

TyIPA Manual

tyipa 0.1.1

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 ipa.text()	3
2.5	Using ipa.text() 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	Syllability	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.

- `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.raised.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

2 Quick guide

2.1 Importing TyIPA

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

```
#import "@preview/tyipa:0.1.1" as ipa
```

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

The IPA symbol for a voiceless velar nasal is ☉.

Note: You should always import TyIPA at the module level as shown above (rather than e.g. `#import "@preview/tyipa:0.1.1":*`) 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:

the -below-variant, because that is the “default” for the voiceless diacritic.

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/extrahigh 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.

You can also provide a longer string to the diacritic functions, in which case each character of the string will be combined with the diacritic, e.g. `ipa.diac.syllabic("hello")` ⇒ hello.

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): ts` - Tied symbols lower-case t and lower-case esh.
 - `#ipa.diac.tied("gb"):` gb - Tied symbols lower-case g and lower-case b.
 - `#ipa.diac.tied-below("ts"):` ts - Tied symbols lower-case t and lower-case s.

Diacritics can, in principle, be further combined and “stacked”, though it is worth paying attention to how well your chosen font handles such combinations of diacritics, as not all of them manage this equally well.

```
Though improbable,  
#highlight(  
    ipa.diac.aspirated()  
        ipa.diac.falling-rising()  
            ipa.diac.voiceless()  
                ipa.diac.retracted()  
                    ipa.sym.o.open  
                )  
            )  
        )  
    )  
)  
represents a falling-rising voiceless  
aspirated retracted open-mid rounded  
back vowel
```

Though improbable,  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( code
    ipa.sym.o.open
) #ipa.sym.syllable-
break#ipa.sym.ezh#ipa.sym.u#ipa.diac.extra-
short(
    ipa.sym.schwa
```

[bə.ʒuãʃ] preview

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

[hɔ̃ zuɛ̃k] preview

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/
" // "	//hɔ:ləʊ//

delim: ...	Example output
" ["	[hə'ləʊ]
" [["	[[hə'ləʊ]]
" ("	(hə'ləʊ)
" (("	((hə'ləʊ))
" < "	<hə'ləʊ>
" << "	《hə'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)
]
['seif.ti,iz im'pʰɔ:.tntf']
```

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 “*treiglophobia*” in a partial transcription of *He said that she suffers from treiglophobia*. 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 /ʃi: sʌ.fəz fʌm treɪ.glo.fo:.bi.ja/.

2.6 Choosing a font

Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim aequo doleamus animo, cum corpore dolemus, fieri.

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

- [Doulos SIL](#) ↗
- [Charis](#) ↗
- [Gentium](#) ↗
- [Andika](#) ↗
- [Noto](#) ↗ (both Serif and Sans variants)
- [Brill](#) ↗ (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](#) ↗.

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 symbols 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 `ä`.
- 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

	Lower-case a Name: <code>a</code> Escape: <code>\u{61}</code>		Stretched lower-case c Name: <code>b.hook-top</code> Escape: <code>\u{297}</code> Note: Obsolete
	Raised lower-case a Name: <code>a.raised</code> Escape: <code>\u{1d43}</code>		Curly-tail lower-case c Name: <code>c.tail.curly</code> Escape: <code>\u{255}</code>
	Turned lower-case a Name: <code>aturned</code> Escape: <code>\u{250}</code>		Superscript curly-tail lower-case c Name: <code>c.tail.curly.raised</code> Escape: <code>\u{1d9d}</code>
	Superscript turned lower-case a Name: <code>aturned.raised</code> Escape: <code>\u{1d44}</code>		Lower-case d Name: <code>d</code> Escape: <code>\u{64}</code>
	Lower-case b Name: <code>b</code> Escape: <code>\u{62}</code>		Superscript lower-case d Name: <code>d.raised</code> Escape: <code>\u{1d48}</code>
	Hook-top lower-case b Name: <code>b.hook-top</code> Escape: <code>\u{253}</code>		Right-tail lower-case d Name: <code>d.tail.right</code> Escape: <code>\u{256}</code>
	Superscript lower-case b Name: <code>b.hook-top</code> Escape: <code>\u{1d47}</code>		Lower-case e Name: <code>e</code> Escape: <code>\u{65}</code>
	Lower-case c Name: <code>c</code> Escape: <code>\u{63}</code>		Superscript lower-case e Name: <code>e.raised</code> Escape: <code>\u{1d49}</code>
	Lower-case c-cedilla Name: <code>c.cedilla</code> Escape: <code>\u{e7}</code>		Reversed lower-case e Name: <code>e.reversed</code> Escape: <code>\u{258}</code>
	Hook-top lower-case c Name: <code>c.hook-top</code> Escape: <code>\u{188}</code>		Schwa Name: <code>schwa</code> Escape: <code>\u{259}</code> Note: aka turned lower-case e
	Superscript lower-case c Name: <code>c.raised</code> Escape: <code>\u{1d9c}</code>		Schwa with hook Name: <code>schwa.hook</code> Escape: <code>\u{25a}</code>
			Superscript schwa Name: <code>schwa.raised</code> Escape: <code>\u{1d4a}</code>
			Lower-case f Name: <code>f</code> Escape: <code>\u{66}</code>
			Superscript lower-case f Name: <code>f.raised</code> Escape: <code>\u{1da0}</code>
			Lower-case g Name: <code>g</code> Escape: <code>\u{261}</code> Note: Always single-storey g
			Hook-top lower-case g Name: <code>g.hook-top</code> Escape: <code>\u{260}</code> Note: Always single-storey g
			Raised lower-case g Name: <code>g.raised</code> Escape: <code>\u{1da2}</code> Note: Always single-storey g
			Lower-case h Name: <code>h</code> Escape: <code>\u{68}</code>
			Barred lower-case h Name: <code>h.barred</code> Escape: <code>\u{127}</code>

 Superscript barred lower-case h Name: <code>h.barred.raised</code> Escape: <code>\u{10795}</code>	 Curly-tail lower-case j Name: <code>j.tail.curly</code> Escape: <code>\u{29d}</code>
 Hengma Name: <code>sym.h.engma</code> Escape: <code>\u{a727}</code>	 Superscript curly-tail lower-case j Name: <code>j.tail.curly.raised</code> Escape: <code>\u{1da8}</code>
 Hook-top hengma Name: <code>sym.h.engma.hook-top</code> Escape: <code>\u{267}</code>	 Lower-case k Name: <code>k</code> Escape: <code>\u{6b}</code>
 Superscript hook-top hengma Name: <code>h.engma.hook-top.raised</code> Escape: <code>\u{10797}</code>	 Hook-top lower-case k Name: <code>k.hook-top</code> Escape: <code>\u{199}</code> Note: Obsolete
 Hook-top lower-case h Name: <code>sym.h.hook-top</code> Escape: <code>\u{266}</code>	 Superscript lower-case k Name: <code>k</code> Escape: <code>\u{1d4f}</code>
 Superscript hook-top lower-case h Name: <code>h.hook-top.raised</code> Escape: <code>\u{2b1}</code>	 Superscript turned lower-case k Name: <code>k</code> Escape: <code>\u{29e}</code> Note: Obsolete
 Superscript lower-case h Name: <code>h.raised</code> Escape: <code>\u{2b0}</code>	 Lower-case l Name: <code>l</code> Escape: <code>\u{6c}</code>
 Turned lower-case h Name: <code>hturned</code> Escape: <code>\u{265}</code>	 Belted lower-case l Name: <code>l.belted</code> Escape: <code>\u{26c}</code>
 Fish-hook turned lower-case h Name: <code>hturned.fish-hook</code> Escape: <code>\u{2ae}</code>	 Superscript belted lower-case l Name: <code>l.belted.raised</code> Escape: <code>\u{1079b}</code>
 Right-tail fish-hook turned lower-case h Name: <code>hturned.fish-hook.tail.right</code> Escape: <code>\u{2af}</code>	 Superscript lower-case l Name: <code>l.raised</code> Escape: <code>\u{2e1}</code>
 Superscript turned lower-case h Name: <code>hturned.raised</code> Escape: <code>\u{1da3}</code>	 Right-tail lower-case l Name: <code>l.tail.right</code> Escape: <code>\u{26d}</code>
 Lower-case i Name: <code>i</code> Escape: <code>\u{69}</code>	 Superscript right-tail lower-case l Name: <code>l.tail.right.raised</code> Escape: <code>\u{1da9}</code>
 Barred lower-case i Name: <code>i.barred</code> Escape: <code>\u{268}</code>	 Lower-case l with tilde Name: <code>l.tilde</code> Escape: <code>\u{26b}</code>
 Superscript barred lower-case i Name: <code>i.barred.raised</code> Escape: <code>\u{1da4}</code>	 Superscript lower-case l with tilde ¹ Name: <code>l.tilde.raised</code> Escape: <code>\u{ab5e}</code>
 Superscript lower-case i Name: <code>i.raised</code> Escape: <code>\u{2071}</code>	 Lower-case m Name: <code>m</code> Escape: <code>\u{6d}</code>
 Lower-case j Name: <code>j</code> Escape: <code>\u{6a}</code>	 Mengma Name: <code>m.engma</code> Escape: <code>\u{271}</code>
 Barred dotless lower-case j Name: <code>j.dotless.barred</code> Escape: <code>\u{25f}</code>	 Superscript mengma Name: <code>m.engma.raised</code> Escape: <code>\u{1dac}</code>
 Hook-top barred dotless lower-case j Name: <code>j.dotless.barred.hook-top</code> Escape: <code>\u{284}</code>	 Superscript lower-case m Name: <code>m.raised</code> Escape: <code>\u{1d50}</code>
 Superscript hook-top barred dotless lower-case j Name: <code>j.dotless.barred.hook-top.raised</code> Escape: <code>\u{10798}</code>	
 Superscript lower-case j Name: <code>j</code> Escape: <code>\u{2b2}</code>	

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

 Superscript lower-case s Name: <code>s.raised</code> Escape: <code>\u{2e2}</code>	 Superscript turned lower-case v Name: <code>vturned.raised</code> Escape: <code>\u{1dba}</code>
 Right-tail lower-case s Name: <code>s.tail.right</code> Escape: <code>\u{282}</code>	 Lower-case w Name: <code>w</code> Escape: <code>\u{77}</code>
 Superscript right-tail lower-case s Name: <code>s.tail.right</code> Escape: <code>\u{1db3}</code>	 Superscript lower-case w Name: <code>w.raised</code> Escape: <code>\u{2b7}</code>
 Lower-case t Name: <code>t</code> Escape: <code>\u{74}</code>	 Turned lower-case w Name: <code>wturned.raised</code> Escape: <code>\u{28d}</code>
 Hook-top lower-case t Name: <code>t.hook-top</code> Escape: <code>\u{1ad}</code> Note: Obsolete	 Superscript turned lower-case w Name: <code>wturned.raised</code> Escape: <code>\u{ab69}</code>
 Superscript lower-case t Name: <code>t.raised</code> Escape: <code>\u{1d57}</code>	 Lower-case x Name: <code>x</code> Escape: <code>\u{78}</code>
 Right-tail lower-case t Name: <code>t.tail.right</code> Escape: <code>\u{288}</code>	 Superscript lower-case x Name: <code>x.raised</code> Escape: <code>\u{2e3}</code>
 Superscript right-tail lower-case t Name: <code>t.tail.right.raised</code> Escape: <code>\u{107af}</code>	 Lower-case y Name: <code>y</code> Escape: <code>\u{79}</code>
 Turned lower-case t Name: <code>tturned.raised</code> Escape: <code>\u{287}</code> Note: Obsolete	 Superscript lower-case y Name: <code>y.raised</code> Escape: <code>\u{2b8}</code>
 Lower-case u Name: <code>u</code> Escape: <code>\u{75}</code>	 Turned lower-case y Name: <code>yturned.raised</code> Escape: <code>\u{28e}</code>
 Barred lower-case u Name: <code>ubarred.raised</code> Escape: <code>\u{289}</code>	 Superscript turned lower-case y Name: <code>yturned.raised</code> Escape: <code>\u{107a0}</code>
 Superscript barred lower-case u Name: <code>ubarred.raised</code> Escape: <code>\u{1db6}</code>	 Lower-case z Name: <code>z</code> Escape: <code>\u{7a}</code>
 Superscript lower-case u Name: <code>u.raised</code> Escape: <code>\u{1d58}</code>	 Superscript lower-case z Name: <code>z.raised</code> Escape: <code>\u{2b8}</code>
 Lower-case v Name: <code>v</code> Escape: <code>\u{76}</code>	 Right-tail lower-case z Name: <code>ztail.right</code> Escape: <code>\u{290}</code>
 Cursive lower-case v Name: <code>v.cursive</code> Escape: <code>\u{28b}</code>	 Superscript right-tail lower-case z Name: <code>ztail.right.raised</code> Escape: <code>\u{1dbc}</code>
 Superscript cursive lower-case v Name: <code>v.raised</code> Escape: <code>\u{1db9}</code>	 Curly-tail lower-case z Name: <code>ztail.curly</code> Escape: <code>\u{291}</code>
 Hook-top lower-case v Name: <code>v.hook-top</code> Escape: <code>\u{2c71}</code>	 Superscript curly-tail lower-case z Name: <code>ztail.curly.raised</code> Escape: <code>\u{1dbd}</code>
 Superscript hook-top lower-case v Name: <code>v.hook-top.raised</code> Escape: <code>\u{107b0}</code>	
 Superscript lower-case v Name: <code>v.raised</code> Escape: <code>\u{1d5b}</code>	
 Turned lower-case v Name: <code>vturned.raised</code> Escape: <code>\u{28c}</code>	

3.2 Small-capital Latin-based

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

 Small-capital G Name: G Escape: \u{262}	 Superscript lower-case alpha Name: alpha.raised Escape: \u{1d45}
 Hook-top small-capital G Name: G.hook-top Escape: \u{29b}	 Turned lower-case alpha Name: alphaturned Escape: \u{252}
 Superscript hook-top small-capital G Name: Ghook-top.raised Escape: \u{10794}	 Superscript turned lower-case alpha Name: alphaturned.raised Escape: \u{1d9b}
 Superscript small-capital G Name: G.raised Escape: \u{10792}	 Lower-case beta Name: beta Escape: \u{3b2}
 Small-capital H Name: H Escape: \u{29c}	 Superscript lower-case beta Name: betaraised Escape: \u{1d5d}
 Superscript small-capital H Name: H.raised Escape: \u{10796}	 Lower-case gamma Name: gamma Escape: \u{263}
 Small-capital I Name: I Escape: \u{26a}	 Superscript lower-case gamma Name: gammaraised Escape: \u{2e0}
 Superscript small-capital I Name: I.raised Escape: \u{1da6}	 Lowe-case epsilon Name: epsilon Escape: \u{25b}
 Small-capital Y Name: Y Escape: \u{28f}	 Superscript lower-case epsilon Name: epsilon.raised Escape: \u{1d4b}
 Superscript small-capital Y Name: Y.raised Escape: \u{107b2}	 Reversed lower-case epsilon Name: epsilon.reversed Escape: \u{25c}
 Small-capital L Name: L Escape: \u{29f}	 Closed reversed lower-case epsilon Name: epsilon.reversed.closed Escape: \u{25e}
 Superscript small-capital L Name: L.raised Escape: \u{1dab}	 Superscript reversed lower-case epsilon Name: epsilon.reversed.closed.raised Escape: \u{1078f}
 Small-capital N Name: N Escape: \u{274}	 Reversed lower-case epsilon with hook Name: epsilon.reversed.hook Escape: \u{25d}
 Superscript small-capital N Name: N.raised Escape: \u{1db0}	 Superscript reversed lower-case epsilon Name: epsilon.reversed.raised Escape: \u{1d9f}
 Small-capital R Name: R Escape: \u{280}	 Lower-case theta Name: theta Escape: \u{3b8}
 Inverted small-capital R Name: R.inverted Escape: \u{281}	 Superscript lower-case theta Name: theta.raised Escape: \u{1dbf}
 Superscript inverted small-capital R Name: R.inverted.raised Escape: \u{2b6}	 Lower-case iota Name: iota Escape: \u{269}
 Superscript small-capital R Name: R.raised Escape: \u{107aa}	 Lower-case upsilon Name: upsilon Escape: \u{28a}
 Lower-case alpha Name: alpha Escape: \u{251}	 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: <code>phi.raised</code> Escape: <code>\u{1db2}</code>		Superscript exclamation-mark Name: <code>exclamation-mark.raised</code> Escape: <code>\u{a71d}</code>
	Lower-case chi Name: <code>chi</code> Escape: <code>\u{3c7}</code>		Glottal stop Name: <code>glottal-stop</code> Escape: <code>\u{294}</code>
	Superscript lower-case chi Name: <code>chi.raised</code> Escape: <code>\u{1d61}</code>		Superscript glottal stop Name: <code>glottal-stop.raised</code> Escape: <code>\u{2c0}</code>
	Closed lower-case omega Name: <code>omega.closed</code> Escape: <code>\u{277}</code>		Reversed glottal stop Name: <code>glottal-stop.reversed</code> Escape: <code>\u{295}</code>
	Superscript lower-case omega Name: <code>omega.closed.raised</code> Escape: <code>\u{107a4}</code>		Barred reversed glottal stop Name: <code>glottal-stop.reversed.barred</code> Escape: <code>\u{2a2}</code>

3.4 Ligatures, digraphs and others

	Lower-case a-e ligature Name: <code>ae</code> Escape: <code>\u{e6}</code>		Superscript reversed glottal stop Name: <code>glottal-stop.reversed.raised</code> Escape: <code>\u{2e4}</code>
	Superscript lower-case a-e ligature Name: <code>ae.raised</code> Escape: <code>\u{10783}</code>		Barred glottal stop Name: <code>glottal-stop.barred</code> Escape: <code>\u{2a1}</code>
	Bull's eye Name: <code>bulls-eye</code> Escape: <code>\u{298}</code>		Superscript barred glottal stop Name: <code>glottal-stop.barred.raised</code> Escape: <code>\u{107b3}</code>
	Superscript bull's eye Name: <code>bulls-eye.raised</code> Escape: <code>\u{107b5}</code>		Inverted glottal stop Name: <code>glottal-stop.inverted</code> Escape: <code>\u{296}</code> Note: Obsolete
	Lower-case esh Name: <code>esh</code> Escape: <code>\u{283}</code>		Lower-case l-ezh ligature Name: <code>lezh</code> Escape: <code>\u{26e}</code>
	Superscript lower-case esh Name: <code>esh.raised</code> Escape: <code>\u{1db4}</code>		Superscript lower-case l-ezh ligature Name: <code>lezh.raised</code> Escape: <code>\u{1079e}</code>
	Reversed lower-case esh Name: <code>esh.reversed</code> Escape: <code>\u{285}</code>		Lower-case o-e ligature Name: <code>oe</code> Escape: <code>\u{153}</code>
	Curly-tail lower-case esh Name: <code>esh.tail.curly</code> Escape: <code>\u{286}</code> Note: Obsolete		Superscript lower-case o-e ligature Name: <code>oe.raised</code> Escape: <code>\u{a7f9}</code>
	Lower-case eth Name: <code>eth</code> Escape: <code>\u{f0}</code>		Small-capital o-e ligature Name: <code>OE</code> Escape: <code>\u{276}</code>
	Superscript lower-case eth Name: <code>eth.raised</code> Escape: <code>\u{1d9e}</code>		Superscript small-capital o-e ligature Name: <code>OE.raised</code> Escape: <code>\u{107a3}</code>
	Lower-case ezh Name: <code>ezh</code> Escape: <code>\u{292}</code>		Pipe Name: <code>pipe</code> Escape: <code>\u{1c0}</code>
	Superscript lower-case ezh Name: <code>ezh.raised</code> Escape: <code>\u{1dbe}</code>		Doubel pipe Name: <code>pipe.double</code> Escape: <code>\u{1c1}</code>
	Curly-tail lower-case ezh Name: <code>ezh.raised.tail.curly</code> Escape: <code>\u{293}</code> Note: Obsolete		Superscript double pipe Name: <code>pipe.double.raised</code> Escape: <code>\u{1c1}</code>
	Exclamation mark Name: <code>exclamation-mark</code> Escape: <code>\u{1c3}</code>		Double-barred pipe Name: <code>pipe.double-barred</code> Escape: <code>\u{1c2}</code>

	Superscript double-barred pipe Name: <code>pipe.double-barred.raised</code> Escape: <code>\u{1c2}</code>
	Superscript pipe Name: <code>pipe.raised</code> Escape: <code>\u{107b6}</code>
	Ram's horn Name: <code>rams-horn</code> Escape: <code>\u{264}</code>
	Superscript ram's horn Name: <code>rams-horn.raised</code> Escape: <code>\u{10791}</code>
	Dotted circle placeholder Name: <code>placeholder.circle</code> Escape: <code>\u{25cc}</code>
	Non-breaking space (used as a placeholder) Name: <code>placeholder.blank</code> Escape: <code>\u{a0}</code>

3.5 Suprasegmentals

	Undertie Name: <code>undertie</code> Escape: <code>\u{203f}</code>
	Primary stress mark Name: <code>stress-mark.primary</code> Escape: <code>\u{2c8}</code>
	Secondary stress mark Name: <code>stress-mark.secondary</code> Escape: <code>\u{2cc}</code>
	Syllable break Name: <code>Syllable break</code> Escape: <code>\u{2e}</code>
	Long length mark Name: <code>length-mark.long</code> Escape: <code>\u{2d0}</code>
	Half-long length mark Name: <code>length-mark.half-long</code> Escape: <code>\u{2d1}</code>
	Minor group mark Name: <code>group-mark.minor</code> Escape: <code>\u{7c}</code>
	Major group mark Name: <code>group-mark.major</code> Escape: <code>\u{2016}</code>
	Upstep mark Name: <code>upstep</code> Escape: <code>\u{a71b}</code>
	Downstep mark Name: <code>downstep</code> Escape: <code>\u{a71c}</code>
	Global rise mark Name: <code>global-rise</code> Escape: <code>\u{2197}</code>
	Global fall mark Name: <code>global-fall</code> Escape: <code>\u{2198}</code>
	Extra-high tone-bar Name: <code>tone-bar.extra-high</code> Escape: <code>\u{2e5}</code>

	High tone-bar Name: <code>tone-bar.high</code> Escape: <code>\u{2e6}</code>
	Mid tone-bar Name: <code>tone-bar.mid</code> Escape: <code>\u{2e7}</code>
	Low tone-bar Name: <code>tone-bar.low</code> Escape: <code>\u{2e8}</code>
	Extra-low tone-bar Name: <code>tone-bar.extra-low</code> Escape: <code>\u{2e9}</code>

3.6 Brackets

	Left angle bracket Name: <code>bracket.angle.left</code> Escape: <code>\u{27e8}</code>
	Right angle bracket Name: <code>bracket.angle.right</code> Escape: <code>\u{232a}</code>
	Left double angle bracket Name: <code>bracket.angle.double.left</code> Escape: <code>\u{27ea}</code>
	Right double angle bracket Name: <code>bracket.angle.double.right</code> Escape: <code>\u{27eb}</code>
	Left parenthesis Name: <code>bracket.paren.left</code> Escape: <code>\u{28}</code>
	Superscript left parenthesis Name: <code>bracket.paren.left.raised</code> Escape: <code>\u{207d}</code>
	Right parenthesis Name: <code>bracket.paren.right</code> Escape: <code>\u{29}</code>
	Superscript right parenthesis Name: <code>bracket.paren.right.raised</code> Escape: <code>\u{207e}</code>
	Left double parenthesis Name: <code>bracket.paren.double.left</code> Escape: <code>\u{2985}</code>
	Right double parenthesis Name: <code>bracket.paren.double.right</code> Escape: <code>\u{2986}</code>
	Left square bracket Name: <code>bracket.square.left</code> Escape: <code>\u{5b}</code>
	Right square bracket Name: <code>bracket.square.right</code> Escape: <code>\u{5d}</code>
	Left double square bracket Name: <code>bracket.square.double.left</code> Escape: <code>\u{27e6}</code>
	Right double square bracket Name: <code>bracket.square.double.right</code> Escape: <code>\u{27e7}</code>

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.

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

	Lower-case alpha raised Name: <code>alpha.raised</code> Aliases: <ul style="list-style-type: none">- <code>a.cursive.raised</code>- <code>a.script.raised</code>- <code>a.single-storey.raised</code>
	Lower-case alpha turned Name: <code>alphaturned</code> Aliases: <ul style="list-style-type: none">- <code>a.cursive.turned</code>- <code>a.script.turned</code>- <code>a.single-storey.raised</code>
	Lower-case alpha turned raised Name: <code>alphaturned.raised</code> Aliases: <ul style="list-style-type: none">- <code>a.cursive.turned.raised</code>- <code>a.script.turned.raised</code>- <code>a.single-storey.raised</code>
	Lower-case c tail curly Name: <code>c.tail.curly</code> Aliases: <ul style="list-style-type: none">- <code>c.tail</code>
	Lower-case c tail curly raised Name: <code>c.tail.curly.raised</code> Aliases: <ul style="list-style-type: none">- <code>c.tail.raised</code>
	Lower-case d tail right Name: <code>d.tail.right</code> Aliases: <ul style="list-style-type: none">- <code>d.retroflex</code>- <code>d.tail</code>
	Lower-case schwa Name: <code>schwa</code> Aliases: <ul style="list-style-type: none">- <code>e.turned</code>
	Lower-case schwa hook Name: <code>schwahook</code> Aliases: <ul style="list-style-type: none">- <code>e.turned.hook</code>
	Lower-case schwa raised Name: <code>schwaraised</code> Aliases: <ul style="list-style-type: none">- <code>e.turned.raised</code>
	Lower-case g Name: <code>g</code> Aliases: <ul style="list-style-type: none">- <code>g.single-storey</code>
	Lower-case g hook top Name: <code>ghooktop</code> Aliases: <ul style="list-style-type: none">- <code>g.single-storey.hook-top</code>
	Lower-case g raised Name: <code>g.raised</code> Aliases: <ul style="list-style-type: none">- <code>g.single-storey.raised</code>
	Lower-case h engma Name: <code>hengma</code> Aliases: <ul style="list-style-type: none">- <code>hengma</code>
	Lower-case h engma hook top Name: <code>hengmahooktop</code> Aliases: <ul style="list-style-type: none">- <code>hengma.hook-top</code>
	Lower-case h engma hook top raised Name: <code>hengmahooktop.raised</code> Aliases: <ul style="list-style-type: none">- <code>hengma.hook-top.raised</code>
	Lower-case h engma raised Name: <code>hengmaraised</code> Aliases: <ul style="list-style-type: none">- <code>hengma.raised</code>
	Lower-case h turned fish hook tail right Name: <code>hturned.fishhook.tail.right</code> Aliases: <ul style="list-style-type: none">- <code>h.turned.fish-hook.tail</code>
	Lower-case j tail curly Name: <code>j.tail.curly</code> Aliases: <ul style="list-style-type: none">- <code>j.tail</code>
	Lower-case j tail curly raised Name: <code>j.tail.curly.raised</code> Aliases: <ul style="list-style-type: none">- <code>j.tail.raised</code>
	Lower-case l tail right Name: <code>l.tail.right</code> Aliases: <ul style="list-style-type: none">- <code>l.retroflex</code>- <code>l.tail</code>

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.

	Lower-case alpha Name: <code>alpha</code> Aliases: <ul style="list-style-type: none">- <code>a.cursive</code>- <code>a.script</code>- <code>a.single-storey</code>
---	---

l	<i>Canon:</i> l.tail.right.raised <i>Aliases:</i> – l.retroflex.raised – l.tail.raised	Λ	<i>Canon:</i> vturned.raised <i>Aliases:</i> – wedge.raised
m	<i>Canon:</i> m.engma <i>Aliases:</i> – mengma	Ζ	<i>Canon:</i> z.tail.right <i>Aliases:</i> – z.retroflex – z.tail
ŋ	<i>Canon:</i> m.engma.raised <i>Aliases:</i> – mengma.raised	ζ	<i>Canon:</i> z.tail.right.raised <i>Aliases:</i> – z.retroflex.raised – z.tail.raised
ŋ̊	<i>Canon:</i> n.engma <i>Aliases:</i> – enmga	ð	<i>Canon:</i> eth <i>Aliases:</i> – edh
ŋ̊	<i>Canon:</i> n.engma.raised <i>Aliases:</i> – enmga.raised	○	<i>Canon:</i> placeholder.circle <i>Aliases:</i> – placeholder
ɳ	<i>Canon:</i> n.tail.right <i>Aliases:</i> – n.retroflex – n.tail	՚	<i>Canon:</i> stress-mark.primary <i>Aliases:</i> – stress-mark
ɳ̊	<i>Canon:</i> n.tail.right.raised <i>Aliases:</i> – n.retroflex.raised – n.tail.raised	՞	<i>Canon:</i> length-mark.long <i>Aliases:</i> – length-mark
r̊	<i>Canon:</i> r.tail.right <i>Aliases:</i> – r.retroflex – r.tail	՚՚	<i>Canon:</i> group-mark.major <i>Aliases:</i> – group-mark
r̊	<i>Canon:</i> r.tail.right.raised <i>Aliases:</i> – r.retroflex.raised – r.tail.raised	tc̊	<i>Canon:</i> tc.tail.curly <i>Aliases:</i> – tc.tail
ɿ̊	<i>Canon:</i> r.turned.tail.right <i>Aliases:</i> – r.turned.retroflex – r.turned.tail	tc̊	<i>Canon:</i> tc.tail.curly.raised <i>Aliases:</i> – tc.tail.raised
ɿ̊	<i>Canon:</i> r.turned.tail.right.raised <i>Aliases:</i> – r.turned.retroflex.raised – r.turned.tail.raised		
ʂ	<i>Canon:</i> s.tail.right <i>Aliases:</i> – s.retroflex – s.tail		
ʂ̊	<i>Canon:</i> s.tail.right.raised <i>Aliases:</i> – s.retroflex.raised – s.tail.raised		
t̊	<i>Canon:</i> t.tail.right <i>Aliases:</i> – t.retroflex – t.tail		
t̊	<i>Canon:</i> t.tail.right.raised <i>Aliases:</i> – t.retroflex.raised – t.tail.raised		
ʊ	<i>Canon:</i> epsilon <i>Aliases:</i> – u.horseshoe		
v̊	<i>Canon:</i> epsilon.raised <i>Aliases:</i> – u.horseshoe.raised		
ʊ̊	<i>Canon:</i> v.cursive <i>Aliases:</i> – v.script		
v̊	<i>Canon:</i> v.cursive.raised <i>Aliases:</i> – v.script.raised		
ѧ	<i>Canon:</i> vturned <i>Aliases:</i> – wedge		

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 ѧ.

- 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

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

4.2 Place Of Articulation

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

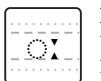
4.3 Quality

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

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

4.4 Quantity

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



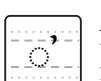
Length mark

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



Acute accent + macron

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



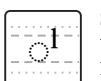
Apostrophe

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



Macron + acute accent

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



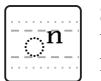
Superscript 1

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



Grave accent (over)

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



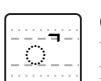
Superscript n

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



Grave accent + macron

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



Corner

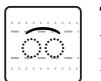
Use: No audible release
Name: `no-release(base: str | symbol)`
Escape: `\u{31a}`



Macron

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

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.



Macron + grave accent

Use: Mid-low falling contour tone

Name: `mid-low-falling(base: str | symbol)`

Escape: `\u{1dc6}`



Wedge, hacek

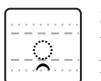
Use: Rising contour tone
Name: `rising(base: str | symbol)`
Escape: `\u{30c}`



Grave + acute + grave accent

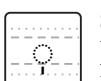
Use: Rising-falling contour tone
Name: `rising-falling(base: str | symbol)`
Escape: `\u{1dc8}`

4.7 Syllability



Subscript arch

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



Syllability mark

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

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:

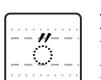
Copyright 2025 Florian Breit

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.

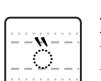
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT

4.8 Tone



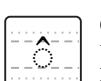
Double acute accent (over)

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



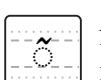
Double grave accent (over)

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



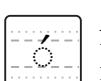
Circumflex

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}`

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>[†].

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>[†]. 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 rturned 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 upsilon  
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 upsilon 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 upsilon 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  
upsilon 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 upsilon 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, wind ən (θ)ə 'sʌn wər dis'pjutɪŋ 'wɪtʃ wəz ðə 'stɹaŋgə, wən ə 'tuævələr, kem ə'laj 'ræpt ɪn ə 'wɔːm 'klok.

ðe ə'gɪd ðət ðə 'wʌn hu 'fərst sək'sidəd ɪn 'mekɪŋ ðə 'rævələr 'tek ɪz 'klok ,af ʃʊd bi kən'sidəd 'stɹaŋgə ðən
ðɪ 'eðər.

ðən ðə 'noʊθ, wind 'blu əz i 'kvɪd, bət ðə 'moʊ hi 'blu ðə 'moʊ 'klosli dɪd ðə 'tuvlər 'fold hɪz 'klok ə'rəʊnd ɪm;
,ən ət 'læst ðə 'noʊð, wind ,gev 'ʌp ðɪ ə'tempt. 'ðən ðə 'ʌn 'ʃaɪnd ,aʊt 'wɔːmli, ənd ɪ'midiətlɪ ðə 'tuævlər 'tʊk
,af ɪz 'klok.

ən 'so ðə 'noʊθ, wind wəz ə'blaiȝ tɪ kən'fes ðət ðə 'sʌnwəz ðə 'stɹaŋgə əv ðə 'tu.

code

preview