

# TyIPA Manual tyipa 0.1.0

*Write phonetic transcriptions using the IPA.*



## Contents

1	Introduction .....	2
2	Quick guide .....	2
2.1	Importing TyIPA .....	2
2.2	Inserting IPA symbols .....	2
2.3	Adding diacritics to symbols .....	2
2.4	Direct IPA input via <code>ipa.text()</code> .....	3
2.5	Using <code>ipa.text()</code> with a content-block (Experimental!) .....	4
2.6	Choosing a font .....	4
3	List of available IPA symbols .....	4
3.1	Lower-case Latin-based .....	5
3.2	Small-capital Latin-based .....	8
3.3	Greek-based .....	9
3.4	Ligatures, digraphs and others .....	10
3.5	Suprasegmentals .....	11
3.6	Brackets .....	11
3.7	Deprecated ligatures .....	11
3.8	Aliases .....	12
4	List of available IPA diacritics .....	13
4.1	Phonation .....	14
4.2	Place Of Articulation .....	14
4.3	Quality .....	14
4.4	Quantity .....	14
4.5	Release .....	15
4.6	Segmentation .....	15
4.7	Syllabicity .....	15
4.8	Tone .....	15
5	License .....	15
6	Problems & suggestions .....	16
A	Extended example: The North Wind and the Sun .....	17

# 1 Introduction

**TyIPA** is a module for working with the International Phonetic Alphabet (the IPA) in Typst, in a *typsty* style. It provides access to a library of IPA symbols via an interface highly similar to Typst's `std.sym` interface, combined with accenting functions covering the large variety of accents needed for IPA transcriptions. Lastly, to facilitate the ease of coding transcriptions in a Typst document, **TyIPA** provides a conversion function that provides direct access to IPA symbols and accents, similar to how `std.sym` etc. can be accessed directly in math mode.

**TyIPA** relies fully on Unicode and on fonts implementing the IPA-specific parts of UTF-8 in a compliant way. Unlike for example TIPA for LaTeX, **TyIPA** does not attempt to draw, modify or encode its own IPA characters to plug any gaps or fix problems with fonts.


## 2 Quick guide

### 2.1 Importing TyIPA

To start using **TyIPA**, import it in your Typst project:

```
#import "@preview/tyipa:0.1.0" as ipa code

The IPA symbol for a voiceless velar nasal is
#highlight(
  ipa.diac.voiceless-above(
    ipa.sym.n.engma
  )
).


The IPA symbol for a voiceless velar nasal is preview .
```

Note: You should always import **TyIPA** at the module level as shown above (rather than e.g. `#import "@preview/tyipa:0.1.0":*`) because this avoids name collisions that would mean that **TyIPA** shadows items from the standard library. **This manual assumes that you've imported the whole module and aliased it to `ipa` as illustrated above.**

**TyIPA** provides three principal facilities:




- **ipa.sym**: A module providing comprehensive and systematic access to the symbols of the IPA.
- **ipa.diac**: A module providing a comprehensive set of functions to attach the various combining diacritics of the IPA to one or more symbols.
- **ipa.text**: A function that allows convenient access to the IPA symbols and diacritic functions from within text strings.

### 2.2 Inserting IPA symbols

To insert single IPA symbols, use the `#ipa.sym` submodule provided by **TyIPA**. The symbols are named systematically in a way that should be familiar to most phoneticians, though note particularly the choice to use `raised` as a descriptor for superscript items (e.g. `ipa.sym.y.turned.raised` corresponds to the symbol commonly known as “Superscript turned lower-case y”, viz. .


Generally the order is the inverse of how you would name a symbol (i.e. each successive name component narrows down the selection and/or indicates a layer of modification).




Examples:

- `#ipa.sym.r`:  – Lower-case r.
- `#ipa.sym.r.turned`:  – Turned lower-case r.
- `#ipa.sym.r.turned.tail.right`:  – Right-tail turned lower-case r.

For a comprehensive list of symbols provided by **TyIPA**, see Section 3.





### 2.3 Adding diacritics to symbols

For combining diacritics (aka *accents*), **TyIPA** provides the `#ipa.diac` submodule. The submodule contains numerous functions which take a symbol or string as an argument and add augment its input with the diacritic. The diacritic functions are named after the modification they signify according to the IPA, e.g. the acute accent  is encoded by `ipa.diac.high()` because it signifies a high level tone.

Some diacritics (e.g. the ring-diacritic for voicelessness) can be placed either above or below the symbol they modify. In this case, there are two diacritic functions named `...-above()` and `...-below()` respectively, and there is an alias without the `-above/-below` which maps to whatever is the default. For example, `ipa.diac.voiceless-above("x")` yields , `ipa.diac.voiceless-below("x")` yields , and `ipa.diac.voiceless("x")` yields  – the same as

Examples:

Examples:

- `#ipa.diac.high(ipa.sym.epsilon)`:  – Lower-case epsilon with acute accent/high level tone.
- `#ipa.diac.extra-high(ipa.sym.epsilon)`:  – Lower-case epsilon with double-acute accent/extra high level tone.
- `#ipa.diac.voiceless(ipa.sym.n)`:  – Lower-case n with ring below/voiceless diacritic.
- `#ipa.diac.voiceless-above(ipa.sym.engma)`:  – Lower-case engma with ring above/voiceless diacritic above.

A notable exception to the above general pattern is the diacritic function `ipa.diac.tied()`, which always expects a string of length exactly 2, since the tie-bar is meant to show that two symbols represent a single segment (e.g. an affricate or a double articulation).

*Examples:*

- `#ipa.diac.tied(ipa.sym.t + ipa.sym.esh):` **t̥** – Tied symbols lower-case t and lower-case esh.
- `#ipa.diac.tied("gb"):` **gb̥** – Tied symbols lower-case g and lower-case b.
- `#ipa.diac.tied-below("ts"):` **ts̥** – Tied symbols lower-case t and lower-case s.

Though improbable,

```
#highlight(
  ipa.diac.aspirated(
    ipa.diac.falling-rising(
      ipa.diac.voiceless(
        ipa.diac.retracted(
          ipa.sym.o.open
        )
      )
    )
  )
)
```

Though improbable, ʒ̞ʰ represents a falling-rising voiceless aspirated retracted open-mid rounded back vowel.

A comprehensive list of the diacritic functions provided by **TyIPA**, see Section 4.

## 2.4 Direct IPA input via `ipa.text()`

Most often, several IPA symbols and/or diacritics are needed at a time, e.g. to transcribe a word or a short passage. It would be somewhat cumbersome to have to enter this as an extensive series of fully-qualified `ipa.sym.{...}` and `ipa.diac.{...}` calls. So, similar to how Typst provides direct unqualified calls to the builtin `sym.{...}` symbols inside math mode, **TyIPA** provides the **`ipa.text(...)`** function which allows unqualified direct access to the IPA symbols and diacritics.

Say we want to transcribe the French word *bonjour* 'good day'. With fully qualified symbol and diacritic calls we could do this as follows:

```

\[#ipa.sym.b#ipa.diac.nasalized(
  ipa.sym.o.open
)#ipa.sym.syllable-
break#ipa.sym.ezh#ipa.sym.u#ipa.diac.extra-
short(
  ipa.sym.schwa
)#ipa.sym.R.inverted\]

```

The above is horrendously verbose, isn't it? So let's try the same with `#ipa.text(...)` instead.

```
#ipa.text(
    "b nasalized(o.open) syllable-break "
    + "ezh u extra-short(schwa) R.inverted",
    delim: "["
)
```

That's a whole lot simpler. Also note how `ipa.text(...)` takes an optional argument `delim` of type `str` which automatically adds the desired brackets around the transcription.

There are several options for `delim`:

delim: ...	Example output
" / "	/hɛ'ləʊ/
" // "	//hɛ'ləʊ//

delim: ...	Example output
"["	[hɛ'ləʊ]
"[[ "	[[hɛ'ləʊ]]
"( "	(hɛ'ləʊ)
"(( "	((hɛ'ləʊ))
"< "	<hɛ'ləʊ>
"<< "	<<hɛ'ləʊ>>

For a longer example of a transcribed passage of *The Sun and the North Wind* using the `ipa.text(...)` function, see Appendix A.

## 2.5 Using `ipa.text()` with a content-block (Experimental!)

As an experimental feature, you can also use `ipa.text(...)` with an argument of type `content`. For example:

```
#ipa.text(delim: "[")[
  stress-mark s e I f . t i
  ~ stress-mark.secondary I z
  ~ I m stress-mark aspirated(p)
  long(o.open) . t syllabic(n)
  no-release(t)
]
```

[ˈseɪf.ti ɪz ɪmˈpʰɔː.tɪtʰ]

This might often be more convenient to type than using a string, however there are some potential pitfalls and caveats:

- Whereas you can just type a double space (" ") to get a space in the output if passing a string as input, you have to hard code spaces with `~` in content mode, because Typst will already squash whitespace characters together, so they cannot be preserved.
- If you write several paragraphs, each of them will essentially be treated as a separate call to `ipa.text()`, which means that if you assign a delimiter, each paragraph will show the selected brackets.
- The transliteration mechanism will target *anything* that is text within the passed content, so e.g. the bullets of a list might receive unintentional bracketing.
- You'll have to escape characters that you'd normally have to escape in content text, e.g. you have to write `\/` to get the forward slash `/` iff it is the first character on a line, whereas in a string `" / "` would always be fine.

However, while passing content has its limitations and pitfalls, it also has the advantage of allowing further formatting directly within a transcription, for example imagine we wanted to highlight “suffers” and “treigloffobia” in a partial transcription of <He said that she suffers from **treigloffobia**>. With a content-block, we can easily do this:

```
He said that
#ipa.text[
  \esh i length-mark ~
  _s wedge . f schwa z_ ~
  f r.turned wedge m ~
  *t r e I . g l o.open .
  f long(o) . b i . j a*\//
]
```

He said that /ʃiː sʌ.fəz fɪʌm treɪ.glə.foː.bi.jə/.

## 2.6 Choosing a font

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequale doleamus animo, cum corpore dolemus, fieri.

Here are a couple of fonts you might consider trying out:

- Doulos SIL<sup>†</sup>
- Charis<sup>†</sup>
- Gentium<sup>†</sup>
- Andika<sup>†</sup>
- Noto<sup>†</sup> (both Serif and Sans variants)
- Brill<sup>†</sup> (free for non-commercial use only)

There's also a brief but rather handy comparison of a number of font's IPA capabilities on Christopher Bergmann's blog at [isoglosse.de](https://isoglosse.de)<sup>†</sup>.

The font used in this manual is *Gentium*.

## 3 List of available IPA symbols

The following is a comprehensive list of the IPA symbols (and a few extra symbols, namely brackets) made available by **TyIPA**.












Each symbol shows to the left the symbol itself, here in the font “Gentium”, with the dashed lines




















showing the baseline and cap-height, and the dotted lines showing the descender- and ascender-heights of the font.





















Next to each symbol is shown:



















- The symbol's descriptive name in boldface (what a phonetician might call the symbol, which is not usually what the Unicode standard calls it).
- The symbol's canonical name in TyIPA, which can be either used as the fully-qualified `#ipa.sym.{NAME}` anywhere or as just `{NAME}` inside a `#ipa.text("...")` expression. For example, if the name is given as `schwa`, you can use `#ipa.sym.schwa` or `#ipa.text("schwa")` to encode `ə`.
- The symbol's escape code, which is the symbol's Unicode codepoint surrounded by `\u{...}`.
- A note on the symbol's use (if any). For example, whether the symbol is obsolete, deprecated, has a non-standard use, or an alternative name. Note that this does not give aliases, for which see Section 3.8.

### 3.1 Lower-case Latin-based

	<b>Lower-case a</b> Name: <code>a</code> Escape: <code>\u{61}</code>
	<b>Raised lower-case a</b> Name: <code>a.raised</code> Escape: <code>\u{1d43}</code>
	<b>Turned lower-case a</b> Name: <code>a.turn</code> Escape: <code>\u{250}</code>
	<b>Superscript turned lower-case a</b> Name: <code>a.turn.raised</code> Escape: <code>\u{1d44}</code>
	<b>Lower-case b</b> Name: <code>b</code> Escape: <code>\u{62}</code>
	<b>Hook-top lower-case b</b> Name: <code>b.hook-top</code> Escape: <code>\u{253}</code>
	<b>Superscript lower-case b</b> Name: <code>b.hook-top</code> Escape: <code>\u{1d47}</code>
	<b>Lower-case c</b> Name: <code>c</code> Escape: <code>\u{63}</code>
	<b>Lower-case c-cedilla</b> Name: <code>c.cedilla</code> Escape: <code>\u{e7}</code>
	<b>Hook-top lower-case c</b> Name: <code>c.hook-top</code> Escape: <code>\u{188}</code>
	<b>Superscript lower-case c</b> Name: <code>c.raised</code> Escape: <code>\u{1d9c}</code>





















	<b>Stretched lower-case c</b> Name: <code>b.hook-top</code> Escape: <code>\u{297}</code> Note: Obsolete
	<b>Curly-tail lower-case c</b> Name: <code>c.tail.curly</code> Escape: <code>\u{255}</code>
	<b>Superscript curly-tail lower-case c</b> Name: <code>c.tail.curly.raised</code> Escape: <code>\u{1d9d}</code>
	<b>Lower-case d</b> Name: <code>d</code> Escape: <code>\u{64}</code>
	<b>Superscript lower-case d</b> Name: <code>d.raised</code> Escape: <code>\u{1d48}</code>
	<b>Right-tail lower-case d</b> Name: <code>d.tail.right</code> Escape: <code>\u{256}</code>
	<b>Lower-case e</b> Name: <code>e</code> Escape: <code>\u{65}</code>
	<b>Superscript lower-case e</b> Name: <code>e.raised</code> Escape: <code>\u{1d49}</code>
	<b>Reversed lower-case e</b> Name: <code>e.reversed</code> Escape: <code>\u{258}</code>
	<b>Schwa</b> Name: <code>schwa</code> Escape: <code>\u{259}</code> Note: aka <i>turned lower-case e</i>
	<b>Schwa with hook</b> Name: <code>schwa.hook</code> Escape: <code>\u{25a}</code>
	<b>Superscript schwa</b> Name: <code>schwa.raised</code> Escape: <code>\u{1d4a}</code>
	<b>Lower-case f</b> Name: <code>f</code> Escape: <code>\u{66}</code>
	<b>Superscript lower-case f</b> Name: <code>f.raised</code> Escape: <code>\u{1da0}</code>
	<b>Lower-case g</b> Name: <code>g</code> Escape: <code>\u{261}</code> Note: Always single-storey g
	<b>Hook-top lower-case g</b> Name: <code>g.hook-top</code> Escape: <code>\u{260}</code> Note: Always single-storey g
	<b>Raised lower-case g</b> Name: <code>g.raised</code> Escape: <code>\u{1da2}</code> Note: Always single-storey g
	<b>Lower-case h</b> Name: <code>h</code> Escape: <code>\u{68}</code>
	<b>Barred lower-case h</b> Name: <code>h.barred</code> Escape: <code>\u{127}</code>


















	<b>Superscript barred lower-case h</b> Name: h.barred.raised Escape: \u{10795}
	<b>Hengma</b> Name: sym.h.engma Escape: \u{a727}
	<b>Hook-top hengma</b> Name: sym.h.engma.hook-top Escape: \u{267}
	<b>Superscript hook-top hengma</b> Name: h.engma.hook-top.raised Escape: \u{10797}
	<b>Hook-top lower-case h</b> Name: sym.h.hook-top Escape: \u{266}
	<b>Superscript hook-top lower-case h</b> Name: h.hook-top.raised Escape: \u{2b1}
	<b>Superscript lower-case h</b> Name: h.raised Escape: \u{2b0}
	<b>Turned lower-case h</b> Name: h.turn Escape: \u{265}
	<b>Fish-hook turned lower-case h</b> Name: h.turn.fish-hook Escape: \u{2ae}
	<b>Right-tail fish-hook turned lower-case h</b> Name: h.turn.fish-hook.tail.right Escape: \u{2af}
	<b>Superscript turned lower-case h</b> Name: h.turn.raised Escape: \u{1da3}
	<b>Lower-case i</b> Name: i Escape: \u{69}
	<b>Barred lower-case i</b> Name: i.barred Escape: \u{268}
	<b>Superscript barred lower-case i</b> Name: i.barred.raised Escape: \u{1da4}
	<b>Superscript lower-case i</b> Name: i.raised Escape: \u{2071}
	<b>Lower-case j</b> Name: j Escape: \u{6a}
	<b>Barred dotless lower-case j</b> Name: j.dotless.barred Escape: \u{25f}
	<b>Hook-top barred dotless lower-case j</b> Name: j.dotless.barred.hook-top Escape: \u{284}
	<b>Superscript hook-top barred dotless lower-case j</b> Name: j.dotless.barred.hook-top.raised Escape: \u{10798}
	<b>Superscript lower-case j</b> Name: j Escape: \u{2b2}

	<b>Curly-tail lower-case j</b> Name: j.tail.curly Escape: \u{29d}
	<b>Superscript curly-tail lower-case j</b> Name: j.tail.curly.raised Escape: \u{1da8}
	<b>Lower-case k</b> Name: k Escape: \u{6b}
	<b>Hook-top lower-case k</b> Name: k.hook-top Escape: \u{199} Note: Obsolete
	<b>Superscript lower-case k</b> Name: k Escape: \u{1d4f}
	<b>Superscript turned lower-case k</b> Name: k Escape: \u{29e} Note: Obsolete
	<b>Lower-case l</b> Name: l Escape: \u{6c}
	<b>Belted lower-case l</b> Name: l.belted Escape: \u{26c}
	<b>Superscript belted lower-case l</b> Name: l.belted.raised Escape: \u{1079b}
	<b>Superscript lower-case l</b> Name: l.raised Escape: \u{2e1}
	<b>Right-tail lower-case l</b> Name: l.tail.right Escape: \u{26d}
	<b>Superscript right-tail lower-case l</b> Name: l.tail.right.raised Escape: \u{1da9}
	<b>Lower-case l with tilde</b> Name: l.tilde Escape: \u{26b}
	<b>Superscript lower-case l with tilde<sup>1</sup></b> Name: l.tilde.raised Escape: \u{ab5e}
	<b>Lower-case m</b> Name: m Escape: \u{6d}
	<b>Mengma</b> Name: m.engma Escape: \u{271}
	<b>Superscript mengma</b> Name: m.engma.raised Escape: \u{1dac}
	<b>Superscript lower-case m</b> Name: m.raised Escape: \u{1d50}



<sup>1</sup>Apparently some fonts (e.g. *Times New Roman*) have mis-implemented *superscript lower-case l with tilde* as showing a double-tilde; the Unicode codepoint is correct.

**Turned lower-case m**Name: `m.turned`Escape: `\u{26f}`**Superscript turned lower-case m**Name: `m.turned.raised`Escape: `\u{1dac}`**Right-leg turned lower-case m**Name: `m.turned.right-leg`Escape: `\u{270}`**Superscript right-leg turned lower-case m**Name: `m.turned.right-leg.raised`Escape: `\u{1dad}`**Lower-case n**Name: `n`Escape: `\u{6e}`**Engma**Name: `n.engma`Escape: `\u{14b}`**Superscript engma**Name: `n.engma.raised`Escape: `\u{1d51}`**Right-tail lower-case n**Name: `n.tail.right`Escape: `\u{273}`**Superscript right-tail lower-case n**Name: `n.tail.right.raised`Escape: `\u{1daf}`**Left-tail lower-case n**Name: `n.tail.left`Escape: `\u{272}`**Superscript left-tail lower-case n**Name: `n.tail.left.raised`Escape: `\u{1dae}`**Lower-case o**Name: `o`Escape: `\u{6f}`**Barred lower-case o**Name: `o.barred`Escape: `\u{275}`**Superscript barred lower-case o**Name: `o.barred.raised`Escape: `\u{1db1}`**Open lower-case o**Name: `o.open`Escape: `\u{254}`**Superscript open lower-case o**Name: `o.open.raised`Escape: `\u{1d53}`**Superscript lower-case o**Name: `o.raised`Escape: `\u{1d52}`**Slashed lower-case o**Name: `o.slashed`Escape: `\u{f8}`**Superscript slashed lower-case o**Name: `o.slashed.raised`Escape: `\u{107a2}`**Lower-case p**Name: `p`Escape: `\u{70}`**Hook-top lower-case p**Name: `p.hook-top`Escape: `\u{1a5}`Note: `Obsolete`**Superscript lower-case p**Name: `p.raised`Escape: `\u{1d56}`**Lower-case q**Name: `q`Escape: `\u{71}`**Hook-top lower-case q**Name: `q.hook-top`Escape: `\u{2a0}`Note: `Obsolete`**Superscript hook-top lower-case q**Name: `q.hook-top.raised`Escape: `\u{1078d}`Note: `Obsolete`**Superscript lower-case q**Name: `q.raised`Escape: `\u{107a5}`**Lower-case r**Name: `r`Escape: `\u{72}`**Fish-hook r**Name: `r.fish-hook`Escape: `\u{27e}`**Reversed fish-hook r**Name: `r.fish-hook.reversed`Escape: `\u{27f}`Note: `Non-standard, sinology use`**Long-leg lower-case r**Name: `r.long-leg`Escape: `\u{27c}`Note: `Obsolete`**Turned long-leg lower-case r**Name: `r.long-leg.turned`Escape: `\u{27a}`**Superscript lower-case r**Name: `r.raised`Escape: `\u{2b3}`**Right-tail lower-case r**Name: `r.tail.right`Escape: `\u{27d}`**Superscript right-tail lower-case r**Name: `r.tail.right.raised`Escape: `\u{107a9}`**Turned lower-case r**Name: `r.turned`Escape: `\u{279}`**Superscript turned lower-case r**Name: `r.turned.raised`Escape: `\u{2b4}`**Right-tail turned lower-case r**Name: `r.turned.tail.right`Escape: `\u{27b}`**Superscript right-tail turned lower-case r**Name: `r.turned.tail.right.raised`Escape: `\u{2b5}`**Lower-case s**Name: `s`Escape: `\u{73}`

	<b>Superscript lower-case s</b> Name: <code>s.raised</code> Escape: <code>\u{2e2}</code>
	<b>Right-tail lower-case s</b> Name: <code>s.tail.right</code> Escape: <code>\u{282}</code>
	<b>Superscript right-tail lower-case s</b> Name: <code>s.tail.right</code> Escape: <code>\u{1db3}</code>
	<b>Lower-case t</b> Name: <code>t</code> Escape: <code>\u{74}</code>
	<b>Hook-top lower-case t</b> Name: <code>t.hook-top</code> Escape: <code>\u{1ad}</code> Note: <code>Obsolete</code>
	<b>Superscript lower-case t</b> Name: <code>t.raised</code> Escape: <code>\u{1d57}</code>
	<b>Right-tail lower-case t</b> Name: <code>t.tail.right</code> Escape: <code>\u{288}</code>
	<b>Superscript right-tail lower-case t</b> Name: <code>t.tail.right.raised</code> Escape: <code>\u{107af}</code>
	<b>Turned lower-case t</b> Name: <code>t.turn</code> Escape: <code>\u{287}</code> Note: <code>Obsolete</code>
	<b>Lower-case u</b> Name: <code>u</code> Escape: <code>\u{75}</code>
	<b>Barred lower-case u</b> Name: <code>u.barred</code> Escape: <code>\u{289}</code>
	<b>Superscript barred lower-case u</b> Name: <code>u.barred.raised</code> Escape: <code>\u{1db6}</code>
	<b>Superscript lower-case u</b> Name: <code>u.raised</code> Escape: <code>\u{1d58}</code>
	<b>Lower-case v</b> Name: <code>v</code> Escape: <code>\u{76}</code>
	<b>Cursive lower-case v</b> Name: <code>v.cursive</code> Escape: <code>\u{28b}</code>
	<b>Superscript cursive lower-case v</b> Name: <code>v.raised</code> Escape: <code>\u{1db9}</code>
	<b>Hook-top lower-case v</b> Name: <code>v.hook-top</code> Escape: <code>\u{2c71}</code>
	<b>Superscript hook-top lower-case v</b> Name: <code>v.hook-top.raised</code> Escape: <code>\u{107b0}</code>
	<b>Superscript lower-case v</b> Name: <code>v.raised</code> Escape: <code>\u{1d5b}</code>
	<b>Turned lower-case v</b> Name: <code>v.turn</code> Escape: <code>\u{28c}</code>

	<b>Superscript turned lower-case v</b> Name: <code>v.turn.raised</code> Escape: <code>\u{1dba}</code>
	<b>Lower-case w</b> Name: <code>w</code> Escape: <code>\u{77}</code>
	<b>Superscript lower-case w</b> Name: <code>w.raised</code> Escape: <code>\u{2b7}</code>
	<b>Turned lower-case w</b> Name: <code>w.turn</code> Escape: <code>\u{28d}</code>
	<b>Superscript turned lower-case w</b> Name: <code>w.turn.raised</code> Escape: <code>\u{ab69}</code>
	<b>Lower-case x</b> Name: <code>x</code> Escape: <code>\u{78}</code>
	<b>Superscript lower-case x</b> Name: <code>x.raised</code> Escape: <code>\u{2e3}</code>
	<b>Lower-case y</b> Name: <code>y</code> Escape: <code>\u{79}</code>
	<b>Superscript lower-case y</b> Name: <code>y.raised</code> Escape: <code>\u{2b8}</code>
	<b>Turned lower-case y</b> Name: <code>y.turn</code> Escape: <code>\u{28e}</code>
	<b>Superscript turned lower-case y</b> Name: <code>y.turn.raised</code> Escape: <code>\u{107a0}</code>
	<b>Lower-case z</b> Name: <code>z</code> Escape: <code>\u{7a}</code>
	<b>Superscript lower-case z</b> Name: <code>z.raised</code> Escape: <code>\u{2b8}</code>
	<b>Right-tail lower-case z</b> Name: <code>z.tail.right</code> Escape: <code>\u{290}</code>
	<b>Superscript right-tail lower-case z</b> Name: <code>z.tail.right.raised</code> Escape: <code>\u{1dbc}</code>
	<b>Curly-tail lower-case z</b> Name: <code>z.tail.curly</code> Escape: <code>\u{291}</code>
	<b>Superscript curly-tail lower-case z</b> Name: <code>z.tail.curly.raised</code> Escape: <code>\u{1dbd}</code>

## 3.2 Small-capital Latin-based

	<b>Small-capital B</b> Name: <code>B</code> Escape: <code>\u{299}</code>
	<b>Superscript small-capital B</b> Name: <code>B.raised</code> Escape: <code>\u{10784}</code>

**Small-capital G**

Name: G

Escape: `\u{262}`**Hook-top small-capital G**

Name: G.hook-top

Escape: `\u{29b}`**Superscript hook-top small-capital G**

Name: G.hook-top.raised

Escape: `\u{10794}`**Superscript small-capital G**

Name: G.raised

Escape: `\u{10792}`**Small-capital H**

Name: H

Escape: `\u{29c}`**Superscript small-capital H**

Name: H.raised

Escape: `\u{10796}`**Small-capital I**

Name: I

Escape: `\u{26a}`**Superscript small-capital I**

Name: I.raised

Escape: `\u{1da6}`**Small-capital Y**

Name: Y

Escape: `\u{28f}`**Superscript small-capital Y**

Name: Y.raised

Escape: `\u{107b2}`**Small-capital L**

Name: L

Escape: `\u{29f}`**Superscript small-capital L**

Name: L.raised

Escape: `\u{1dab}`**Small-capital N**

Name: N

Escape: `\u{274}`**Superscript small-capital N**

Name: N.raised

Escape: `\u{1db0}`**Small-capital R**

Name: R

Escape: `\u{280}`**Inverted small-capital R**

Name: R.inverted

Escape: `\u{1281}`**Superscript inverted small-capital R**

Name: R.inverted.raised

Escape: `\u{2b6}`**Superscript small-capital R**

Name: R.raised

Escape: `\u{107aa}`**Superscript lower-case alpha**

Name: alpha.raised

Escape: `\u{1d45}`**Turned lower-case alpha**

Name: alpha.turn

Escape: `\u{252}`**Superscript turned lower-case alpha**

Name: alpha.turn.raised

Escape: `\u{1d9b}`**Lower-case beta**

Name: beta

Escape: `\u{3b2}`**Superscript lower-case beta**

Name: beta.raised

Escape: `\u{1d5d}`**Lower-case gamma**

Name: gamma

Escape: `\u{263}`**Superscript lower-case gamma**

Name: gamma.raised

Escape: `\u{2e0}`**Lower-case epsilon**

Name: epsilon

Escape: `\u{25b}`**Superscript lower-case epsilon**

Name: epsilon.raised

Escape: `\u{1d4b}`**Reversed lower-case epsilon**

Name: epsilon.reversed

Escape: `\u{25c}`**Closed reversed lower-case epsilon**

Name: epsilon.reversed.closed

Escape: `\u{25e}`**Superscript reversed lower-case epsilon**

Name: epsilon.reversed.closed.raised

Escape: `\u{1078f}`**Reversed lower-case epsilon with hook**

Name: epsilon.reversed.hook

Escape: `\u{25d}`**Superscript reversed lower-case epsilon**

Name: epsilon.reversed.raised

Escape: `\u{1d9f}`**Lower-case theta**

Name: theta

Escape: `\u{3b8}`**Superscript lower-case theta**

Name: theta.raised

Escape: `\u{1dbf}`**Lower-case iota**

Name: iota

Escape: `\u{269}`

Note: Obsolete

**Lower-case upsilon**

Name: upsilon

Escape: `\u{28a}`**Superscript lower-case upsilon**

Name: upsilon.raised

Escape: `\u{1db7}`**Lower-case phi**

Name: phi


Escape: `\u{278}`

### 3.3 Greek-based


**Lower-case alpha**


Name: alpha


Escape: `\u{251}`


 **Superscript lower-case phi**  
Name: phi.raised  
Escape: \u{1db2}


 **Lower-case chi**  
Name: chi  
Escape: \u{3c7}


 **Superscript lower-case chi**  
Name: chi.raised  
Escape: \u{1d61}


 **Closed lower-case omega**  
Name: omega.closed  
Escape: \u{277}


 **Superscript lower-case omega**  
Name: omega.closed.raised  
Escape: \u{107a4}

 **Superscript exclamation-mark**  
Name: exclamation-mark.raised  
Escape: \u{a71d}

 **Glottal stop**  
Name: glottal-stop  
Escape: \u{294}

 **Superscript glottal stop**  
Name: glottal-stop.raised  
Escape: \u{2c0}


 **Reversed glottal stop**  
Name: glottal-stop.reversed  
Escape: \u{295}


 **Barred reversed glottal stop**  
Name: glottal-stop.reversed.barred  
Escape: \u{2a2}


 **Superscript barred reversed glottal stop**  
Name: glottal-stop.reversed.barred.raised  
Escape: \u{107b4}


 **Superscript reversed glottal stop**  
Name: glottal-stop.reversed.raised  
Escape: \u{2e4}


 **Barred glottal stop**  
Name: glottal-stop.barred  
Escape: \u{2a1}


 **Superscript barred glottal stop**  
Name: glottal-stop.barred.raised  
Escape: \u{107b3}

 **Inverted glottal stop**  
Name: glottal-stop.inverted  
Escape: \u{296}  
Note: Obsolete


 **Lower-case l-ezh ligature**  
Name: lezh  
Escape: \u{26e}

 **Superscript lower-case l-ezh ligature**  
Name: lezh.raised  
Escape: \u{1079e}


 **Lower-case o-e ligature**  
Name: oe  
Escape: \u{153}


 **Superscript lower-case o-e ligature**  
Name: oe.raised  
Escape: \u{a7f9}


 **Small-capital o-e ligature**  
Name: OE  
Escape: \u{276}

 **Superscript small-capital o-e ligature**  
Name: OE.raised  
Escape: \u{107a3}


 **Pipe**  
Name: pipe  
Escape: \u{1c0}

 **Double pipe**  
Name: pipe.double  
Escape: \u{1c1}

 **Superscript double pipe**  
Name: pipe.double.raised  
Escape: \u{1c1}


 **Double-barred pipe**  
Name: pipe.double-barred  
Escape: \u{1c2}


### 3.4 Ligatures, digraphs and others


 **Lower-case a-e ligature**  
Name: ae  
Escape: \u{e6}

 **Superscript lower-case a-e ligature**  
Name: ae.raised  
Escape: \u{10783}


 **Bull's eye**  
Name: bulls-eye  
Escape: \u{298}


 **Superscript bull's eye**  
Name: bulls-eye.raised  
Escape: \u{107b5}


 **Lower-case esh**  
Name: esh  
Escape: \u{283}


 **Superscript lower-case esh**  
Name: esh.raised  
Escape: \u{1db4}


 **Reversed lower-case esh**  
Name: esh.reversed  
Escape: \u{285}


 **Curly-tail lower-case esh**  
Name: esh.tail.curly  
Escape: \u{286}  
Note: Obsolete


 **Lower-case eth**  
Name: eth  
Escape: \u{f0}

 **Superscript lower-case eth**  
Name: eth.raised  
Escape: \u{1d9e}

 **Lower-case ezh**  
Name: ezh  
Escape: \u{292}

 **Superscript lower-case ezh**  
Name: ezh.raised  
Escape: \u{1dbe}

 **Curly-tail lower-case ezh**  
Name: ezh.raised.tail.curly  
Escape: \u{293}  
Note: Obsolete

 **Exclamation mark**  
Name: exclamation-mark  
Escape: \u{1c3}



**Superscript double-barred pipe**  
 Name: `pipe.double-barred.raised`  
 Escape: `\u{1c2}`



**Superscript pipe**  
 Name: `pipe.raised`  
 Escape: `\u{107b6}`



**Ram's horn**  
 Name: `rams-horn`  
 Escape: `\u{264}`



**Superscript ram's horn**  
 Name: `rams-horn.raised`  
 Escape: `\u{10791}`



**Dotted circle placeholder**  
 Name: `placeholder.circle`  
 Escape: `\u{25cc}`



**Non-breaking space (used as a placeholder)**  
 Name: `placeholder.blank`  
 Escape: `\u{a0}`

### 3.5 Suprasegmentals



**Undertie**  
 Name: `undertie`  
 Escape: `\u{203f}`



**Primary stress mark**  
 Name: `stress-mark.primary`  
 Escape: `\u{2c8}`



**Secondary stress mark**  
 Name: `stress-mark.secondary`  
 Escape: `\u{2cc}`



**Syllable break**  
 Name: `Syllable break`  
 Escape: `\u{2e}`



**Long length mark**  
 Name: `length-mark.long`  
 Escape: `\u{2d0}`



**Half-long length mark**  
 Name: `length-mark.half-long`  
 Escape: `\u{2d1}`



**Minor group mark**  
 Name: `group-mark.minor`  
 Escape: `\u{7c}`



**Major group mark**  
 Name: `group-mark.major`  
 Escape: `\u{2016}`



**Upstep mark**  
 Name: `upstep`  
 Escape: `\u{a71b}`



**Downstep mark**  
 Name: `downstep`  
 Escape: `\u{a71c}`



**Global rise mark**  
 Name: `global-rise`  
 Escape: `\u{2197}`



**Global fall mark**  
 Name: `global-fall`  
 Escape: `\u{2198}`



**Extra-high tone-bar**  
 Name: `tone-bar.extra-high`  
 Escape: `\u{2e5}`



**High tone-bar**  
 Name: `tone-bar.high`  
 Escape: `\u{2e6}`



**Mid tone-bar**  
 Name: `tone-bar.mid`  
 Escape: `\u{2e7}`



**Low tone-bar**  
 Name: `tone-bar.low`  
 Escape: `\u{2e8}`



**Extra-low tone-bar**  
 Name: `tone-bar.extra-low`  
 Escape: `\u{2e9}`

### 3.6 Brackets



**Left angle bracket**  
 Name: `bracket.angle.left`  
 Escape: `\u{27e8}`



**Right angle bracket**  
 Name: `bracket.angle.right`  
 Escape: `\u{232a}`



**Left double angle bracket**  
 Name: `bracket.angle.double.left`  
 Escape: `\u{27ea}`



**Right double angle bracket**  
 Name: `bracket.angle.double.right`  
 Escape: `\u{27eb}`



**Left parenthesis**  
 Name: `bracket.paren.left`  
 Escape: `\u{28}`



**Superscript left parenthesis**  
 Name: `bracket.paren.left.raised`  
 Escape: `\u{207d}`



**Right parenthesis**  
 Name: `bracket.paren.right`  
 Escape: `\u{29}`



**Superscript right parenthesis**  
 Name: `bracket.paren.right.raised`  
 Escape: `\u{207e}`



**Left double parenthesis**  
 Name: `bracket.paren.double.left`  
 Escape: `\u{2985}`



**Right double parenthesis**  
 Name: `bracket.paren.double.right`  
 Escape: `\u{2986}`



**Left square bracket**  
 Name: `bracket.square.left`  
 Escape: `\u{5b}`



**Right square bracket**  
 Name: `bracket.square.right`  
 Escape: `\u{5d}`



**Left double square bracket**  
 Name: `bracket.square.double.left`  
 Escape: `\u{27e6}`















**Right double square bracket**  
 Name: `bracket.square.double.right`  
 Escape: `\u{27e7}`

### 3.7 Deprecated ligatures

The following ligatures for affricates are deprecated. It is generally recommended to use a com-

bination of the constituent symbols, possibly with a tie-bar (see `#ipa.diac.tie()`) to indicate that they represent a single entity.

	<b>Lower-case d-ezh ligature</b> Name: <code>dezh</code> Escape: <code>\u{2a4}</code> Note: Deprecated
	<b>Superscript lower-case d-ezh ligature</b> Name: <code>dezh.raised</code> Escape: <code>\u{1078a}</code> Note: Deprecated
	<b>Lower-case d-z ligature</b> Name: <code>dz</code> Escape: <code>\u{2a3}</code> Note: Deprecated
	<b>Superscript lower-case d-z ligature</b> Name: <code>dz.raised</code> Escape: <code>\u{10787}</code> Note: Deprecated
	<b>Lower-case d-curly-z ligature</b> Name: <code>dz.curly</code> Escape: <code>\u{2a5}</code> Note: Deprecated
	<b>Superscript lower-case d-curly-z ligature</b> Name: <code>dz.curly.raised</code> Escape: <code>\u{10789}</code> Note: Deprecated
	Name: <code>tesh</code> Escape: <code>\u{2a7}</code> Note: Deprecated
	Name: <code>tesh.raised</code> Escape: <code>\u{107ae}</code> Note: Deprecated
	<b>Lower-case t-s ligature</b> Name: <code>ts</code> Escape: <code>\u{2a6}</code> Note: Deprecated
	<b>Superscript lower-case t-s ligature</b> Name: <code>ts.raised</code> Escape: <code>\u{107ac}</code> Note: Deprecated
	<b>Lower-case t-curly-tail-c ligature</b> Name: <code>tc.tail.curly</code> Escape: <code>\u{2a8}</code> Note: Deprecated
	<b>Superscript lower-case t-curly-tail-c ligature</b> Name: <code>tc.tail.curly.raised</code> Escape: <code>\u{107ab}</code> Note: Deprecated

### 3.8 Aliases

Several of the symbols listed in the previous sections are also commonly known by other names (an *alias*) or should be logically referenceable by such an alias (e.g. a schwa is really a turned e). In a few cases, an alias might simply represent a common short-hand for the full symbol name (e.g. length-mark for length-mark.primary). For convenience, TyIPA includes the following aliases.

ɑ	Canon:	alpha
	Aliases:	
		– a.cursive
		– a.script
		– a.single-storey

ɑ	Canon:	alpha.raised
	Aliases:	
		– a.cursive.raised
		– a.script.raised
		– a.single-storey.raised
ɒ	Canon:	alpha.turn
	Aliases:	
		– a.cursive.turn
		– a.script.turn
		– a.single-storey.raised
ɒ	Canon:	alpha.turn.raised
	Aliases:	
		– a.cursive.turn.raised
		– a.script.turn.raised
		– a.single-storey.raised
ɔ	Canon:	c.tail.curly
	Aliases:	– c.tail
ɔ	Canon:	c.tail.curly.raised
	Aliases:	– c.tail.raised
ɔ	Canon:	d.tail.right
	Aliases:	– d.retroflex
		– d.tail
ə	Canon:	schwa
	Aliases:	– e.turn
ə	Canon:	schwa.hook
	Aliases:	– e.turn.hook
ə	Canon:	schwa.raised
	Aliases:	– e.turn.raised
ɡ	Canon:	ɡ
	Aliases:	– ɡ.single-storey
ɡ	Canon:	ɡ.hook-top
	Aliases:	– ɡ.single-storey.hook-top
ɡ	Canon:	ɡ.raised
	Aliases:	– ɡ.single-storey.raised
ħ	Canon:	h.engma
	Aliases:	– hengma
ħ	Canon:	h.engma.hook-top
	Aliases:	– hengma.hook-top
ħ	Canon:	h.engma.hook-top.raised
	Aliases:	– hengma.hook-top.raised
ħ	Canon:	h.engma.raised
	Aliases:	– hengma.raised
ɥ	Canon:	h.turn.fish-hook.tail.right
	Aliases:	– h.turn.fish-hook.tail
ɶ	Canon:	j.tail.curly
	Aliases:	– j.tail
ɶ	Canon:	j.tail.curly.raised
	Aliases:	– j.tail.raised
ɷ	Canon:	l.tail.right
	Aliases:	– l.retroflex
		– l.tail

l		Canon: <b>l.tail.right.raised</b> Aliases: – l.retroflex.raised – l.tail.raised
m̥		Canon: <b>m.engma</b> Aliases: – mengma
m̥̥		Canon: <b>m.engma.raised</b> Aliases: – mengma.raised
n̥		Canon: <b>n.engma</b> Aliases: – enmga
n̥̥		Canon: <b>n.engma.raised</b> Aliases: – enmga.raised
ɳ		Canon: <b>n.tail.right</b> Aliases: – n.retroflex – n.tail
ɳ̥		Canon: <b>n.tail.right.raised</b> Aliases: – n.retroflex.raised – n.tail.raised
ɽ		Canon: <b>r.tail.right</b> Aliases: – r.retroflex – r.tail
ɽ̥		Canon: <b>r.tail.right.raised</b> Aliases: – r.retroflex.raised – r.tail.raised
ɿ		Canon: <b>r.turned.tail.right</b> Aliases: – r.turned.retroflex – r.turned.tail
ɿ̥		Canon: <b>r.turned.tail.right.raised</b> Aliases: – r.turned.retroflex.raised – r.turned.tail.raised
ʃ		Canon: <b>s.tail.right</b> Aliases: – s.retroflex – s.tail
ʃ̥		Canon: <b>s.tail.right.raised</b> Aliases: – s.retroflex.raised – s.tail.raised
ʈ		Canon: <b>t.tail.right</b> Aliases: – t.retroflex – t.tail
ʈ̥		Canon: <b>t.tail.right.raised</b> Aliases: – t.retroflex.raised – t.tail.raised
ʊ		Canon: <b>upsilon</b> Aliases: – u.horseshoe
ʊ̥		Canon: <b>upsilon.raised</b> Aliases: – u.horseshoe.raised
ʋ		Canon: <b>v.cursive</b> Aliases: – v.script
ʋ̥		Canon: <b>v.cursive.raised</b> Aliases: – v.script.raised
ʌ		Canon: <b>v.turned</b> Aliases: – wedge

ʌ		Canon: <b>v.turned.raised</b> Aliases: – wedge.raised
z̥		Canon: <b>z.tail.right</b> Aliases: – z.retroflex – z.tail
z̥̥		Canon: <b>z.tail.right.raised</b> Aliases: – z.retroflex.raised – z.tail.raised
Ǿ		Canon: <b>eth</b> Aliases: – edh
◌̯		Canon: <b>placeholder.circle</b> Aliases: – placeholder
ˈ		Canon: <b>stress-mark.primary</b> Aliases: – stress-mark
ː		Canon: <b>length-mark.long</b> Aliases: – length-mark
		Canon: <b>group-mark.major</b> Aliases: – group-mark
ᵹ		Canon: <b>tc.tail.curly</b> Aliases: – tc.tail
ᵹ̥		Canon: <b>tc.tail.curly.raised</b> Aliases: – tc.tail.raised

## 4 List of available IPA diacritics

The following is a comprehensive list of the diacritics implemented by **TyIPA**.







Each entry shows to the left the diacritic mark itself, here in the font “Gentium” applied to a dotted-circle placeholder ‘◌̯’, with the dashed lines showing the baseline and cap-height, and the dotted lines showing the descender- and ascender-heights of the font.

Next to each entry’s illustration is shown:





- The diacritic’s IPA name in boldface.
- The symbol’s use, i.e. what the diacritic signifies when applied to a symbol.
- The symbols canonical name in **TyIPA**, which can be either used as the fully-qualified `ipa.diac.{NAME}(...)` anywhere or as just `{NAME}(...)` inside a `ipa.text("...")` expression. For example, if the name is given as `nasalized`(base: symbol | str), you can use either `ipa.diac.nasalized(ipa.sym.schwa)` or `ipa.text("nasalized(schwa)")` to encode **ă**.

- The diacritic's escape code, which is the Unicode codepoint for the diacritic surrounded by `\u{...}`.
- Aliases that can be used for the same diacritic, if any.
- A note on the diacritic's use (if any). For example, whether it is obsolete or deprecated.




## 4.1 Phonation
















	<b>Superscript h</b> Use: Aspirated Name: <code>aspirated</code> (base: str   symbol) Escape: <code>\u{2b0}</code>
	<b>Subscript umlaut</b> Use: Breathy voiced Name: <code>breathy</code> (base: str   symbol) Escape: <code>\u{324}</code>
	<b>Subscript tilde</b> Use: Creaky voiced Name: <code>creaky</code> (base: str   symbol) Escape: <code>\u{330}</code>
	<b>Subscript wedge</b> Use: Voiced Name: <code>voiced</code> (base: str   symbol) Escape: <code>\u{32c}</code>
	<b>Over-ring</b> Use: Voiceless (above) Name: <code>voiceless-above</code> (base: str   symbol) Escape: <code>\u{30a}</code>
	<b>Under-ring</b> Use: Voiceless Name: <code>voiceless-below</code> (base: str   symbol) Escape: <code>\u{325}</code> Alias: <code>voiceless</code> (base: str   symbol)

## 4.2 Place Of Articulation



	<b>Inverted subscript bridge</b> Use: Apical Name: <code>apical</code> (base: str   symbol) Escape: <code>\u{33a}</code>
	<b>Subscript bridge</b> Use: Dental Name: <code>dental</code> (base: str   symbol) Escape: <code>\u{32a}</code>
	<b>Subscript square</b> Use: Laminal Name: <code>laminal</code> (base: str   symbol) Escape: <code>\u{33b}</code>
	<b>Subscript seagull</b> Use: Linguolabial Name: <code>linguolabial</code> (base: str   symbol) Escape: <code>\u{33c}</code>

## 4.3 Quality

	<b>Subscript plus</b> Use: Advanced Name: <code>advanced</code> (base: str   symbol) Escape: <code>\u{31f}</code>
	<b>Advancing sign</b> Use: Advanced tongue root (ATR) Name: <code>atr</code> (base: str   symbol) Escape: <code>\u{318}</code>
	<b>Umlaut</b> Use: Centralized Name: <code>centralized</code> (base: str   symbol) Escape: <code>\u{308}</code>

	<b>Superscript w</b> Use: Labialized Name: <code>labialized</code> (base: str   symbol) Escape: <code>\u{2b7}</code>
	<b>Subscript left half-ring</b> Use: Less rounded Name: <code>less-rounded</code> (base: str   symbol) Escape: <code>\u{31c}</code>
	<b>Lowering sign</b> Use: Lowered Name: <code>lowered</code> (base: str   symbol) Escape: <code>\u{31e}</code>
	<b>Over-cross</b> Use: Mid-centralized Name: <code>mid-centralized</code> (base: str   symbol) Escape: <code>\u{33d}</code>
	<b>Subscript right half-ring</b> Use: More rounded Name: <code>more-rounded</code> (base: str   symbol) Escape: <code>\u{339}</code>
	<b>Superscript tilde</b> Use: Nasalized Name: <code>nasalized</code> (base: str   symbol) Escape: <code>\u{303}</code>
	<b>Superscript j</b> Use: Palatalized Name: <code>palatalized</code> (base: str   symbol) Escape: <code>\u{2b2}</code>
	<b>Superscript reversed glottal stop</b> Use: Pharyngealized Name: <code>pharyngealized</code> (base: str   symbol) Escape: <code>\u{2e4}</code>
	<b>Raising sign</b> Use: Raised Name: <code>raised</code> (base: str   symbol) Escape: <code>\u{31d}</code>
	<b>Under-bar</b> Use: Retracted Name: <code>retracted</code> (base: str   symbol) Escape: <code>\u{320}</code>
	<b>Right hook</b> Use: Rhoticity Name: <code>rhotic</code> (base: str   symbol) Escape: <code>\u{2de}</code>
	<b>Retroflex hook</b> Use: Diacritic retroflexion Name: <code>retroflex</code> (base: str   symbol) Escape: <code>\u{322}</code>
	<b>Retracting sign</b> Use: Retracted tongue root (RTR) Name: <code>rtr</code> (base: str   symbol) Escape: <code>\u{319}</code>
	<b>Superscript gamma</b> Use: Velarized Name: <code>velarized</code> (base: str   symbol) Escape: <code>\u{2e0}</code>
	<b>Superimposed tilde</b> Use: Velarized or pharyngealized Name: <code>velopharyngealized</code> (base: str   symbol) Escape: <code>\u{334}</code> Alias: <code>dark</code> (base: str   symbol)

## 4.4 Quantity

	<b>Breve</b> Use: Extra-short Name: <code>extra-short</code> (base: str   symbol) Escape: <code>\u{306}</code>
	<b>Half-length mark</b> Use: Half-long Name: <code>half-long</code> (base: str   symbol) Escape: <code>\u{2d1}</code>



### Length mark

Use: Long  
 Name: `long`(base: str | symbol)  
 Escape: `\u{2d0}`

## 4.5 Release



### Apostrophe

Use: Ejective  
 Name: `ejective`(base: str | symbol)  
 Escape: `\u{2bc}`



### Superscript l

Use: Lateral release  
 Name: `lateral-release`(base: str | symbol)  
 Escape: `\u{2e1}`



### Superscript n

Use: Nasal release  
 Name: `nasal-release`(base: str | symbol)  
 Escape: `\u{207f}`



### Corner

Use: No audible release  
 Name: `no-release`(base: str | symbol)  
 Escape: `\u{31a}`

## 4.6 Segmentation



### Top tie bar

Use: Tie bar (above)  
 Name: `tied-above`(base: str)  
 Escape: `\u{361}`  
 Alias: `tied`(base: str)  
 Note: Expects an argument of length exactly 2.



### Bottom tie bar

Use: Tie bar (below)  
 Name: `tied-below`(base: str)  
 Escape: `\u{35c}`  
 Note: Expects an argument of length exactly 2.

## 4.7 Syllabicity



### Subscript arch

Use: Non-syllabic  
 Name: `non-syllabic`(base: str | symbol)  
 Escape: `\u{32f}`



### Syllabicity mark

Use: Syllabic  
 Name: `syllabic`(base: str | symbol)  
 Escape: `\u{329}`

## 4.8 Tone



### Double acute accent (over)

Use: Extra high level tone  
 Name: `extra-high`(base: str | symbol)  
 Escape: `\u{30b}`



### Double grave accent (over)

Use: Extra low level tone  
 Name: `extra-low`(base: str | symbol)  
 Escape: `\u{30f}`



### Circumflex

Use: Falling contour tone  
 Name: `falling`(base: str | symbol)  
 Escape: `\u{302}`



### Acute + grave + acute accent

Use: Falling-rising contour tone  
 Name: `falling-rising`(base: str | symbol)  
 Escape: `\u{1dc9}`



### Acute accent (over)

Use: High level tone  
 Name: `high`(base: str | symbol)  
 Escape: `\u{301}`



### Acute accent + macron

Use: High-mid falling contour tone  
 Name: `high-mid-falling`(base: str | symbol)  
 Escape: `\u{1dc7}`



### Macron + acute accent

Use: High rising contour tone  
 Name: `high-rising`(base: str | symbol)  
 Escape: `\u{1dc4}`



### Grave accent (over)

Use: Low level tone  
 Name: `low`(base: str | symbol)  
 Escape: `\u{300}`



### Grave accent + macron

Use: Low rising contour tone  
 Name: `low-rising`(base: str | symbol)  
 Escape: `\u{1dc5}`



### Macron

Use: Mid level tone  
 Name: `mid`(base: str | symbol)  
 Escape: `\u{304}`



### Macron + grave accent

Use: Mid-low falling contour tone  
 Name: `mid-low-falling`(base: str | symbol)  
 Escape: `\u{1dc6}`



### Wedge, hacek

Use: Rising contour tone  
 Name: `rising`(base: str | symbol)  
 Escape: `\u{30c}`



### Grave + acute + grave accent

Use: Rising-falling contour tone  
 Name: `rising-falling`(base: str | symbol)  
 Escape: `\u{1dc8}`

## 5 License

TyIPA is © 2025 by Florian Breit.

TyIPA, including its documentation and this manual, is licensed under the **MIT License**.

The License text is as follows:

Copyright 2025 Florian Breit

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT

SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The source code is available from <https://github.com/thatfloflo/tyipa><sup>†</sup>.

## 6 Problems & suggestions

**TyIPA** started out as a little experiment to learn more about Typst. There are probably a few bugs I haven't spotted, and there will be things that could've been done better or can be improved.

So whether you have found a bug, or you have a suggestion for improving **TyIPA** (including this manual!), or you have some other question or feedback regarding **TyIPA**, please post these to the issue tracker on the official GitHub repository at <https://github.com/thatfloflo/tyipa><sup>†</sup>. Thank you.

## A Extended example: The North Wind and the Sun

Broad American English transcription of *The North Wind and the Sun* passage from the Handbook of the International Phonetic Association (IPA, 1994, p. 44).

```
#ipa.text("eth schwa stress-mark n o r.turned theta stress-mark.secondary w I n d  
schwa n (theta) schwa stress-mark s v.turned n w schwa.hook d I s stress-mark p j u  
t I n.engma stress-mark w I tied(t esh) w schwa z eth schwa stress-mark s t r.turned  
alpha n.engma g schwa.hook, w epsilon n schwa stress-mark t r.turned ae v schwa l  
schwa.hook stress-mark.secondary k e m schwa stress-mark l alpha n.engma stress-mark  
r.turned ae p t I n schwa stress-mark w o r.turned m stress-mark k l o k.")  
  
#ipa.text("eth e schwa stress-mark g r.turned i d eth schwa t eth schwa stress-mark  
w wedge n h u stress-mark f schwa.hook s t s schwa k stress-mark s i d schwa d I n  
stress-mark m e k I engma eth schwa stress-mark r.turned ae v schwa l schwa.hook  
stress-mark t e k I z stress-mark k l o k stress-mark.secondary alpha f esh epsilon  
d b i k schwa n stress-mark s I d schwa.hook d stress-mark s t r.turned alpha engma g  
schwa.hook eth schwa n eth I stress-mark schwa eth schwa.hook .")  
  
#ipa.text("eth epsilon n eth schwa stress-mark n o r.turned theta stress-  
mark.secondary w I n d stress-mark b l u schwa z i stress-mark k epsilon d , b  
schwa t eth schwa stress-mark m o r.turned h i stress-mark b l u eth schwa stress-  
mark m o r.turned stress-mark k l o s l i d I d eth schwa stress-mark t r.turned v l  
schwa.hook stress-mark f o l d h I z stress-mark k l o k schwa stress-mark r.turned  
a epsilon n d I m ;")  
  
#ipa.text("stress-mark.secondary ae n schwa t stress-mark l ae s t eth schwa stress-  
mark n o r.turned eth stress-mark.secondary w I n d stress-mark.secondary g e v  
stress-mark wedge p eth i schwa stress-mark t epsilon m p t . stress-mark eth epsilon  
n eth schwa stress-mark wedge n stress-mark esh a I n d stress-mark.secondary a  
epsilon t stress-mark w o r.turned m l i, schwa n d I stress-mark m i d i schwa t l i  
eth schwa stress-mark t r.turned ae v l schwa.hook stress-mark t epsilon k stress-  
mark.secondary alpha f I z stress-mark k l o k.")  
  
#ipa.text("schwa n stress-mark s o eth schwa stress-mark n o r.turned theta stress-  
mark.secondary w I n d w schwa z schwa stress-mark b l a I ezh t I k schwa n stress-  
mark f epsilon s eth schwa t eth schwa stress-mark s wedge n w schwa z eth schwa  
stress-mark s t r.turned alpha engma g schwa.hook schwa v eth schwa stress-mark t  
u.")
```

ðə 'noʊθ ,wɪnd ən (θ)ə 'sʌn wə dɪs'pjʊtɪŋ 'wɪtʃ wəz ðə 'stʌŋgə, wɛn ə 'tʌvələ ,kɛm ə'laɪŋ 'æpt ɪn ə 'woʊm  
'klok. preview

ðe ə'gaɪd ðæt ðə 'wʌn hu 'fɜst sək'sɪdəd ɪn 'mekɪŋ ðə 'tʌvələ 'tek ɪz 'klok ,əf ʃʊd bi kən'sɪdəd 'stʌŋgə ðən  
ði 'æðə.

ðɛn ðə 'noʊθ ,wɪnd 'blu əz i 'kʊd, bət ðə 'moʊ hi 'blu ðə 'moʊ 'klosli dɪd ðə 'tʌvlə 'fold hɪz 'klok ə'ʌvnd ɪm;  
,æn ət 'læst ðə 'noʊð ,wɪnd ,gev 'ʌp ði ə'tɛmpt. 'ðɛn ðə 'ʌn 'ʃaɪnd ,ʌt 'woʊmli, ənd ɪ'mɪdiətli ðə 'tʌvələ 'tʊk  
,əf ɪz 'klok.

ən 'so ðə 'noʊθ ,wɪnd wəz ə'blaɪz tɪ kən'fɛs ðæt ðə 'sʌnwəz ðə 'stʌŋgə əv ðə 'tu.