. Sindarin (𐌸𐌹𐌳𐌰𐌶𐌿𐌺)—Mode of Gondor

The implementation of the Mode of Gondor for the Sindarin language mostly follows Reference [5], to which we refer for more information. In this section, we mostly focus on the differences compared with the implementation of the Quenya language described in Section 2.4. The most important difference is probably the position of *tehtar*: when applicable, a *tehta* is placed on the *previous* *tengwa* in Quenya but on the *following* one in Sindarin.

2.5.1. Generalities

Table of consonants:

consonant	tengwa	consonant	tengwa	consonant	tengwa	consonant	tengwa
t	ᵛ	p	ᵛ	c / k	ᵛ	s	ᵛ
d	ᵛ	b	ᵛ	g	ᵛ	ss	ᵛ
th	ᵛ	f	ᵛ	kh	ᵛ	h	ᵛ
dh	ᵛ	v	ᵛ	gh	ᵛ	rh	ᵛ
n	ᵛ	m	ᵛ	ng	ᵛ	hw	ᵛ
r	ᵛ	lh	ᵛ	w	ᵛ	l	ᵛ

consonant	tengwa(r)
mh	ᵛ
rt	ᵛ

Table of simple vowels:

vowel	short version	long version
a	ă	â
e	é	ê
i	î	ï
o	ô	ö
u	ú	ü
y	ÿ / ʏ	ÿ

Note: This package is mostly case-insensitive. One exception is the letter i in the Mode of Gondor. As an initial i *usually*, but not always, represents the consonant sound ‘y’ in this mode, we use I to represent the tengwa **λ**:

gondor[Iarwain] ᚱᚾᚩᚱᚰᚱ

gondor[iCherdir] ḏýpən

Table of complex vowels:

ai	ei	ui	ae	oe	au
â	é	û	æ	œ	ö

8

2.6. The One Ring inscription

Although the Black Speech is not implemented yet, the One Ring inscription can be reproduced using the Quenya mode as follows:⁷

```
quenya[
  _>Ka|nssangw|ndfrombtaflofkwô, Ka|nssangw|ngwmbetalô«
  #linebreak()#v(0.7em)
  Ka|nssangwthfrquataflofkwô, fNgwa|mbfrossmokii|qufrpetalo_
]
```

Obviously, that's not quite how the ring inscription is supposed to sound. One reason is simply that the Quenya and Black Speech modes have different relations between symbols and sounds: to obtain the same written result, one has to 'transcribe' the phonetic description to how it would be read in the Quenya mode. Another difference is that some of the tengwa forms used in the ring inscription are generally not used in Quenya; we thus use the symbol £ to get variants. We also use £ to switch between *ṛ* and *ṡ*. Finally, words are separated with | to avoid repeated consonants being combined. Here is the result, with a colour gradient in the background to mimic a golden surface and on the text to represent incandescence:



The full code for the above example is:

```
#let ring-inscr-quenya = "
  _>Ka|nssangw|ndfrombtaflofkwô, Ka|nssangw|ngwmbetalô«
  #linebreak()#v(0.7em)
  Ka|nssangwthfrquataflofkwô, fNgwa|mbfrossmokii|qufrpetalo_
"
#set text(top-edge: "ascender", bottom-edge: "descender",
  fill: gradient.linear(rgb(150,0,0), rgb(100,20,0), rgb(255,0,0), space: rgb, angle:
20deg))
#align(center, block(
  fill: gradient.linear(rgb(157,103,7), rgb(250,250,152), rgb(157,103,7), angle: 80deg),
  inset: (top: 1em, left: 1em, right: 1em, bottom: 1.5em),
  radius: 5pt,
  quenya(eval(ring-inscr-quenya, mode: "markup"))))
```

⁷This is obviously a bit of a hack, meant only to show how the limitations of having only one mode implemented can be circumvented. This example is not supposed to be stable and might render differently in a later version.

3. Math mode?

Use of tengwar in math mode is not fully supported, although it should work in simple cases. In math mode, you'll need to apply the conversion function to each part of a formula you want to write in Tengwar, which can be made slightly less cumbersome by redefining it to a shorter command:

```
#let q = quenya
$
#q[6] times #q[7] = #q[42]
$
#v(lem)
$
integral_#q[0]^#q[2] #q[t]^#q[3] upright(d)#q[t]
= [ #q[t]^#q[4] / #q[4] ]_#q[0]^#q[2]
= #q[2]^#q[4] / #q[4]
= #q[16] / #q[4]
= #q[4]
$
#v(lem)
$
#q[t] :
  mat(delim: "(",
    RR & -> RR ;
    #q[a] & |-> #q[a]^#q[123])
=>
(upright(d)#q[t]lr(#q[a])) / (upright(d)#q[a]) = #q[123 a]^#q[122]
$
```

$$\mathbb{C} \times \mathbb{J} = \mathbb{C} \mathbb{C}$$

$$\int_{\mathbb{J}} p^{\mathbb{C}} dp = \left[\frac{p^{\mathbb{L}}}{\mathbb{L}} \right]_{\mathbb{J}}^{\mathbb{C}} = \frac{\mathbb{C}^{\mathbb{L}}}{\mathbb{L}} = \frac{\mathbb{L}^{\mathbb{C}}}{\mathbb{L}} = \mathbb{L}$$

$$p : \left(\begin{array}{c} \mathbb{R} \rightarrow \mathbb{R} \\ \dot{\mathfrak{p}} \mapsto \dot{\mathfrak{p}}^{\mathbb{C}\mathbb{D}} \end{array} \right) \Rightarrow \frac{dp(\dot{\mathfrak{p}})}{d\dot{\mathfrak{t}}} = \mathbb{C}\mathbb{D} \dot{\mathfrak{p}}^{\mathbb{C}\mathbb{D}}$$

Writing math-heavy content in tengwar would probably require a specific module, though, as well as a different tengwar font designed for this purpose (or a math font designed to work well with a tengwar font).

4. How to contribute

Any kind of contribution is warmly welcome! Here are a few ways you can help:

- **Bug reports:** Some text rendering incorrectly in Tengwar? Unexpected formatting? Any other issue with the code or documentation? Please report it! This module was only tested on a very small corpus so far, and identifying any corner case where it does not work as intended is very useful!
- **References:** There is a lot of content available, both online and in printed books and magazines, about the languages invented by Tolkien, how they relate to his works, and their relevance in today's cultural fabric. I am unfortunately not very familiar with them; but if you know good references please let me know and I'll cite them.
- **Language help:** My knowledge of Tengwar and the languages invented by J. R. R. Tolkien is quite superficial, and I may well have missed or misunderstood some of the rules for writing in Tengwar. If you spot anything that looks wrong, please let me know!
- **Implementation:** The Typst code is likely not quite as efficient nor as clean as it could be. If you can see better ways to implement something, please feel free to let me know or to submit a pull request with an improved version!
- **Feature requests:** Any feature request is welcome. I can't promise I'll have the time and knowledge to implement everything that would be nice to have; but if you'd like to see something implemented please let me know—or submit a pull request if you've already implemented it!

እስክን፡ሀ

Bibliography

- [1] “elfdict.com.” [Online]. Available: <https://www.elfdict.com/>
- [2] J.R.R. Tolkien, *The Lord of the Rings*.
- [3] “Quenya Tengwar – A mini course by Eruantalincë.” [Online]. Available: <https://www.councilofelron.com/languages/QTL.htm>
- [4] “Namárië.” [Online]. Available: <https://tolkiengateway.net/wiki/Nam%C3%A1ri%C3%AB>
- [5] Chris McKay, *Tengwar Textbook—Fourth Edition*. 2004.