



Keyboards and Fonts

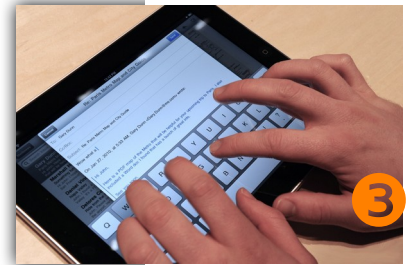
A keyboard is a tool for entering text. It is composed of a hardware part and a software part.

A font is software to make text appear a certain way.



Keyboards - Hardware

The hardware part of a keyboard can be part of the computer①, it can be plugged in to the computer②, or it can be the glass screen of a tablet③.

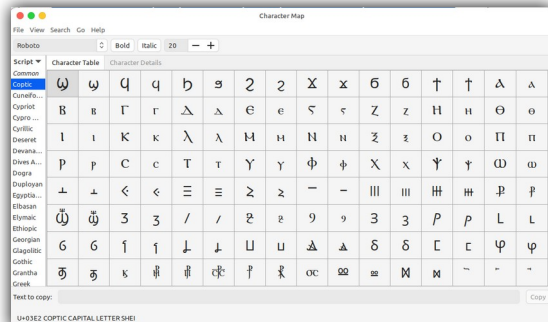


An important detail to remember is that the images on the keys of the keyboard don't necessarily show what will be typed by those keys. That is determined by the software.



Keyboards - Software

When a key is pressed (or a part of the glass screen on a tablet is tapped), software tells the computer what character to make. Modern software does this using **Unicode**.



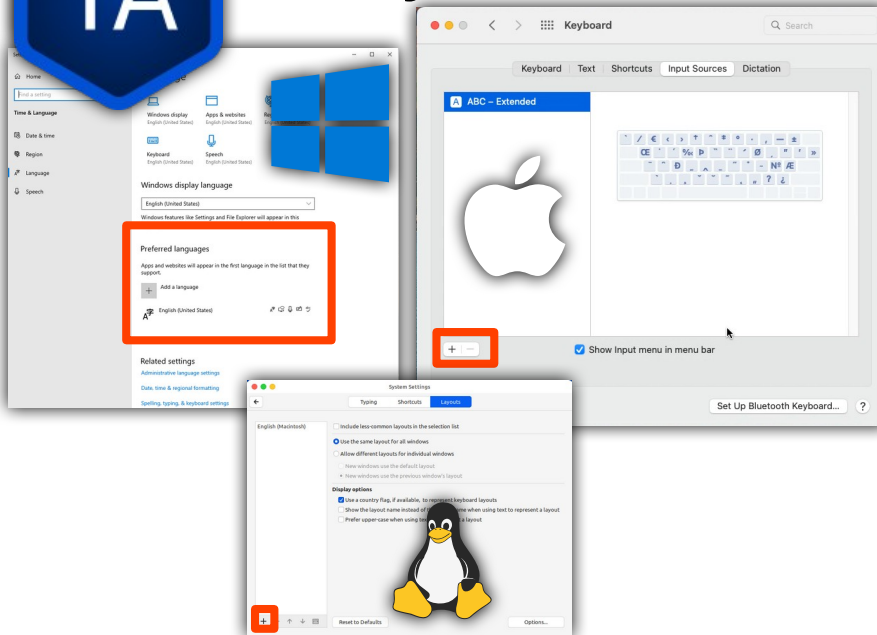
é can be represented in Unicode as U+0065 (LATIN SMALL LETTER E) followed by U+0301 (COMBINING ACUTE ACCENT), but it can also be represented as the precomposed character U+00E9 (LATIN SMALL LETTER E WITH ACUTE)

Unicode is an attempt to allow *all* languages on earth to be represented with *characters* on a computer. Each character is assigned a number, and the keyboard software sends that number to the typing program. In the example above, 0065, 0301, and 00E9 are all numbers of characters.

Unicode is not complete. There are still languages with symbols that are not represented. However, it is the best way for us to share typed data.



Keyboards – Software, pt 2



Each Operating System has its own way of setting up the software for a keyboard.

On Windows, different keyboards are selected by enabling different **languages**.

On macOS and Linux, different keyboard layouts are selected under the **Keyboard** settings.

Different versions of Android put the settings in different places.



Included Keyboards



Windows 11 and macOS ship with around 200 keyboard layouts.



Linux makes nearly 1000 layouts available.

Many of these are very similar to each other, and many of the differing layouts still produce the same characters, but in different places.


We keep using the word “character” instead of “letter” because different writing systems can be very different indeed.

European languages and Korean use an **alphabetic** system, where each consonant or vowel has a shape to represent it.

East Asian languages often use **logographic** or **syllabic** systems, where a shape represents a syllable or an idea.

Languages in India frequently use an **abugida**, where consonants have shapes, and vowels are added onto the consonant.

Hebrew and Arabic use **abjads**, where only consonants have proper shapes. (Vowel additions can be optional.)



The center image is from a PDF that had been made for the people group. We were able to easily select the letters that were used and paste them into our search software to help us find how to type them.



Where's My Character?

The desktop operating systems each have a tool for looking up Unicode characters:



Windows and Linux have apps called **Character Map**. These apps will allow you to search for a character.

[Optional] On macOS it's more complicated. First, **Show Emoji & Symbols** from the Keyboard Menu. Then, click on **Settings** or **More**, depending upon your version. Finally, choose **Customize List**, then scroll down to the group you want to enable. **Unicode** is near the bottom, under **Code Tables**.



How Can I Search?

Experience is the best teacher. When you are more familiar with the writing systems, you will be more able to search:

Latin Small Letter I with Stroke : ï

Latin Small Letter S with Caron : š

Latin Small Letter A with Tilde: ã

Ethiopic Syllable Ddu : ፌ

Unicode characters are described in detail, and by searching for things like strokes, macrons, circumflexes, and other things like that, you can often find the character you're looking for.



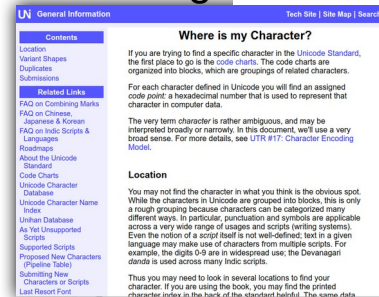
Other Ways to Search



You can use ShapeCatcher.com



or
<https://www.unicode.org/standard/where/>



There are many places online to search for characters. Omniglot.org lists many different writing systems for many different languages. Your program manager or regional director may have access to the SIL Ethnologue, which has information on many languages around the world. Sometimes, searching for the name of the language will give you the help you need.



Typing Unicode Characters

If you want to type a character in Unicode, it is possible to do it directly if you know the code number.

On Windows, you hold the alt key and type **+** and the code number **on the numeric keypad**. Typing the number using the keys at the top of the keyboard will not work. This doesn't work for computers with no numeric keypad.

On macOS there is a special keyboard called **Unicode Hex Input** (under **Input Sources** in the **Keyboard** settings module). With this active, hold option and type the code for the character. You can also type “normal” characters with this input.

On Linux you can either hold **ctrl** and **shift** and type **u + the number** (very complicated) or you can hold the Compose key and various combinations.



Copy-Pasting Characters

Each of the OS tools for looking up a character also gives you the ability to copy that character to paste in to your document.

This isn't helpful if you need to paste type something often.



Windows Keyboards

Once you know what your characters are, how and you type them? Which Windows keyboard will type the character I want?

<https://kbdlayout.info/features/unicode>

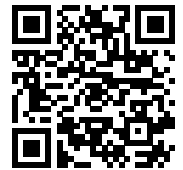


Usually, the best way to type on Windows is to use one of the built-in keyboards. This is guaranteed to be compatible with the OS, and is often an easy way to find what you need.

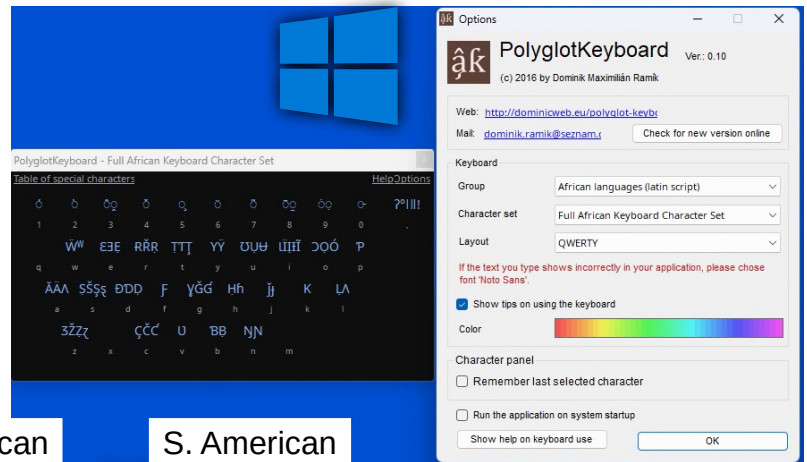
However, finding which keyboard produces what character can be tricky. This website lists all of the Windows keyboards by their Unicode character sets.



Polyglot Keyboard



The PolyglotKeyboard is a good way to type languages from sub-Saharan Africa and much of South America. It is quite easy to use, once you learn how, and it allows a wide range of characters.



African

S. American



Because it allows using the combining characters quite simply, it is a good choice for many people.

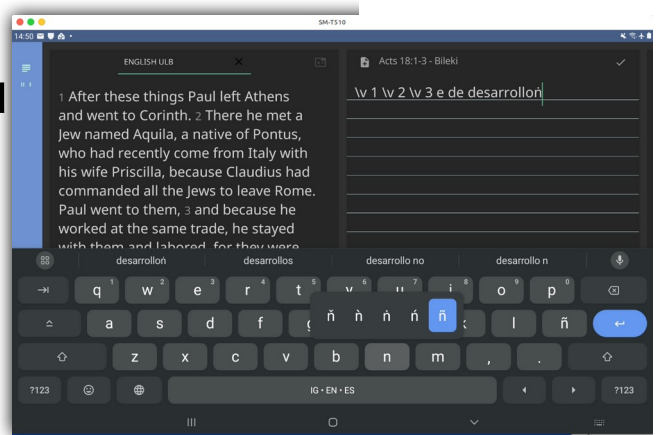
Please Note that this software is for Windows only, although there are two apps for Android: African Keyboard, and South American Keyboard. These two apps are the same product, but for Android.

In the Windows version, you can switch between African Languages, South American Native Languages, and Southeast Asian Languages.



Android Keyboards

Android comes with many keyboards, but not as many as the desktop systems. You can use the African Keyboard or South American Keyboard, or you can see if **Gboard** will do what you need.



Gboard, like most Android keyboards, supports different characters using a long-press of the key that is most similar.

It also comes with many layouts for many languages.



Espanso



Sometimes you don't need a whole keyboard. If you only need a few different characters, it can be easier to make a *shortcut* for the character(s) you need.



<https://espanso.org/install/>



```
- trigger: ":nj"  
  replace: "η"
```

```
- trigger: ":vT"  
  replace: "ř"
```

We like Espanso for the following reasons:

- It is free
- It is available for macOS, Linux, and Windows
- It can be configured very simply using a text file.
- It can be easily turned on and off.

The configuration is done using YAML. The most important thing to know about YAML is that the *spacing* is important. For the simple replacements we will be using, the format is:

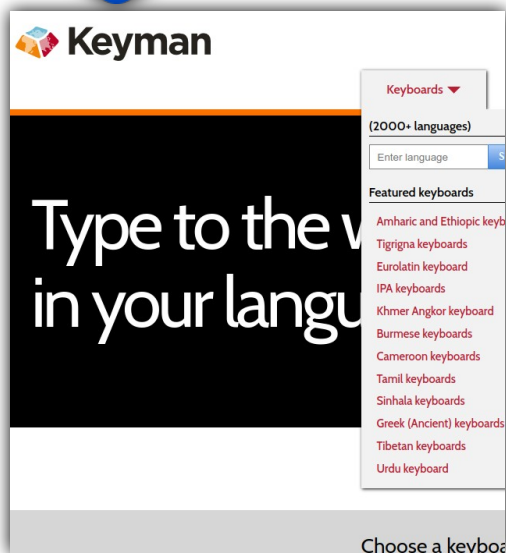
```
<space><space>-<space>trigger: "<trigger text>"
```

```
<space><space><space><space>replace:
```

```
"<replacement character>"
```



Keyman



Finally, if you need a complete keyboard, Keyman software runs on Windows, macOS, Linux *and* Android devices, and has long lists of keyboards.

Keyman not only has thousands of languages supported by its keyboards, it also has tools to create new keyboards. These keyboards can generally be used on *any* of the platforms that can use Keyman.

Creating a keyboard for Keyman is not a simple process. Normally, an existing keyboard or Espanso is a simpler choice.



Fonts

After you have determined how to *type* the language, you will still think about how the language is *displayed*.

۱۱۔ یہ خُدا کے بیٹے یسوع مسیح کی خوشخبری کا شروع ہے۔ ۲۲۔ جیسا
کا شروع ہے۔ ۲۲۔ جیسا یسعیاہ نبی کے صحیفہ میں
لکھا ہے کہ، "دیکھ، میں اپنا پیغمبر تمہارے آگے
بھیجتا ہوں جو تمہاری راہ تیار کرے گا۔ ۳۳۔ بیابان
میں پُکرنے والے کی آواز آتی ہے کہ، 'خُداوند کی راہ تیار
کرو۔ اُس کے راستے سیدھے بناؤ'۔" 4

۱۱۔ یہ خُدا کے بیٹے یسوع مسیح کی خوشخبری کا شروع ہے۔ ۲۲۔ جیسا
یسعیاہ نبی کے صحیفہ میں لکھا ہے کہ، "دیکھ، میں اپنا پیغمبر تمہارے آگے
بھیجتا ہوں جو تمہاری راہ تیار کرے گا۔ ۳۳۔ بیابان میں پُکرنے والے کی
آواز آتی ہے کہ 'خُداوند کی راہ تیار کرو۔ اُس کے راستے سیدھے بناؤ'۔" 4

The text in the two text boxes is the same, but the *font* is different. Sometimes, this can mean the difference between being able to read a text and not.

You can set fonts within BTT-Writer, but to have them display on BIEL takes an additional step.

(The first step is to find a copy of the font that is a) freely available, and b) on the web in an accessible format.

Google's Noto fonts fit both of those categories but there are others. If you find the perfect font that fits (a) but not (b), we may be able to make it work.



Web Fonts and Translation

The *correct* font for this text is called *Nastaliq* or *Nastaleeq*. Google has a version called **Noto Nastaliq Urdu**.

۱۔ یہ خدا کے بیٹے یسوع مسیح کی خوشخبری کا شروع ہے۔ ۲۔ جیسا یسعیاہ نبی کے صحیفہ میں لکھا ہے کہ،
دیکھ، میں اپنا پیغمبر تمہارے آگے بھیجتا ہوں جو تمہاری راہ تیار کرے گا۔ ۳۔ بیابان میں پکارنے والے کی آواز آ
تی ہے کہ 'خداوند کی راہ تیار کرو۔ اُس کے راستے سیدھے بناؤ'۔ ۴۔ یوحنا، بیابان میں پیغمبر دیتا اور گناہوں کی

In order to make this font available for the project on BIEL (or WACS) we need to add something to the project.

In the project folder (found via the data path) we will create a directory called **.apps**
It *must* start with a dot, or period.

In the **.apps** directory we create a directory called **scripture-rendering-pipeline**

In the **scripture-rendering-pipeline** directory, we create a text file called **meta.json**



.apps/scripture-rendering-pipeline/meta.json

```
{  
  "fontUrl" : "https://fonts.googleapis.com/css2?family=Noto+Nastaliq+Urdu:wght@400;700&display=swap",  
  "fontFamily" : "'Noto Nastaliq Urdu', serif;"  
}
```

Here, we point to the place where the web browser will get the font (fonts.googleapis.com) and give specific information about the name of the font and what display characteristics it will have.

This particular font comes from <https://fonts.google.com/>. You can find more information about the different fonts available there, as well as the ways in which you will specify how to display them.

This is an advanced topic. Most of our translation work will display correctly with our default fonts.