

# Manual for the quetta (ᑭᑭ) module

Florent Michel

version 0.0.1

## Contents

<b>1. Introduction</b>	<b>1</b>
1.1. ‘Quetta’?	1
1.2. The Tengwar script	1
<b>2. How to use</b>	<b>1</b>
2.1. Requirements	2
2.2. Design principles	2
2.3. Quenya	3
2.4. Sindarin—Mode of Gondor	4
2.5. Sindarin—Mode of Beleriand	4
2.6. Black Speech	5
<b>3. How to contribute</b>	<b>6</b>
<b>Bibliography</b>	<b>6</b>

## 1. Introduction

### 1.1. ‘Quetta’?

‘Quetta’ (ᑭᑭ) means ‘word’ in Quenya [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 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 Reference [2] and online references such as [wikipedia.tolkiengateway.net](http://wikipedia.tolkiengateway.net), [omniglot.com](http://omniglot.com), or [tecendil.com](http://tecendil.com).

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

## 2. How to use

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.

## 2.1. Requirements

- [Typst](#) version 1.11.1 or up
- The [Tengwar Annatar](#) fonts version 1.20

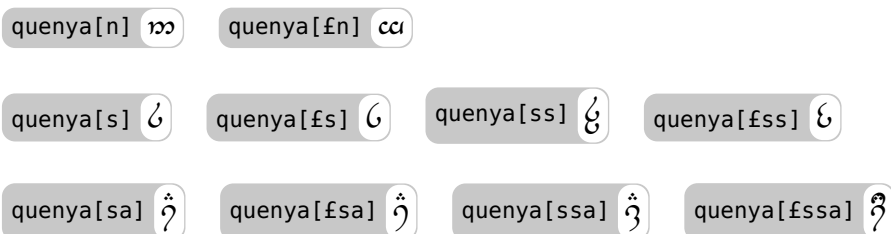
Support for other Tengwar fonts is not currently planned.

## 2.2. Design principles

This module provides one main command for each supported mode—at the moment, only `quenia` 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 `quenia` to `ᑏᑏᑏᑏ`.
2. Application of spelling rules—for instance, converting `ᑏᑏᑏᑏ` to `ᑏᑏᑏᑏ`.
3. Conversion of numbers in base 12 and conversion to the tengwar number system (see below)—for instance, 144 becomes `ᑏᑏᑏ`.
4. Conversion of punctuation symbols—for instance, `?` becomes `ᑏ`.
5. Adjustments to the position of tehtar and to the kerning between some symbols.

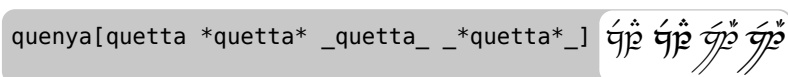
Alternative glyphs, when available, can be obtained with the symbol `£`. For instance, typing `n` produces the tengwa `ᑏ` (*numen*) while typing `£n` produces `ᑏᑏ` (*noldo*):



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



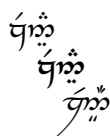
Formatted text is supported, although it is still somewhat experimental:



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

```
#[#show: quenia
  quenia
  #h(1em) *quenia*
  #h(2em) _quenia_
]
```

giving



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 centered 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_])]
]
```



## 2.3. Quenya

The implementation of the Quenya mode follows Reference [3].

Here are a few basic examples:

quenya[aha] ᖃᖅ

quenya[namárië] ᖃᖅᖃᖅᖃᖅ

### 2.3.1. Punctuation

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

quenya[.-] :~

quenya[.,] :·

quenya[.] ::

quenya[,.,] :··

**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 ‘/’.

The decorations  $\approx$  and  $\lesssim$  are obtained using the French quotation marks ‘»’ and ‘«’:

quenya[»quenya«] ᖃᖅᖃᖅᖃᖅ

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

quenya[nn n:n] ᖃᖅ ᖃᖅᖃᖅ

quenya[na n:a] ᖃᖅ ᖃᖅᖃᖅ

### 2.3.2. Number system

Example: quenya[123] ᖃᖅᖃᖅ

### 2.3.3. Example: Namárië

One of the most famous texts written in Quenya is the poem *Namárië* (நம்மர்யீ), originally written in [2]<sup>1</sup> and available for instance in Reference [4]. Below we show the same text without (left) and with (right) the `#show: quenya` command. We use a spacing between line of 0.7em to clearly separate them (some tengwar have a relatively large vertical extension).

#### Namárië

Ai! laurië lantar lassí súrinen,  
yéni únótimë ve rámar aldaron!  
Yéni ve lintë yuldar avánier  
mi oromardi lisse-miruvóreva  
Andúnë pella, Vardo tellumar  
nu luini yassen tintilar i eleni  
ómaryo airtári-lírinen.  
Sí man i yulma nin enquantuva?  
An sí Tintallë Varda Oiolossëo  
ve fanyar máryat Elentári ortanë,  
ar ilyë tier undulávë lumbulë,  
ar sindanóriello caita mornië  
i falmalinnar imbë met, ar hísië  
untúpa Calaciryo míri oialë.  
Sí vanwa ná, Rómello vanwa, Valimar!  
Namárië! Nai hiruvalyë Valimar.  
Nai elyë hiruva. Namárië!

#### நம்மர்யீ

ஈ! டௌரிே டந்தர் லாஸ்சீ சூரினெ,  
யேனி உனோதிமே வெ ராமர் அல்டரெ!  
யேனி வெ லிந்தே யுல்டர் அவானீர்  
மி ஓரொமர்டி லிஸ்செ-மிருவோரேவா  
அநுனே பல்லா, வர்டொ டெல்லுமர்  
நு லுனி யாஸ்செ திந்திலர் இ எலெனி  
ஓமர்யோ அர்தாரி-லீரினெ.  
சீ மான் இ யுல்மா நின் என்குவாதுவா?  
அன் சீ திந்தல்லே வர்டா ஒயோஸ்சே  
வெ ஫ான்யர் மார்யாத் எலெந்தாரி ஓர்தானே,  
அர் இல்யே தீர் உந்டுலாவே லும்புலே,  
அர் சிந்தானோரீல்லோ கைதா மொர்னீ  
இ ஫ால்மாலின்னார் இம்பே மெத், அர் ஹீசிே  
உந்துப்பா காலாசிரியோ மீரி ஓயீலே.  
சீ வான்வா நா, ரோமெல்லோ வான்வா, வாலிமார்!  
நாமாரீே! நை ஹிருவாலியே வாலிமார்.  
நை எல்யே ஹிருவா. நாமாரீே!

### 2.4. Sindarin—Mode of Gondor

Not yet implemented

### 2.5. Sindarin—Mode of Beleriand

Not yet implemented

<sup>1</sup>Book 2, ch. 8 “Farewell to Lórien”

## 2.6. Black Speech

Although the Black Speech is not implemented yet, the One Ring inscription can be reproduced using the Quenya mode as follows:<sup>2</sup>

```
quenya[
  _>Ka:nssangw:ndfrombtaflofkwô, Ka:nssangw:ngwmbetalô«
  #linebreak()#v(0.7em)
  Ka:nssangw:sfrquataflofkwô, 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 **f** to get variants. We also use **f** to switch between **n** and **ŋ**. Finally, words are separated with **:** to avoid repeated consonants being combined. Here is the result, with a color gradient in the background to mimic a golden surface and on the text to represent incandescence:



Not yet implemented

---

<sup>2</sup>This 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. 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!

ᚠᚢᚱᚦᚢᚱ

### 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>