

Using Xed as a Typst Editor

By kalwisti

Last Updated: 5 Oct. 2026

Contents

1. Introduction	1
2. Prerequisites	1
3. How to Install the Xed Extra Plugins	2
3.1. Download the Plugins	2
3.2. Install the Plugins	3
3.3. Enable the Plugins	5
4. Let's Begin Editing	5
5. Dealing with Errors	7
6. Conclusion	8

1. Introduction

In the Typst community, the general consensus is that the official [Typst web app](#) and/or VSCode [VSCodium] plus the Tinymist extension offer the smoothest writing experience.¹ However, if you have a [local Typst installation](#), you may use other editors if you wish. Some alternatives include [vim with the Typst plugin](#), [Geany](#) and [Kate](#). I will offer another option in this article: Xed with some extra plugins created by [Gabriell Araujo](#).

[Xed](#) (aka Text Editor) is a lightweight text editor based on [Pluma](#) (the default text editor of the MATE desktop environment). Xed is part of the [XApp Project](#) which was begun in 2016 by the Linux Mint team. XApps are basic applications designed to work consistently across different GTK-based desktop environments like Cinnamon, MATE, and Xfce. The primary goal of XApps is to provide a consistent user experience regardless of the Linux distribution or DE being used.²

Xed supports most standard editing features, plus a flexible plugin system which can be used to dynamically add new features to the editor. Gabriell Araujo, a Brazilian developer, created some handy plugins for Xed which enhance its functionality. They can be downloaded from his [GitHub repository](#) and easily installed, without requiring any recompiling or patching of Xed.

This method uses Xed — along with g_araujo's plugins and the embedded Typst CLI — and a separate PDF viewer (as the external Preview app), to edit your local Typst files. The preview is asynchronous, i.e., not in real time, but after you make an edit in Xed and save your file, the PDF will auto-update.

If you make a syntax error in your Typst source file, the PDF will not compile. You can then check output in the embedded Terminal (where the Typst CLI is running) for detailed error messages about the problem and how to fix it.

2. Prerequisites

- A local Typst installation with the Typst pre-built binary (ver. 0.15.1, 17 July 2026). Please see my [previous tutorial](#) for the installation procedure on Linux Mint.
- Xed. (As of Oct. 2026, its version is 3.8.9)

¹VSCodium with the Tinymist extension bundles everything you need, and it is updated frequently. It includes instant preview, monitors your source file for changes and a jump-to-location feature which allows you to jump between the Typst source file and the corresponding location in the Preview pane.

²In addition to Xed, other XApps include: Xreader (a document and PDF reader based on Atril), Xviewer (an image viewer based on the Eye of GNOME) and Pix (a photo organizer based on gThumb).

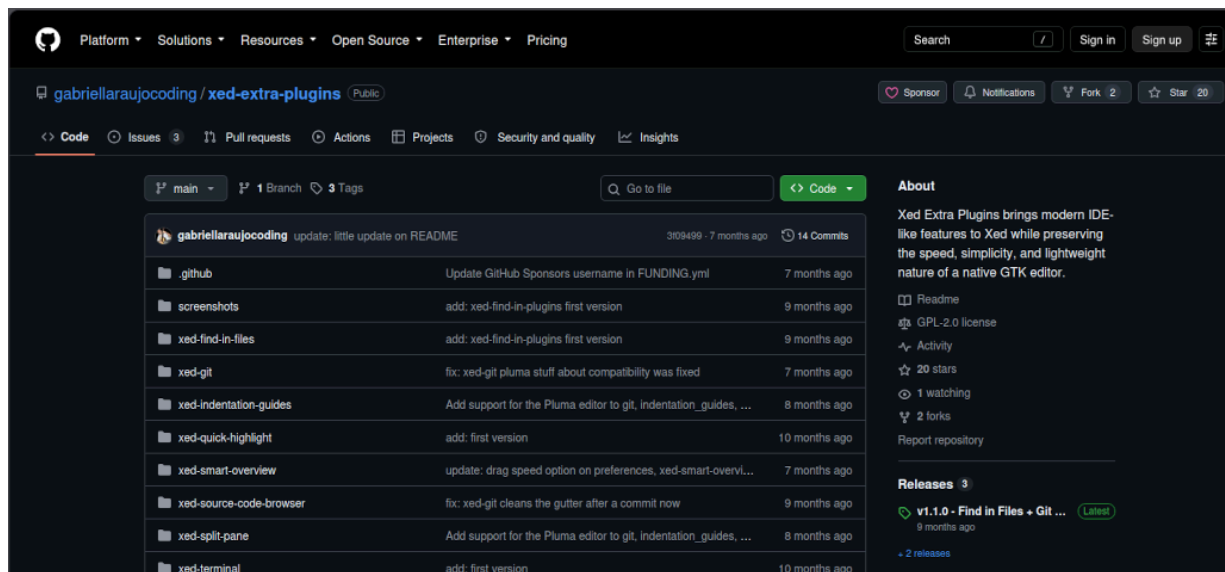
- xed-extra-plugins from Gabriell Araujo's [GitHub repository](#)
 - I will walk through the installation steps in Section 3.
 - The plugins require some additional dependencies which are listed [here](#).
- PDF Reader (such as Xreader, Evince, Atril)

Note:

This procedure was tested within Mint 22.3 Cinnamon and MATE. However, it should work identically with the Xfce and LMDE editions.

3. How to Install the Xed Extra Plugins

3.1. Download the Plugins



Begin by downloading the plugins from Gabriell Araujo's [GitHub repository](#). He lists three different options; I chose the [recommended option](#) (Git clone).

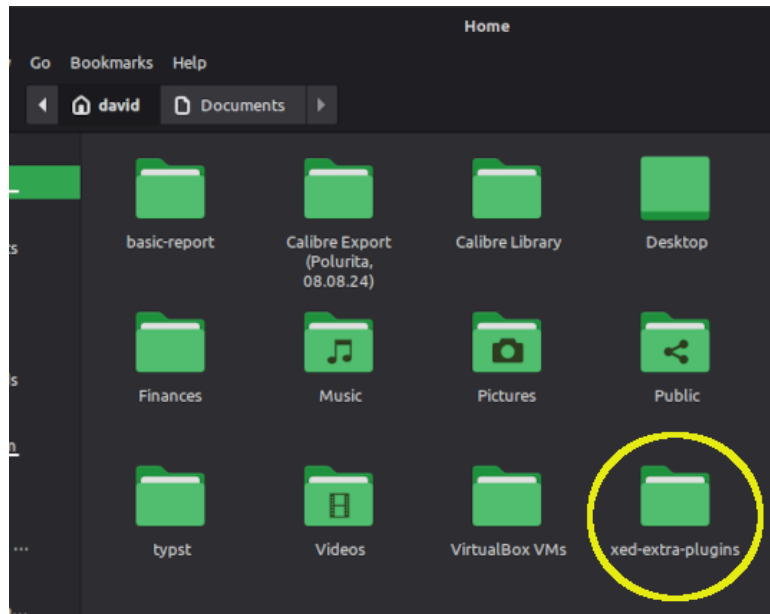
- Verify that you have the git package installed on your system. Next, open a Terminal and issue the command below (as a regular user):

```
$ git clone --depth 1 https://github.com/gabriellaraujocoding/xed-extra-plugins.git
```

This will download the repository contents and create a new directory called xed-extra-plugins inside your /home directory.

You should see output similar to the following:

```
david@ASUSTek-mint:~$ git clone --depth 1 https://github.com/gabriellaraujocoding/xed-extra-plugins.git
Cloning into 'xed-extra-plugins'...
remote: Enumerating objects: 59, done.
remote: Counting objects: 100% (59/59), done.
remote: Compressing objects: 100% (58/58), done.
remote: Total 59 (delta 0), reused 42 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (59/59), 2.38 MiB | 7.12 MiB/s, done.
```

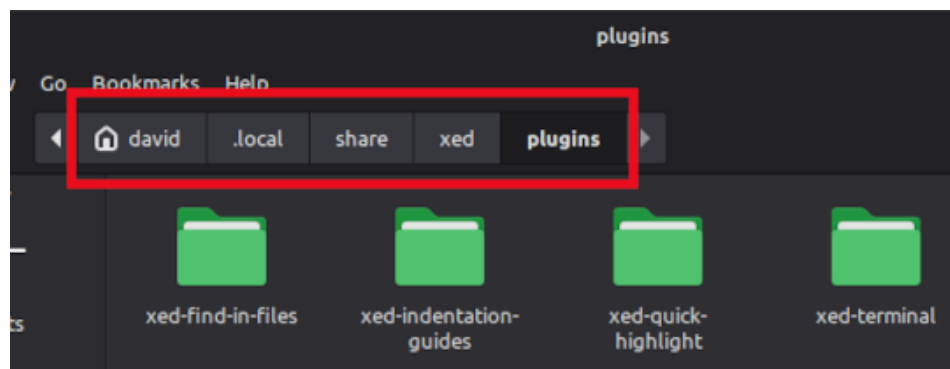


3.2. Install the Plugins

You must also create a (hidden) directory for the new Xed plugins. Open a Terminal and type the command:

```
$ mkdir -p ~/.local/share/xed/plugins
```

(A hidden directory is just a directory whose name begins with a dot [“.”], e.g., “.config” and “.local”. To view hidden files and hidden directories, open your File Manager [Nemo, Thunar, Caja] and type **Ctrl + H**).³



- Next, issue the **cd** command to navigate to the new xed-extra-plugins in your /home directory:

```
david@ASUStek-mint:~$ cd xed-extra-plugins
```

- Copy the plugin folders you want to the new hidden /xed/plugins/ directory. For example:

³The reason for hiding these directories is to prevent accidental deletion or modification by casual users. They are typically used by applications to store settings that are not meant to be directly manipulated by the end-user.

```
david@ASUSTek-mint:~$ cd xed-extra-plugins

david@ASUSTek-mint:~/xed-extra-plugins$ ls
LICENSE          xed-find-in-files      xed-smart-overview
LICENSE_NOTICE   xed-git                 xed-source-code-browser
README.md        xed-indentation-guides xed-split-pane
screenshots      xed-quick-highlight    xed-terminal

david@ASUSTek-mint:~/xed-extra-plugins$ cp -r xed-indentation-guides ~/.local/share/xed/
plugins/

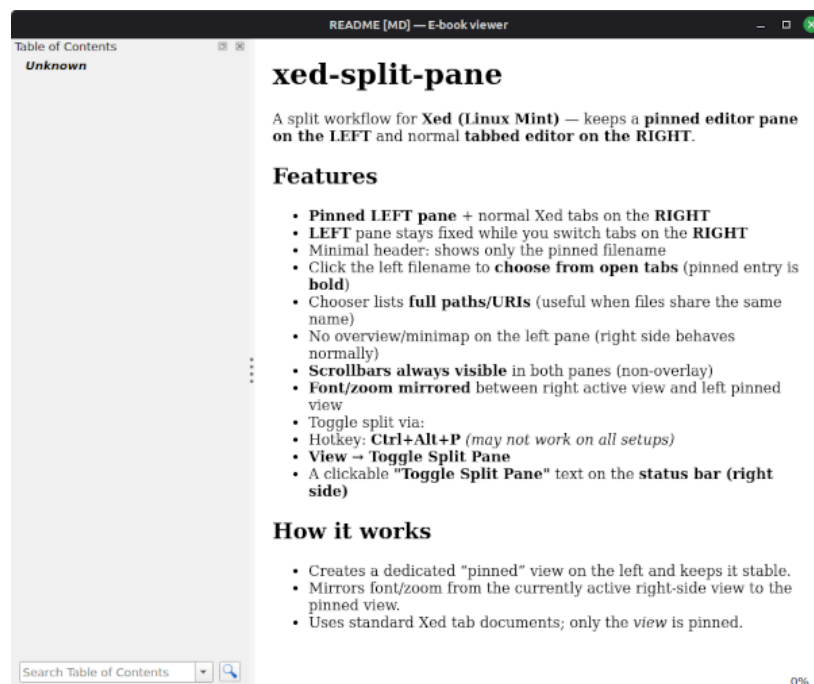
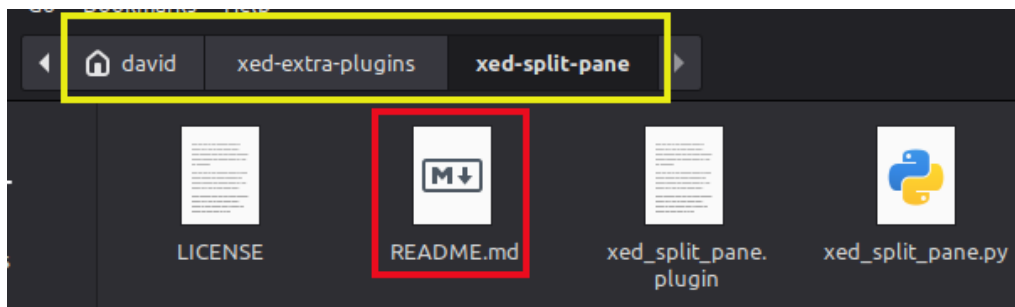
david@ASUSTek-mint:~/xed-extra-plugins$ cp -r xed-terminal ~/.local/share/xed/plugins/

david@ASUSTek-mint:~/xed-extra-plugins$ cp -r xed-quick-highlight ~/.local/share/xed/plugins/
```

Tip:

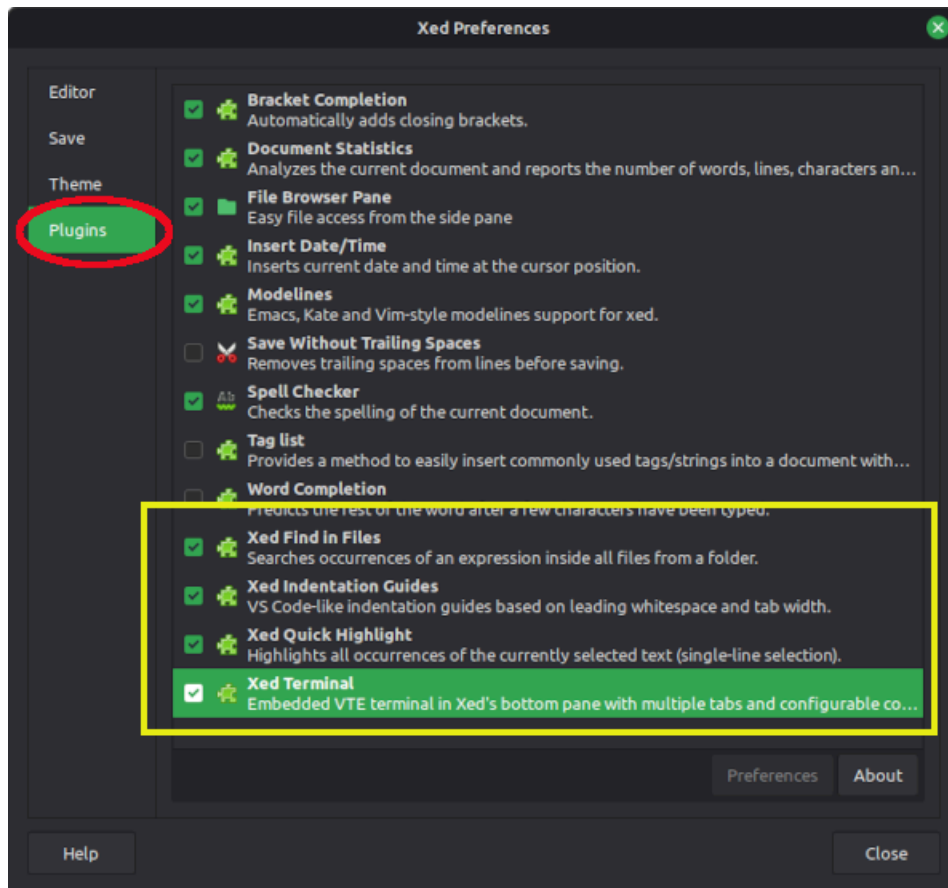
The developer has provided helpful instructions on the use of each plugin in a **README.md** file, located in the various subfolders of the `~/xed-extra-plugins` directory.

The following screenshots show the **README.md** file for the `xed-split-pane` plugin, which makes it easy to copy and paste text from one Xed editor pane to another.



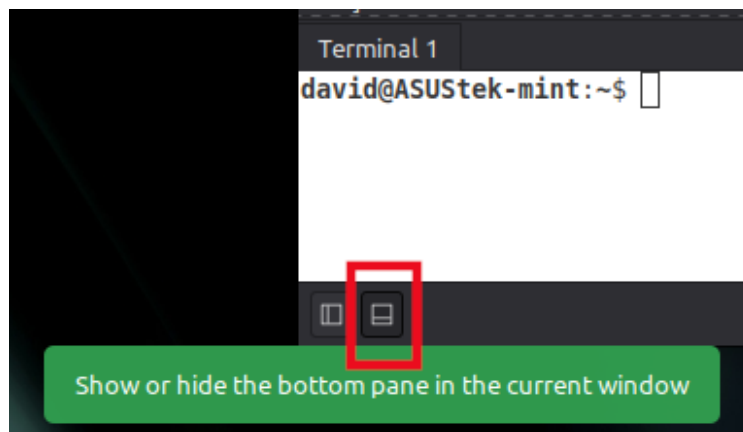
3.3. Enable the Plugins

Finally, enable the plugins (via Xed's **Edit** menu > choose **Preferences** > then **Plugins**) and restart Xed.



4. Let's Begin Editing ...

- First, open your local Typst file with Xed.
- Click on this icon in Xed's bottom panel to open the embedded VTE terminal:



- In the embedded Terminal, issue the appropriate commands to navigate to your Typst source file.
- Generate a PDF file by issuing the command:

```
$ typst compile your-filename.typ
```

- Leave the embedded Terminal open after the PDF is compiled.
- Open the Typst-generated PDF with your favorite PDF Reader (Xreader, Evince, Atril).
- Leave the PDF Reader open on your desktop.

Optional but recommended:

If you wish to add a bit of color syntax highlighting to the displayed source file, you can set the document's filetype to **Markdown** (from Xed's **View** menu > choose **Highlight Mode** > then **Markdown**). The syntax highlighting is minimal compared to the detailed highlighting that you see in VSCode + TinyMist; however, it adds welcome visual contrast to the Typst source file (*personal opinion*).

- Go back to Xed's embedded Terminal and issue the command:

```
$ typst watch your-filename.typ
```

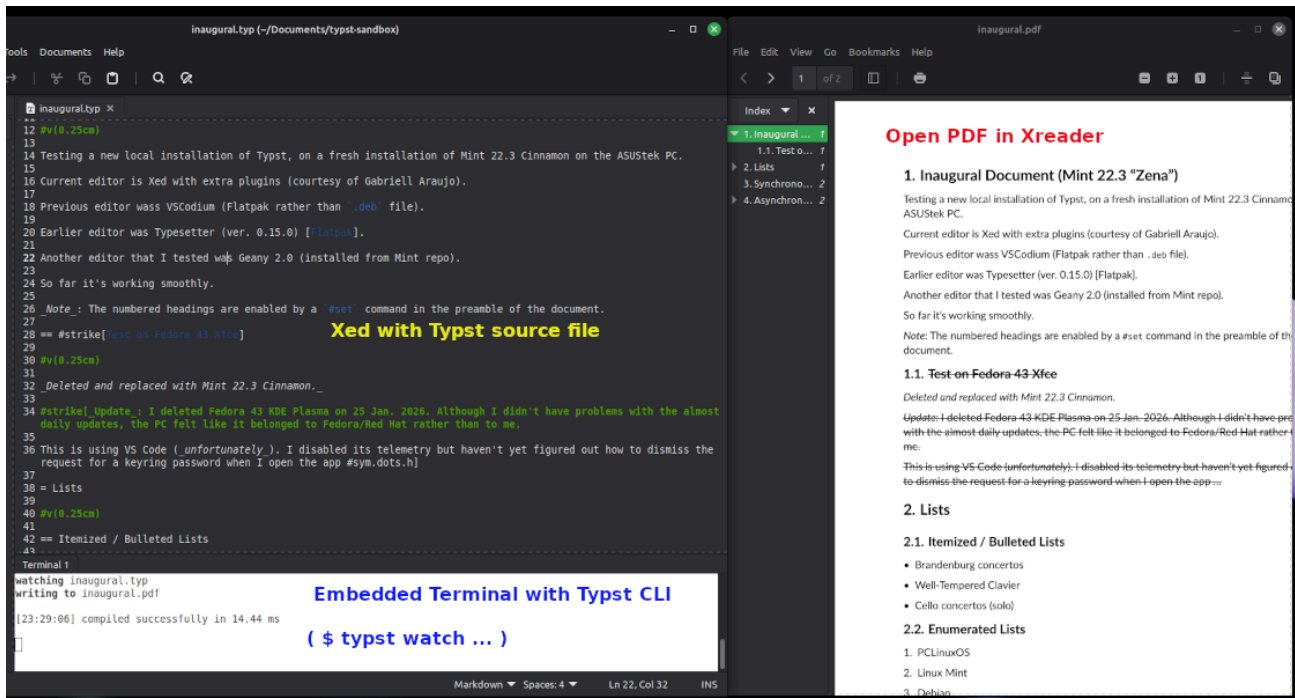
You should see output similar to this:

```
watching your-filename.typ
writing to your-filename.pdf
[18:55:54] compiled successfully in 5.59 ms
```

The watch command monitors the input file and its dependencies for changes and automatically recompiles to PDF.

- Continue to leave Xed's embedded Terminal open.
- When you make an edit in Xed and save your file, the PDF will auto-update.

The screenshot below gives you an idea of how this arrangement looks. (To reduce clutter, you can arrange windows as needed.)



5. Dealing with Errors

If you make a syntax error in your Typst source file, the PDF will not compile. You can then check output in the embedded Terminal (where the Typst CLI is running) for detailed error messages about the problem and how to fix it.

In the first example below, I intentionally omitted the required underscore in the italicized phrase. (It should be “`_je ne sais quoi_`” rather than “`_je ne sais quoi`”). The Terminal output displays an error message “unclosed delimiter” as well as indicating the line number (80) of the error:

```

75
76 = Testing for Errors (Courtesy of `typst watch`)
77
78 #v(0.25cm)
79
80 Omitted closing delimiter for italicized phrases like _je ne sais quoi.
81
82 = Synchronous Preview Methods
83
84 #v(0.25cm)
85
86 + VSodium (or VS Code)
87
Terminal 1
error: unclosed delimiter
  inaugural.typ:80:54
80 Omitted closing delimiter for italicized phrases like _je ne sais quoi.

```

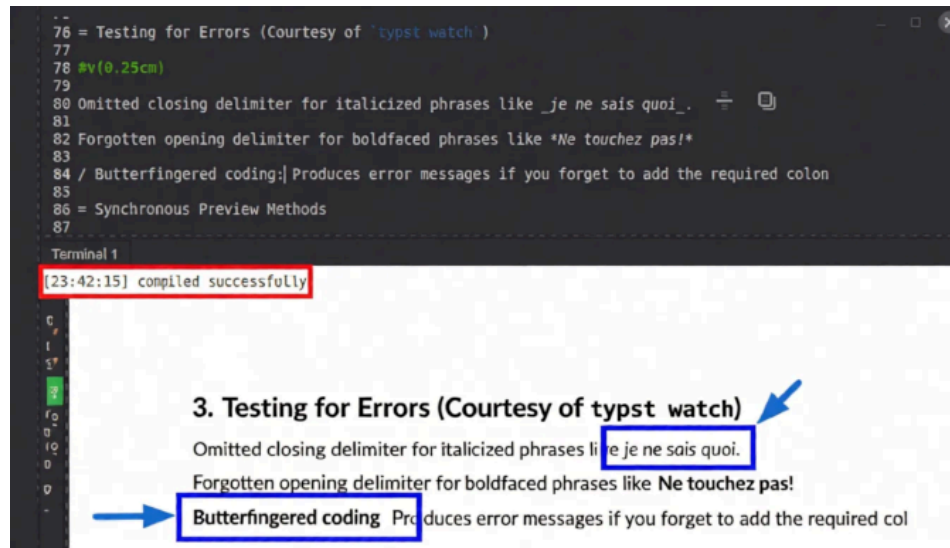
In the second example below, I intentionally omitted the required colon in the term list. (It should be “`/ Butterfingere coding:`” rather than “`/ Butterfingere coding`”). The Terminal output displays an error message “expected colon” as well as indicating the line number (84) of the error:

```

75
76 = Testing for Errors (Courtesy of `typst watch`)
77
78 #v(0.25cm)
79
80 Omitted closing delimiter for italicized phrases like _je ne sais quoi.
81
82 Forgotten opening delimiter for boldfaced phrases like *Ne touchez pas!*
83
84 / Butterfingere coding Produces error messages if you forget to add the required colon
85
86 = Synchronous Preview Methods
87
Terminal 1
error: expected colon
  inaugural.typ:84:87
84 / Butterfingere coding Produces error messages if you forget to add the required colon

```

Once you have corrected these minor mistakes and saved your change, the Typst source file will compile successfully to PDF:



6. Conclusion

Although Xed does not provide a full Typst IDE experience out of the box, it can capably edit your Typst source files. Xed supports limited syntax highlighting and the `xed-terminal` plugin provides an embedded Terminal in Xed's bottom panel, allowing you to access the Typst's CLI `compile` and `incremental watch` commands. The CLI also reports any syntax errors.

This approach has the advantages that no external utilities — such as the `Tinymist` extension — are needed. In addition, people who feel intimidated by `vi(m)` or wish to avoid the CPU-intensive VS Code may appreciate a lightweight, user-friendly text editor such as Xed.

This tutorial was created with Typst. The document's body typeface is Source Serif Pro and the headings use Source Sans Pro.

Good luck and Have fun using Xed with Typst!