How to Install the Typst Compiler
Locally on Salix OS 15.0

By gaucho

In a previous article, I discussed the basic features of Typst, a new typesetting system for the sciences. Although the developers recommend signing up for Typst's web app for the smoothest writing experience, you can also install Typst locally on your computer — without needing to rely on an internet connection. If you are interested in this use case, I will explain how to install the Typst compiler and configure a code editor (VS Code) for use with Typst.

In my experience thus far, installing the Typst compiler is safe and will not bork your Salix installation. I have followed the procedure below on two different computers; I have not experienced any glitches or breakage while I have been using Typst locally.

1. Download the Typst Binary

You should first download the Typst pre-built binary from https://github.com/typst/typst/releases

• Look for Release ver. 0.12.0 (Oct. 18, 2024)

• Download the file: `typst-x86_64-unknown-linux-musl.tar.xz`

This contains the Typst compiler and its CLI, which is everything you need to compile Typst documents locally.

2. Extract the Typst Binary

• Move the tarball from your `Downloads` folder to your `/home` folder.

• Extract/unzip the tarball. It is a modest 39.8 MB in size.

• To simplify the naming convention, I renamed the unpacked directory from "`typst-x86_64-unknown-linux-musl`" to just "`typst`".
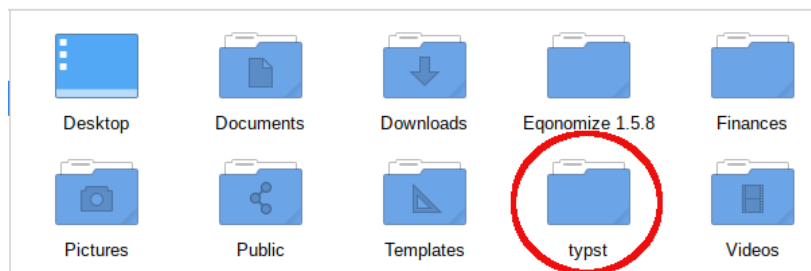
- Within the `typst` directory, use your DE's file manager (Thunar, Caja, etc.) to double-check that the typst binary is set as executable:
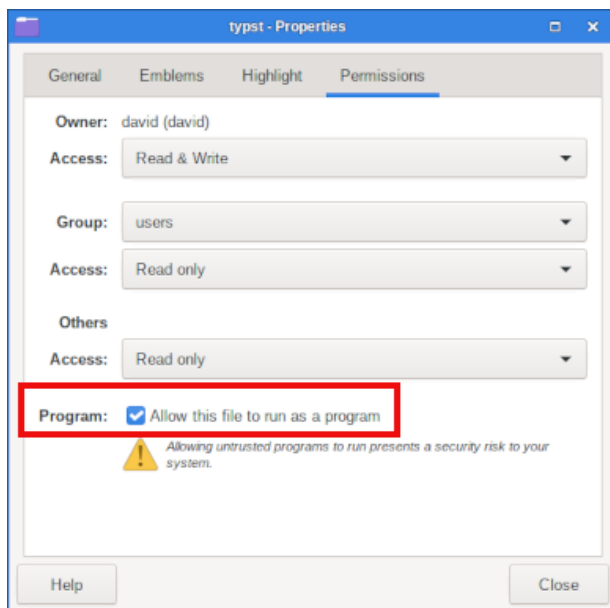
3. Place the Typst Binary in Your $PATH

- You must now place the `typst` directory in your `$PATH`.

A path — or the search path — is the list of directories that will be searched for anything you type on the command line. After editing your path, when you call/invoke Typst, the system will look for [and find] the `typst` binary in your `/home` directory.

Your default `$PATH` in Salix probably looks something like this:

To (permanently) add the `typst` directory to your PATH,

- Open a Terminal and issue the command below in your `/home` directory to edit the `.bashrc` file:

(I chose the nano editor but you may use whatever your preferred text editor is — vim, Emacs, etc.)
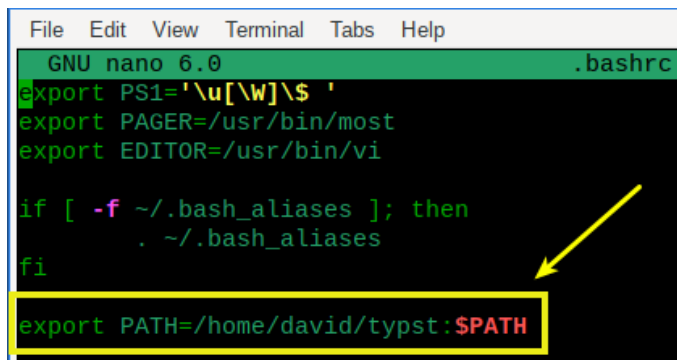
- Type the command: `$ nano .bashrc`

Do not forget to type the initial period (" `.` ") before `bashrc`

- Add the line

```
export PATH=/home/david/typst:$PATH
```

at the bottom of the existing `.bashrc` file:

(Change "david" to whatever the username is on your system.)

- Save the file.

- I logged out of my current session, then logged back in to make sure the change took effect.

- To double-check the results, type: `$ echo $PATH`

Your results should look something like the screenshot below:

If you would like to read some additional background on `$PATH`, the article below is helpful and concise:

https://www.howtogeek.com/658904/how-to-add-a-directory-to-your-path-in-linux/

McKay, Dave. "How to Add a Directory to Your `$PATH` in Linux." *How-To Geek,* 17 Sept. 2023.

4. Choose a Code Editor to Use with Typst

Although Typst can be used exclusively from the command line (as described on the GitHub page), in my experience it is more comfortable to use the app within a code editor.

The two commonly recommended choices are Visual Studio Code (often abbreviated as "VS Code") and VSCodium.

VS Code is a Microsoft product licensed under a non-FLOSS license and it has telemetry reporting enabled (although the telemetry can be disabled). VS Code is a cross-platform code editor that straddles the line between a text editor and an integrated development environment (IDE).

My initial preference was for VSCodium because it is a free/libre binary distribution, licensed under the MIT license (open-source) without Microsoft's proprietary additions. However, after installing the SlackBuild (ver. 1.79.0.23159) [from June 2023], I discovered that the Tinymist Typst extension (discussed in Section 5) did not work correctly. (I suspect this old version of VSCodium is incompatible with the newer extension.) It generated an error message which I could not decipher:
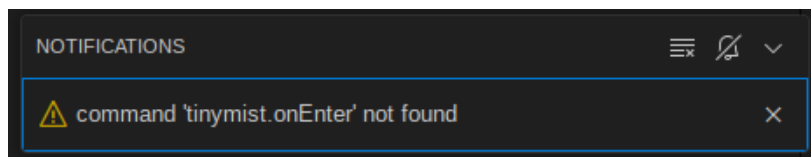
So I installed the SlackBuild of VS Code (ver. 1.95.2) and disabled the telemetry. It has been working fine on my system.

5. Install Recommended Extensions for Typst

Extensions (or plugins) are third-party software packages which extend the code editor's functionality and improve your workflow. You download and install these extensions from the so-called "Extension Marketplace." Due to licensing restrictions, Microsoft's official extensions for VS Code are not available in the VSCodium extension marketplace (which instead uses the open-source registry open-vsx.org).
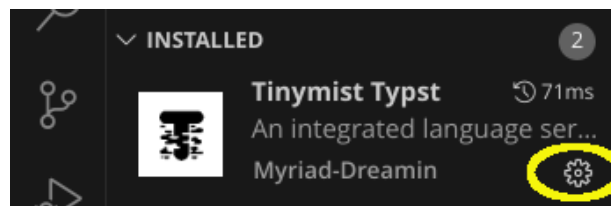
First, find the **Extensions** icon in the Activity Bar of the left-hand panel:

Next, type " typst " in the Search box to filter the available extension offerings:
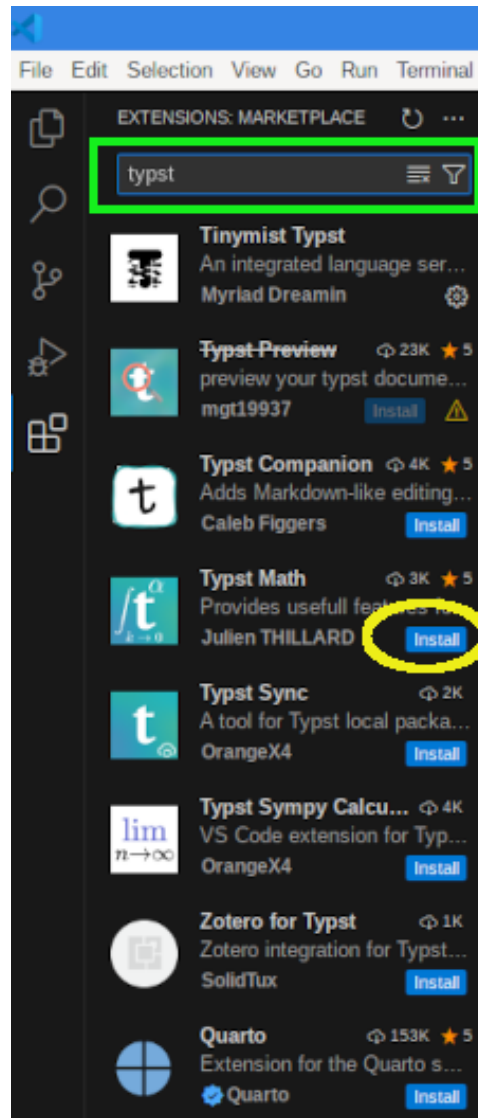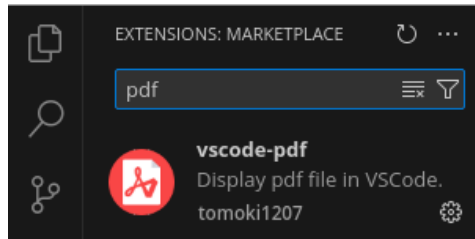
To install an extension, click on the **Install** button beside its description (as shown above). When the installation finishes, the **Install** button will change to the **Manage** button [a gear icon]:
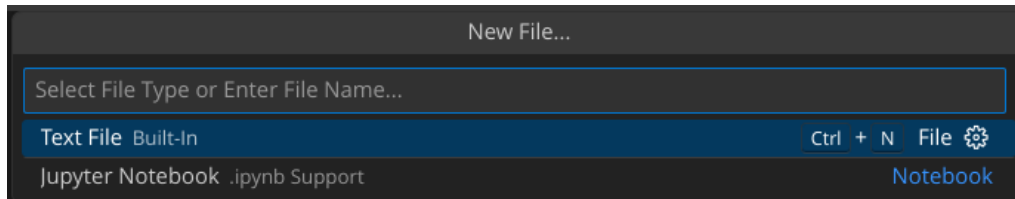

The two extensions I am using with Typst are:

|		
`Tinymist Typst`	(*shown above*)	An integrated language service for Typst
`vscode-pdf`	(*shown below*)	Display PDF in VSCode (or VSCodium)

To start a new Typst file, from the **File** menu > select the **New File** option. A dialog box will open in the editor's toolbar:

Enter the name of your new file (for this example, I called it " `test.typ` "), press Enter/Return and choose the directory (e.g., `Documents`) where you want to save the file.
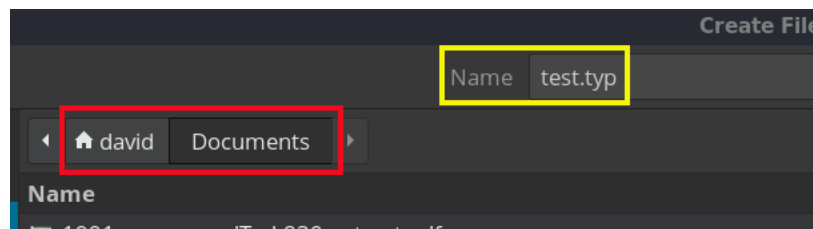
To open a preview of your document, just click on the **Preview** tab in the top portion of the editor pane:
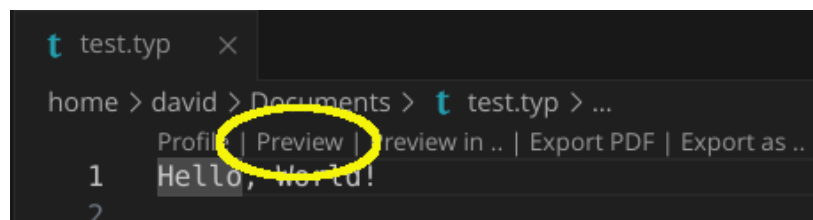
The screenshot below shows a Typst test file, open in the VS Code editor, with a document Preview displaying in the right-hand pane:
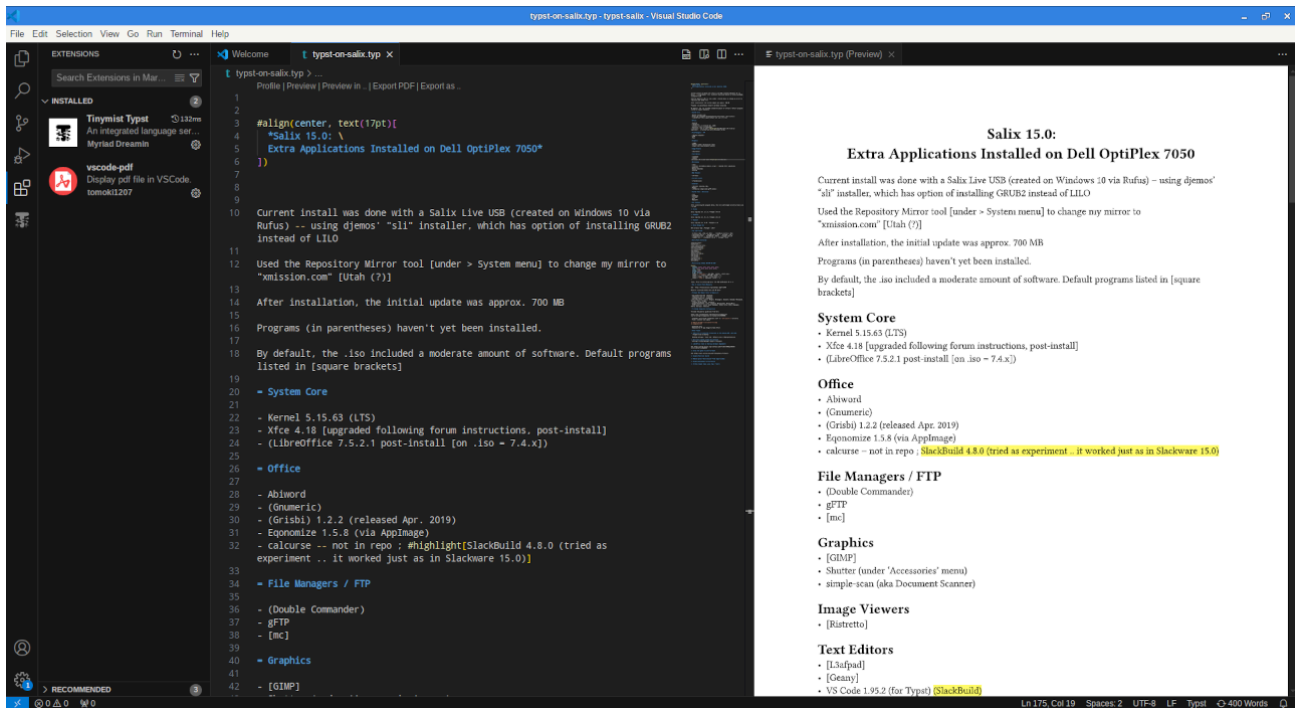
Clicking on the **Export PDF** tab will export your current document as a PDF to your `Documents` folder, and display the generated PDF in a new tab of VSCodium's right-hand pane.
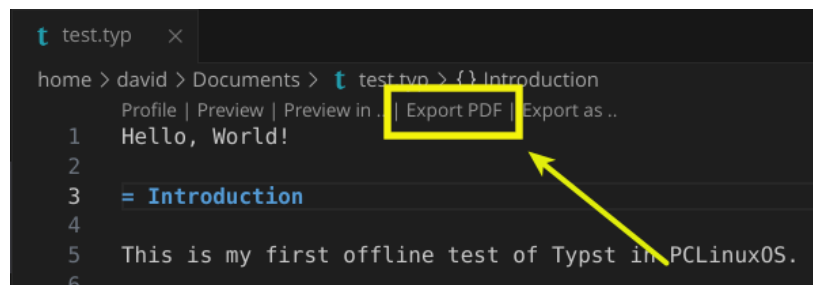

6. Error Messages in VS Code

Error messages display differently in the VS Code editor than in Typst's web app. The Tinymist Typst extension which I installed earlier handles the function of error reporting. In the screenshot below, I was using Typst's "term list" function and intentionally omitted the mandatory colon following the term. (It should be: " / Thylacine: "):
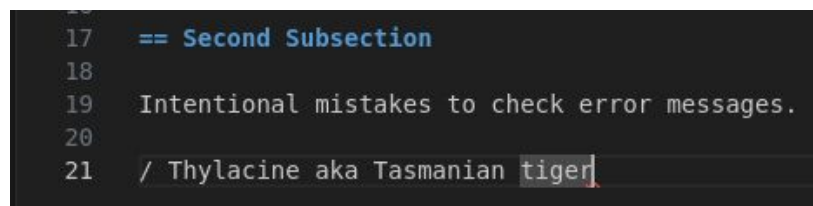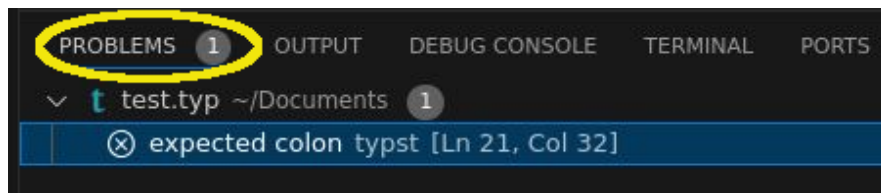

A brief message appears near the editor's toolbar, notifying me that there is a problem in the file:
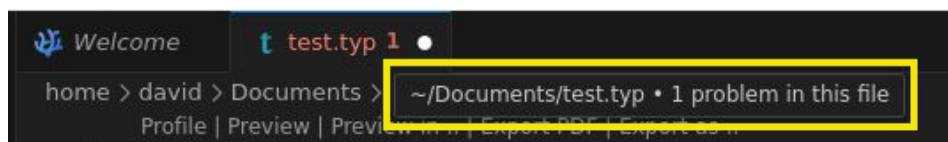
To view Typst's logs, click on the Notification panel in the lower right corner of the editor:

7
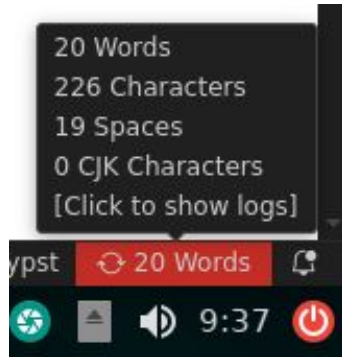
Clicking on the **Problems** tab of the log will display the specific error message as well as its location in the file:
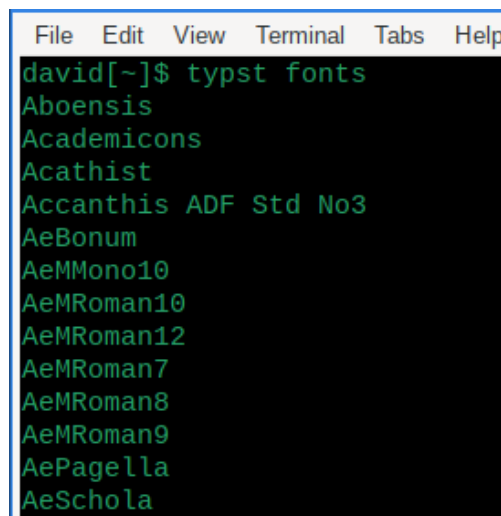

7. Changing the Default Font

By default, Typst uses fonts embedded in the CLI: Libertinus Serif, New Computer Modern, New Computer Modern Math, and DejaVu Sans Mono.

However, the Typst compiler can also use system-wide fonts that you have installed. To display a list of available fonts, open a Terminal and type the command:
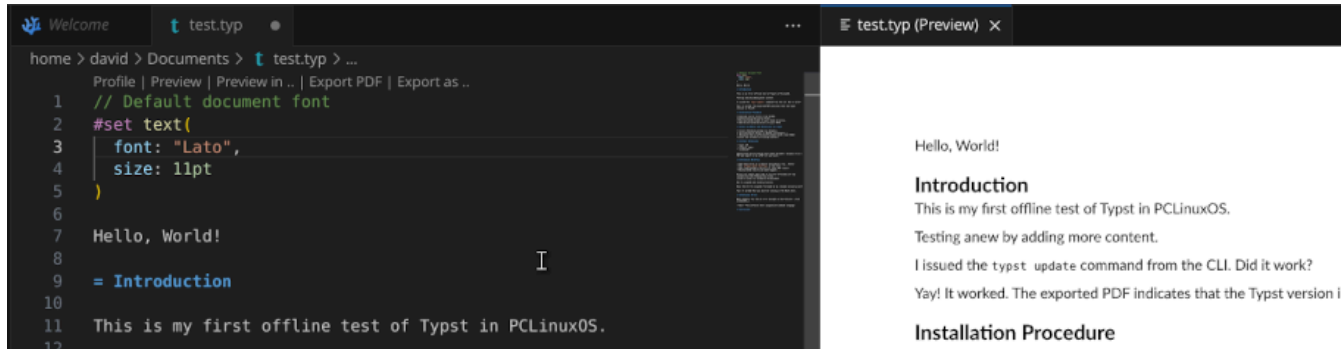
```
$ typst fonts
```


After you have decided which typeface to use, add this code snippet at the beginning of your document:

```
// Default document font
#set text(
  font: "Lato",
  size: 11pt
)
```

Change the typeface and font size to suit your taste. (Lato, the Polish word for 'summer', is an attractive typeface [shown below] which was designed by Łukasz Dziedzic.)

8. Adding Images to Your Typst Project

It was not immediately obvious where to store image files that I wanted to add to my project, so I experimented with a few different methods.

I created my example file ("`test.typ`" [*pictured above*]) in my `Documents` folder. VS Code expects to find both your Typst file and any image files in the same folder.

If your Typst file contains numerous images, you will find that your `Documents` folder quickly becomes cluttered — something you probably want to avoid. I learned that it is possible to create subfolders within the `Documents` folder, which store your image files.

In the screenshot below, you can see that I have one image file (circled in blue) in my `Documents` folder, as well as three image test subfolders (circled in red, green and purple):
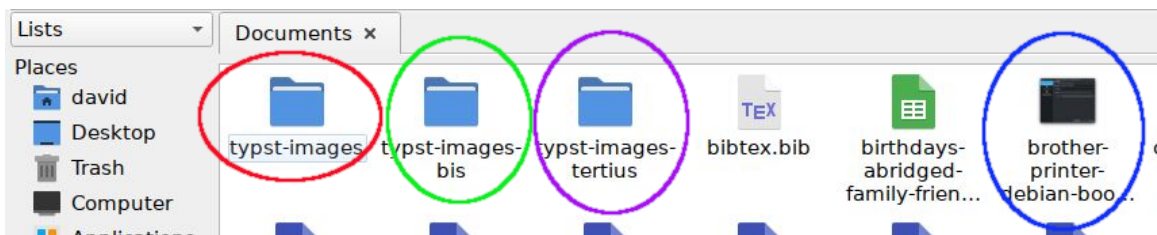
When you type the code for Typst's figure function to insert a figure, the editor will help autofill the image file name(s). If the image is in the same folder as your Typst document, just use the image file name. However, if your image is located in a subfolder, you must include the path, e.g., `/typst-images/Tux.png`, etc.

The screenshot below illustrates that VS Code will detect image files in the `Documents` folder, in addition to the `typst-images`, `typst-images-bis`, and `typst-images-tertius` subfolders. The code completion autofill should begin working when you type the " **image("** " portion of the code snippet:
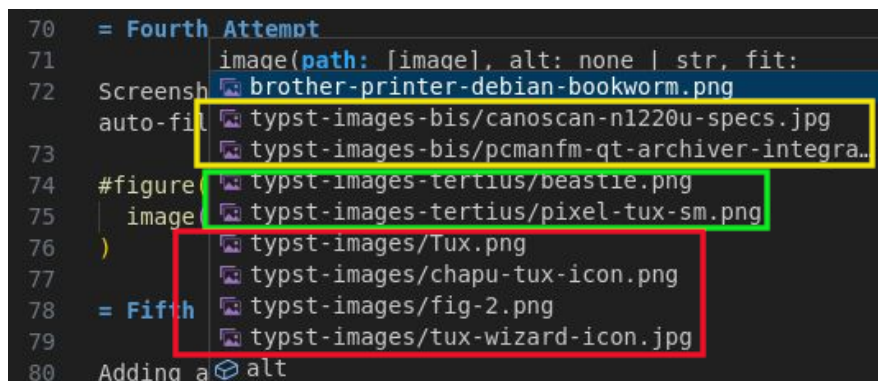
9. A Better Organized Workflow in VS Code

After skimming through the Visual Studio Code documentation and watching a couple of video tutorials, I realized that the recommended approach of using a folder (also known as a workspace) results in a cleaner workflow.

Follow the procedure below to benefit from a more organized workflow:

• Create a new (empty) project folder in your `Documents` folder. (I called this example " `typst-project-2` ")

• Start the VS Code editor.

• From the **File** menu > choose **Open Folder**, then select the "`typst-project-2`" folder.

• The "Workspace Trust" dialog will appear:

This feature lets you decide whether code in your project folder can be executed by VS Code and extensions without your approval.
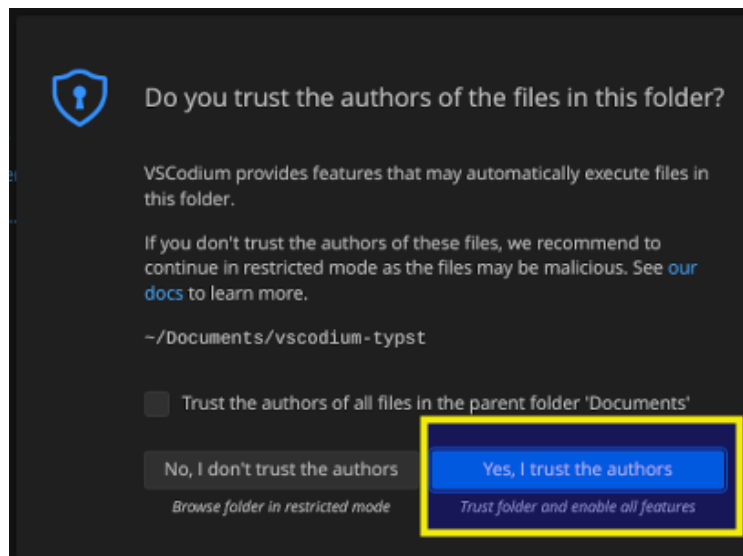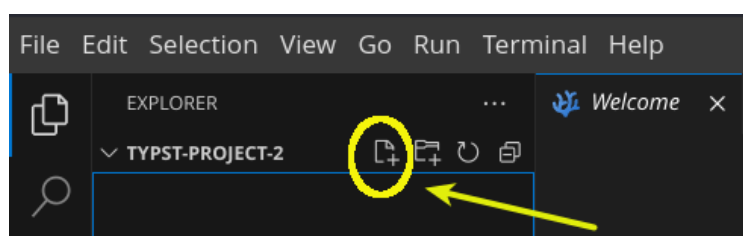

• Click on the blue "**Yes, I trust the authors**" button.

Your new folder (aka workspace) will display in the **Explorer** tab [the stacked sheets of paper icon] of the Activity Bar in the editor. The Explorer view lets you view and manage the files and folders in your workspace:
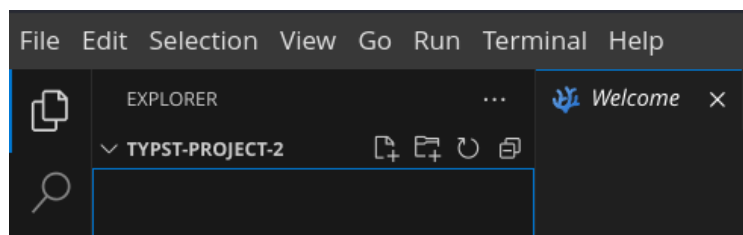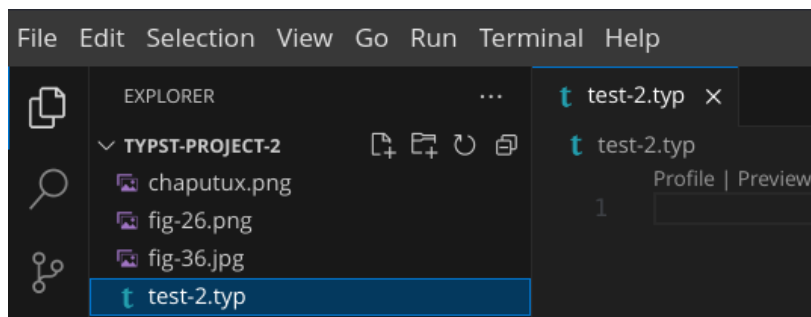

• Click on the **New File** icon:

- In the box, type the name of the Typst file which will be created in your workspace. (I called mine "`test-2.typ`"):
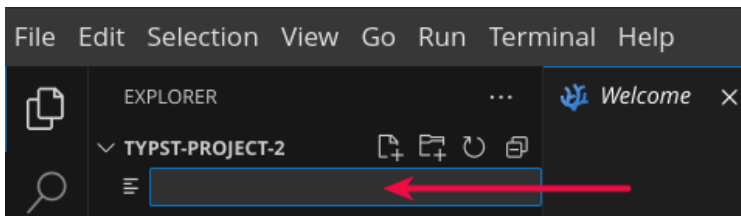
- Copy any image files you wish to use into your "`typst-project-2`" folder.

- The image files and the Typst file will be listed under the Explorer tab:

- Add the image files to your Typst document as you normally would, via the " **#figure** " command.

10. Stay Up to Date with Future Releases

Once you have installed Typst locally using this procedure, you can stay up to date with future releases by opening a Terminal and issuing the command:

```
$ typst update
```

I tried this recently when the Typst developers released new version 0.12.0 on Oct. 18, 2024. (I had previously installed ver. 0.11.1.) The update process worked smoothly.

I hope these instructions will allow you to successfully install Typst on your local Salix system and use it productively with VS Code.

11. Additional Resources

The resources mentioned below focus on Visual Studio Code. (They are also helpful explaining the basics of VSCodium because VSCodium maintains feature parity with VS Code.)

https://code.visualstudio.com/docs/
Visual Studio Code Documentation (official)

The tutorial on getting started with VS Code provides a succinct overview.

https://www.youtube.com/watch?v=ORrELERGIHs

Ruscica, Tim. *Tech with Tim*. "VSCode Tutorial for Beginners: Getting Started with VSCode." *YouTube,* 13 Feb. 2021. (22 min., 56 sec.)

https://www.youtube.com/watch?v=yjeHLSrhPao

Coding with Estefanía. "Learn Visual Studio Code: Course for Beginners." *YouTube,* 20 Jan. 2022. (43 min., 15 sec.)

This tutorial was created with Typst. The document's body typeface is Source Serif Pro and the headings use Source Sans Pro.
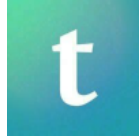
Good luck and Have fun using Typst!