Getting Started with PythonAnywhere

Congratulations on choosing PythonAnywhere as your cloud-based IDE. Feel free to skim through their homepage or skip immediately to the getting-started instructions below.

Sign up for an account

1. On the PythonAnywhere homepage, click the “Pricing & signup” menu item in the top right-hand corner.
2. Click the “Create a Beginner account” button.
3. Complete the form asking for your email address and requesting you to choose a username and password. Click the “I agree” box for the account’s terms and conditions, and finally click the “Register” button.

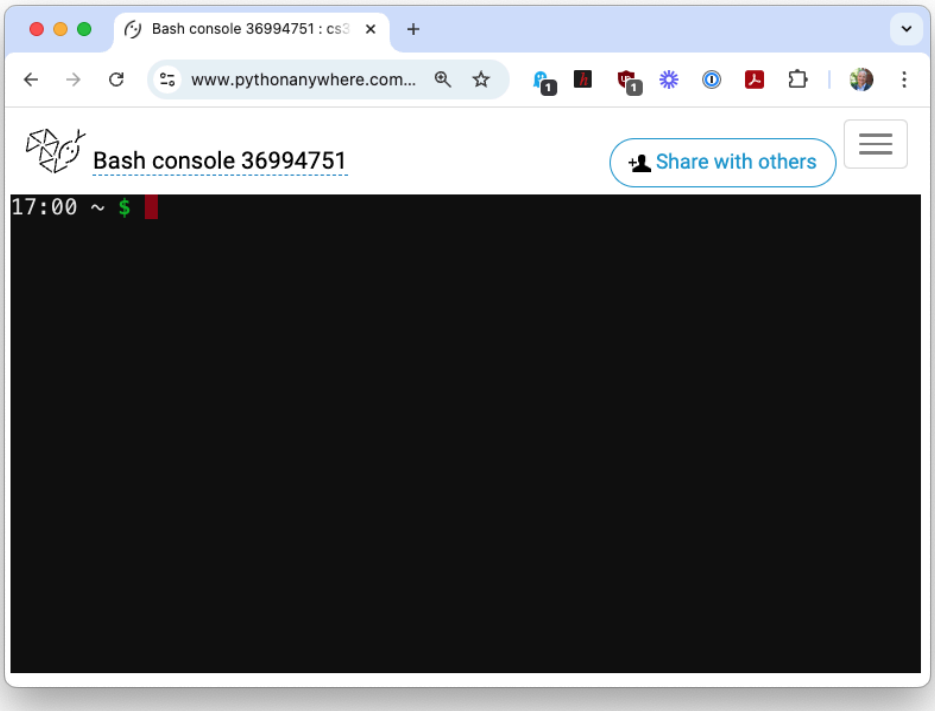
Log in to your account

Once you have a PythonAnywhere account, log in and click into your **Dashboard**, which is one of the navigation options below the main navigation bar at the top of the web page. This dashboard page shows you things like how much storage you’ve used, any consoles you have running, and any files you’ve recently opened. Since you just started, you should have none.

Let’s get you some files so that you can do something interesting with your new IDE!

Loading the files for Chapter 1

1. Under “New console” on your Dashboard page, click the button labeled “$ Bash” and wait for the “Bash console” to finish initializing. When it is done, it will look like this:

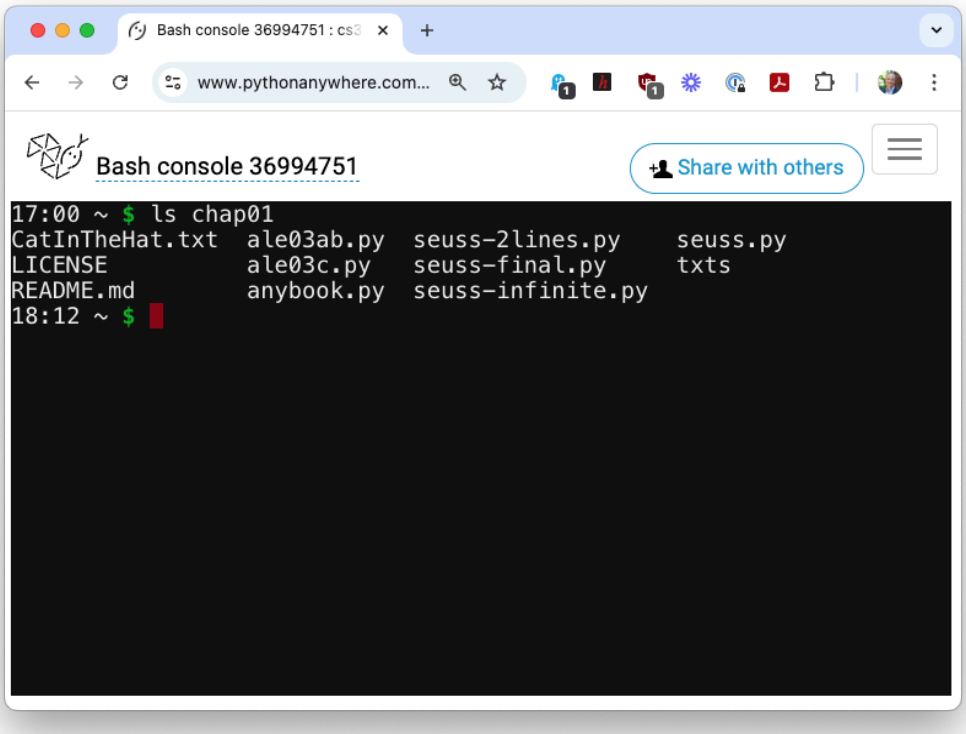


2. At the prompt (i.e., the green dollar sign), type the following and hit return:

```
git clone https://github.com/pswp-book/chap01.git
```

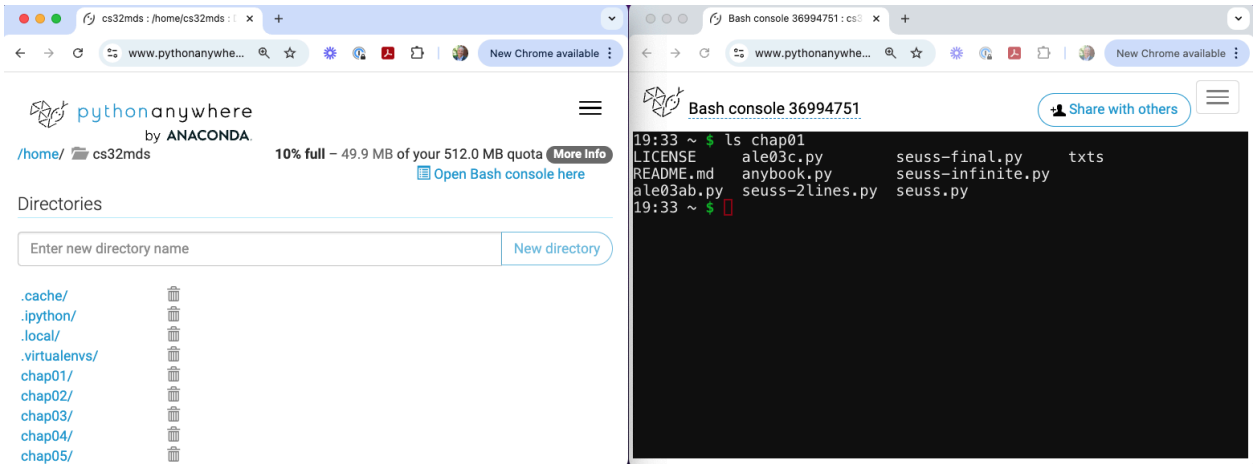
3. This command downloads all the files you’ll need for Chapter 1 from the book’s Github repository, and it puts them in a directory (folder) called `chap01` in your cloud-based PythonAnywhere account. You can see them by typing the following at the command prompt (after the green `$`) and hitting return:

```
ls chap01
```



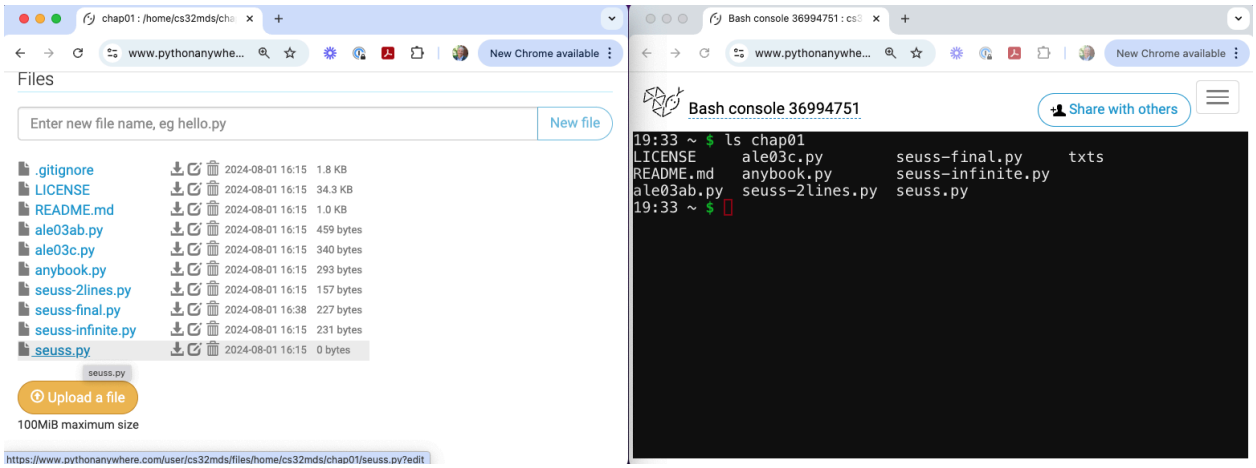
You'll learn more about shell commands like `ls` in Chapter 13.

4. To open the IDE editor on one of these files, click the hamburger menu icon in the upper right-hand corner and select "Files". You might right-click this option and specify that you'd like this page opened in a new browser window. In this way, you can see the console and the editor at the same time.

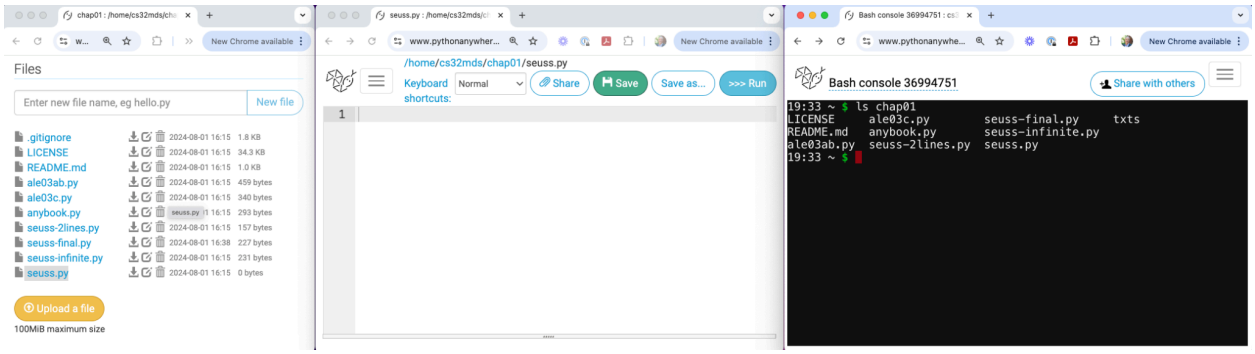


5. I have more folders and files in my PythonAnywhere account than you have, but you should see `chap01` listed under Directories. Click that name and you'll open that folder.

6. Scroll down until you see a file named `seuss.py` and click that name.



7. Clicking a filename in the file browser window will open that file in the IDE's editor. I've clicked `seuss.py` and instructed my browser to open it in a new window.



The middle window is PythonAnywhere's editor in which you can type Python statements. Start reading Chapter 1 to learn what to type there.

Mapping PythonAnywhere's IDE to the book's generic IDE

The book's generic IDE highlights two bars, three panels, and one button all in one IDE window. PythonAnywhere asks that you open the three panels in separate browser windows, which I laid out to look like the book's generic IDE.

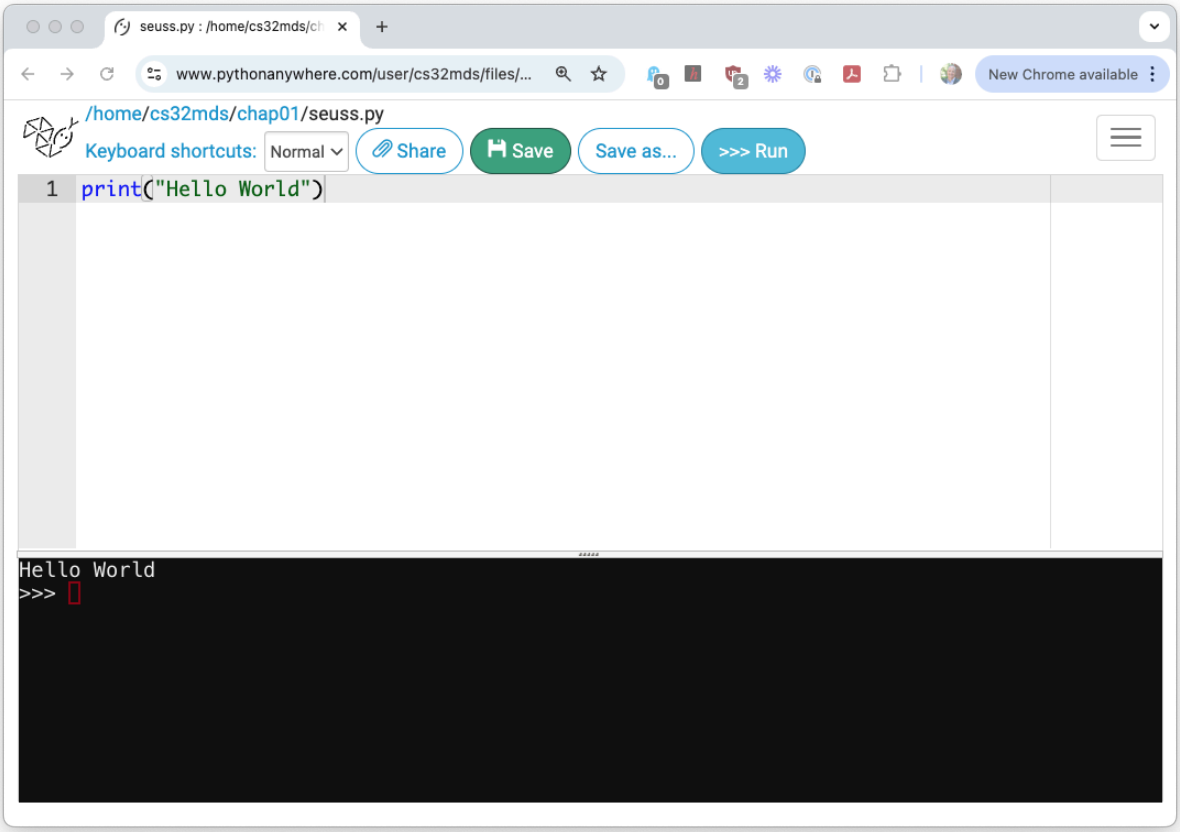
You can see the Run button in the editor window. In PythonAnywhere, there isn't a gear icon that allows you to change a range of IDE configuration settings. The most you can do is to change the "Keyboard shortcuts" defined by using the pulldown menu at the top of the editor window.

Running your first Python program

In the code editor window, type:

```
print("Hello World")
```

Click the ">>> Run" button in the editor window. The result should look something like the following:

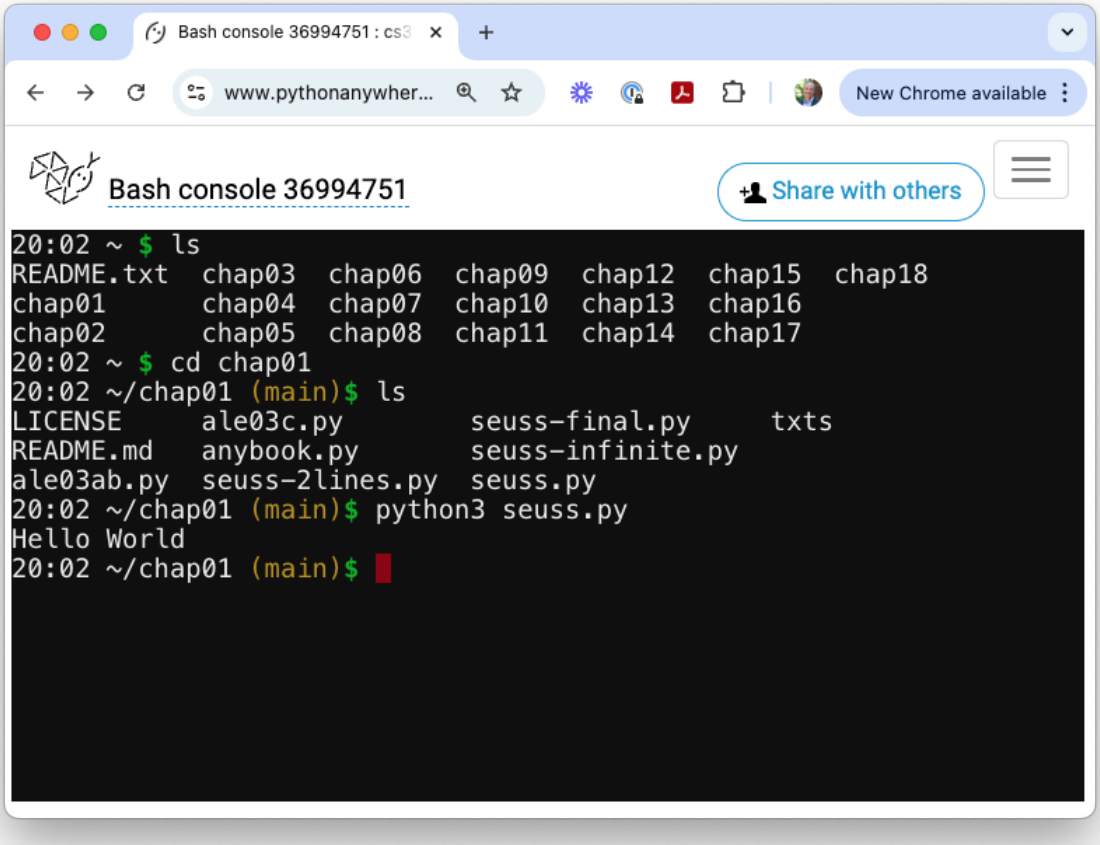


Congratulations on writing and running your first Python program!

Running a Python program in the console

We won't use the Run button much throughout the book, but instead run our programs from the shell command prompt. This is a short description of how to do that.

1. Make sure you **click the Save button** in your editor window to save your edited program in your PythonAnywhere filesystem.
2. **Click in the Console window** so that you can type at the command prompt.
3. We left the Console sitting in the folder that contains the `chap01` folder. Type `cd chap01` so that the shell's focus is in the folder where the file `seuss.py` sits.



Notice that the text before the $ changed to let you know where the shell's focus is.

4. Type `python3 seuss.py`, which invokes the Python interpreter on our program just as the Run button in the editor did. If you want to run a Python program named something else, change `seuss.py` to that filename.
5. To change the shell's focus back to the folder containing chap01, type the following and hit enter:

   ```
   cd ..
   ```

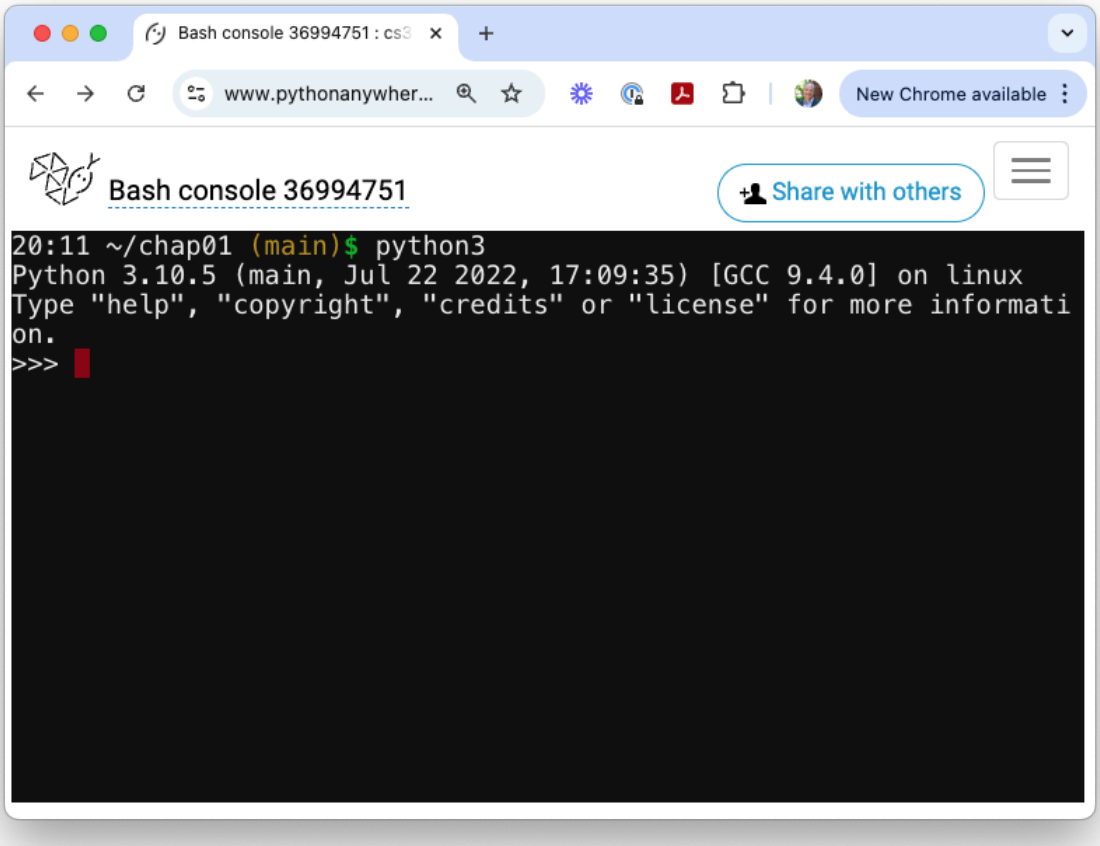
   Go ahead and try moving into and out of the `chap01` folder using the shell.

Starting the interactive Python interpreter

Instead of writing a full Python script, we sometimes want to try out a few Python statements. This is best done in what's called the *interactive Python interpreter*. You can tell you're talking to

the interactive Python interpreter when the command line prompt looks like three greater-than symbols (`>>>`), which is what we saw in the output window when we clicked the "`>>> Run`" button in the PythonAnywhere's editor window.

More generally, we start the interactive Python interpreter in the Console by typing `python3` at the command prompt. Yes, it's like running a Python program but not feeding a program file to the Python interpreter.



Chapter 1 contains examples of how the interactive Python interpreter will be useful to us. For now, just know that you type *Ctrl+D* (i.e., simultaneously press your keyboard's Control and D keys) to exit the interactive Python interpreter.

Try starting and exiting the interactive Python interpreter now.

Loading files from another chapter

Each book chapter comes with files that you can grab using the `git clone` command we used earlier. You simply need to put the shell's focus back to your PythonAnywhere project's main

folder, which in my case was when you saw a `~` before the `$`, and then change the chapter in the `git clone` command's URL (e.g., the Chapter 2 files are available at https://github.com/pswp-book/chap02.git).

Uploading a file

At some point, you'll want to upload a file from your personal computer and have it appear in a directory in PythonAnywhere's file browser. In these cases, you want to use the "Upload a file" button at the bottom of the file browser window I showed earlier.

Installing a package

In Chapter 4, you'll start using packages that are not installed by default in all Python installations. In particular, this chapter will use the `requests` package, which is installed in PythonAnywhere's IDE, but if it wasn't you'd run the Python package manager `pip` in the Console to install the package you need. For example, typing `pip install requests` would install the `requests` package and make it available for you to `import` in your Python programs.

[Version 20260228]