Chapter 3
Setting up the Python Environment

3.1 Introduction

In this chapter we will check to see if you have Python installed on your computer. If you do not have Python installed then we will step through the process of installing Python. This is necessary because when you run a Python program it looks for the `python` interpreter that is used to execute your program or script. Without the `python` interpreter installed on your machine Python programs are just text files!

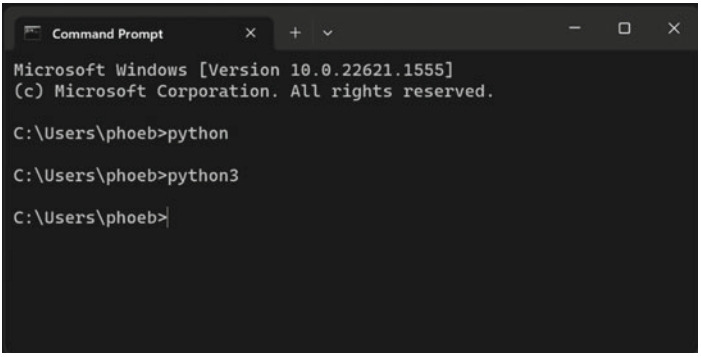
3.2 Check to See if Python is Installed

The first thing you should do is to see if Python 3 is already installed on your computer. First check to see that you don't have Python installed. If it is you don't need to do anything unless it is a very old version of Python 3 such as 3.1 or 3.2.

On a Windows 11 machine you can check the version installed by opening a Command Prompt window (this can be done by searching for Cmd in the 'Search' box in Windows).

Once the Command window is open try typing in `python`. This is illustrated below:

J. Hunt, *A Beginners Guide to Python 3 Programming*, Undergraduate Topics
in Computer Science, https://doi.org/10.1007/978-3-031-35122-8_3

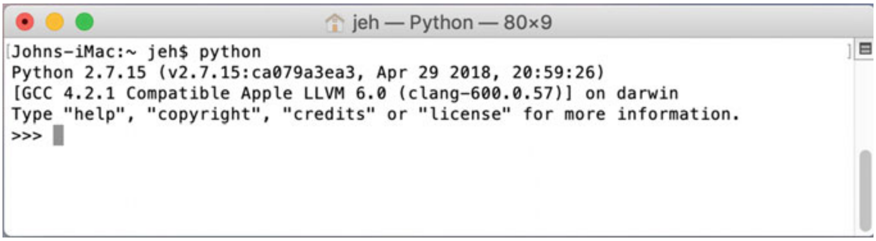


```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\phoeb>python

C:\Users\phoeb>python3

C:\Users\phoeb>
```

You may be prompted to look in the Microsoft store, you can ignore that and close the window.

Note in the above where we have tried both `python` and `python3` in case the latest version has been installed using that name. This is not an uncommon thing to do as on some machines you may have both Python 2 and Python 3 available.

On a system such as a Mac you can use the Terminal and do the same thing. You will probably find that at least Python (2) is pre-installed for you. For example, if you type in `python` on a Mac you will get something like this:



```
Johns-iMac:~ jeh$ python
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 29 2018, 20:59:26)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

This indicates that the above user has version 2.7.15 installed (note you may have another 2.x version installed).

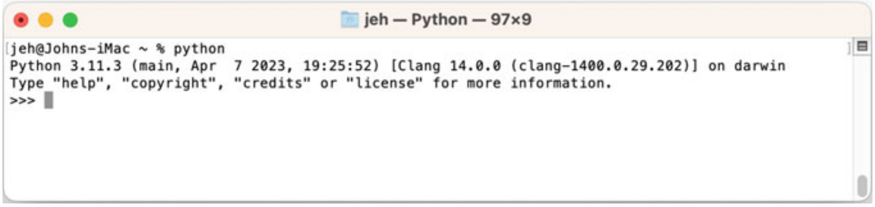
However, be careful if you find that you have Python 2 installed on your machine; this book is focused solely on Python 3.

If you have started a Python interpreter then

- Use `quit()` or `exit()` to exit the Python interpreter; `exit()` is an alias for `quit()` and is provided to make Python easier to use.

If Python 3 was not available, then the following steps will help you to install it.

If the correct version of Python is already available on your computer then you can skip to the next chapter. An example of what might happen if you type in Python and Python 3 is installed on a Mac is given below:



```
jeh@Johns-iMac ~ % python
Python 3.11.3 (main, Apr  7 2023, 19:25:52) [Clang 14.0.0 (clang-1400.0.29.202)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

Here you can see that this machine has Python 3.11.3 installed.

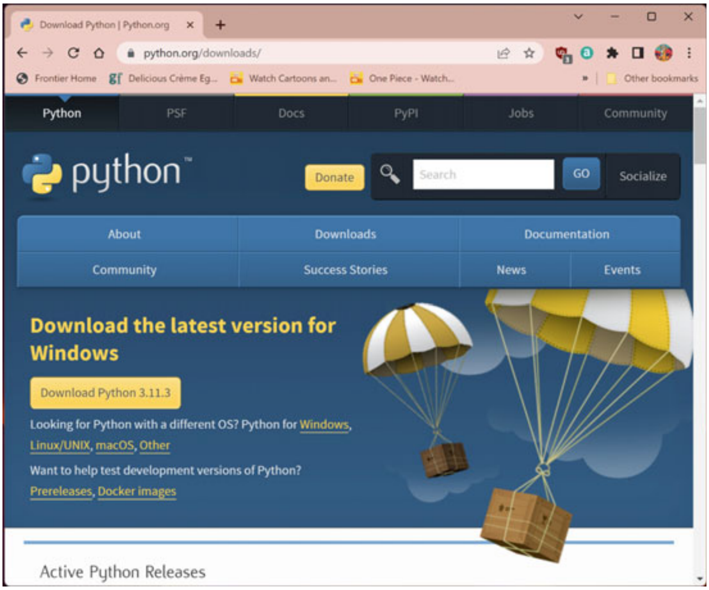
3.3 Installing Python on a Windows PC

3.3.1 Step 1: Downloading Python

Python is available for a wide range of platforms from Windows, to Mac OS and Linux; you will need to ensure that you download the version for your operating system.

Python can be downloaded from the main Python web site which can be found at http://www.python.org/.

As you will see the 'Downloads' link is the second from the left on the large menu bar below the search field. Click on this and you will be taken to the download page; the current Python 3 version at the time of writing is Python 3.11 which is the version we will download (if you find that the version is 3.12 or later then that is fine as well). Click on the 'Download Python 3.x.x' link. For example, for a Windows 11 machine, using 3.11.3, you will see:



Download the latest version for Windows

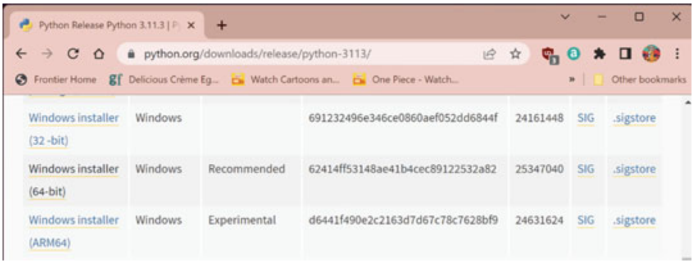
Download Python 3.11.3

Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

Want to help test development versions of Python? Prereleases, Docker images

Active Python Releases

Click on the 'Download' button—this may take you to a page with several different options on it—look for the 'Recommended' option and down load that installer.

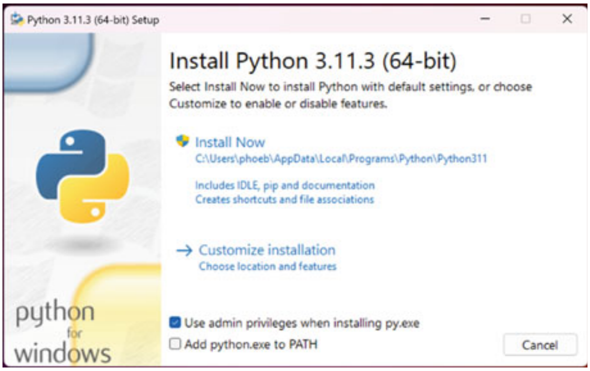


Windows installer (32 -bit)	Windows		691232496e346ce0860aef052dd6844f	24161448	SIG	.sigstore
Windows installer (64-bit)	Windows	Recommended	62414ff53148ae41b4cec89122532a82	25347040	SIG	.sigstore
Windows installer (ARM64)	Windows	Experimental	d6441f490e2c2163d7d67c78c7628bf9	24631624	SIG	.sigstore

Even if a newer version of Python is available (which is likely as the version is updated frequently) the steps should be fundamentally the same.

3.3.2 Step 2: Running the Installer

Once you have download the install *open* it. You will now be prompted for the location to install Python, for example:



Install Python 3.11.3 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\phoeb\AppData\Local\Programs\Python\Python311

Includes IDLE, pip and documentation
Creates shortcuts and file associations

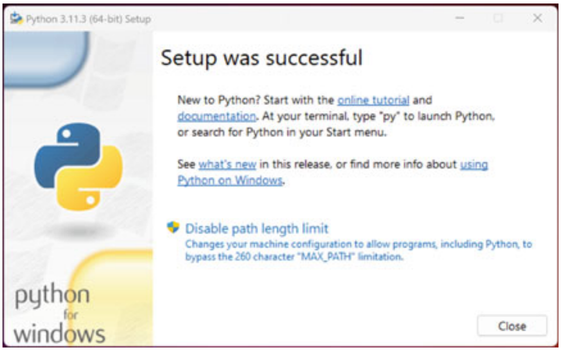
→ Customize installation
Choose location and features

☑ Use admin privileges when installing py.exe
□ Add python.exe to PATH

Note that it is easiest if you click on the ‘Add python.exe to PATH’ option as this will make it available from the command line. If you don’t do this then don’t worry, we can add Python to the `PATH` later on (the `PATH` environment variable is used by Windows to find the location of programs such as the Python interpreter).

Next select the ‘Install Now’ option and follow the installation steps.

If everything went as expected you should now see a confirmatory dialog such as:



Setup was successful

New to Python? Start with the online tutorial and documentation. At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See what's new in this release, or find more info about using Python on Windows.

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

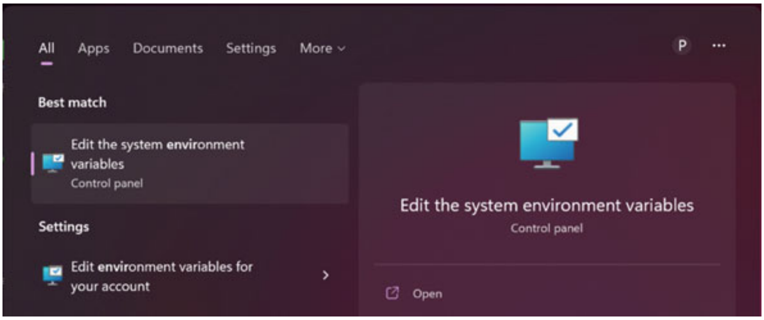
If you are on Windows now close any Command Prompt windows that you have open (the `PATH` variable is not updated for existing Commands). This can be done by typing in ‘exit’ or closing the window.

3.3.3 Step 3: Set up the PATH (Optional)

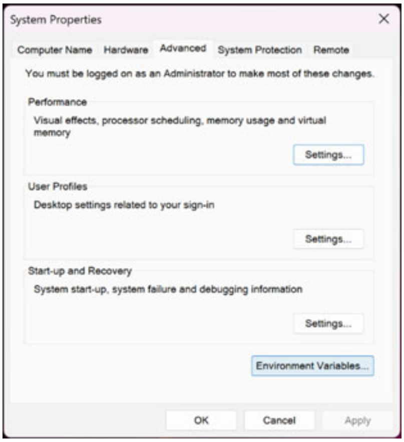
If you did not select ‘Add python.exe to PATH’ at the start of the installation you will now need to set up the `PATH` environment variable. If you did then skip onto the next step.

You can set the `PATH` environment variable using the system environment variables editor.

The easiest way to find this is to type ‘`envir`’ into the Windows search box, it will then list all applications which match this pattern including the ‘Edit the system environment variables’ editor. If you are logged into your machine as a user with ‘admin’ rights use the first one listed, if you are logged in as a user without admin rights select the ‘Edit environment variables for your account’ option.


All
Apps
Documents
Settings
More
P
Best match
Edit the system environment variables
Control panel
Settings
Edit environment variables for your account
Edit the system environment variables
Control panel
Open

One the resulting dialog select the ‘Environment Variables …’ button at the bottom of the page:



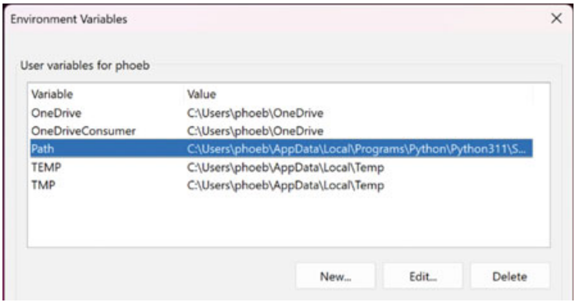
You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage and virtual memory

Desktop settings related to your sign-in

System start-up, system failure and debugging information

One the next dialog select the `PATH` environment variable and select Edit:



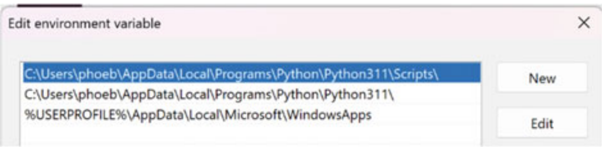
Variable	Value
OneDrive	C:\Users\phoeb\OneDrive
OneDriveConsumer	C:\Users\phoeb\OneDrive
Path	C:\Users\phoeb\AppData\Local\Programs\Python\Python311\S...
TEMP	C:\Users\phoeb\AppData\Local\Temp
TMP	C:\Users\phoeb\AppData\Local\Temp

Now add the location in which you installed Python, by default this will be something like.

```
C:\Users\<username>\AppData\Local\Programs\Python\Python311
C:\Users\<username>\AppData\Local\Programs\Python\Python311\Scripts
```

Note Python311 should be replaced by the version of Python you are installing if it is different.

The end result should look something like:

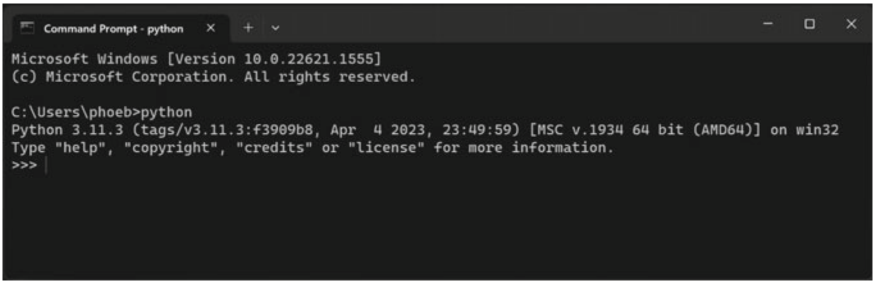


```
C:\Users\phoeb\AppData\Local\Programs\Python\Python311\Scripts\
C:\Users\phoeb\AppData\Local\Programs\Python\Python311\
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
```

Now click on OK until all the windows are closed.

3.3.4 Step 4: Verify the Installation

Next open a new Command Prompt window and type is `python` as shown below:



```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\phoeb>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Congratulations you have installed Python and run the `python` interpreter!

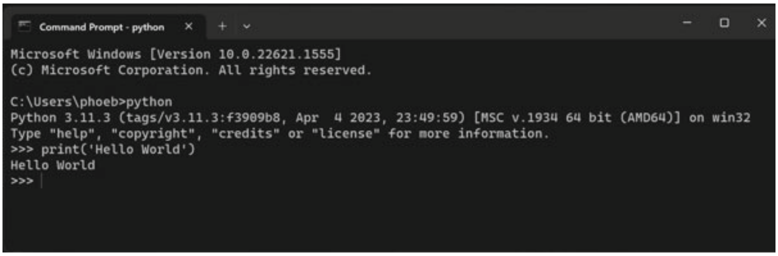
3.3.5 Step 5: Run Some Python

Now at the ‘ >>> ’ prompt type in

```
print('Hello World')
```

Be careful to make sure you use all lowercase letters for the ‘print’ function; Python is very case sensitive which means that as far as Python is concerned `print('Hello World')` and `Print('Hello World')` are completely different things.

Also make sure that you have single quotes around the *Hello World* text this makes it a string. If you get this right then on Windows you should see:

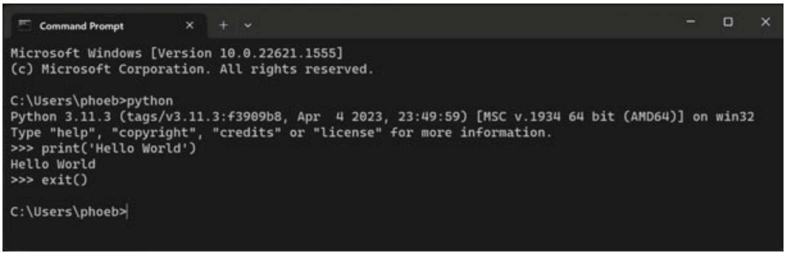


```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\phoeb>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World')
Hello World
>>>
```

You have now run your first Python program. In this case your program printed out the message ‘`Hello World`’ (a traditional first program in most programming languages).

3.3.6 Step 6: Exit the Python Interpreter

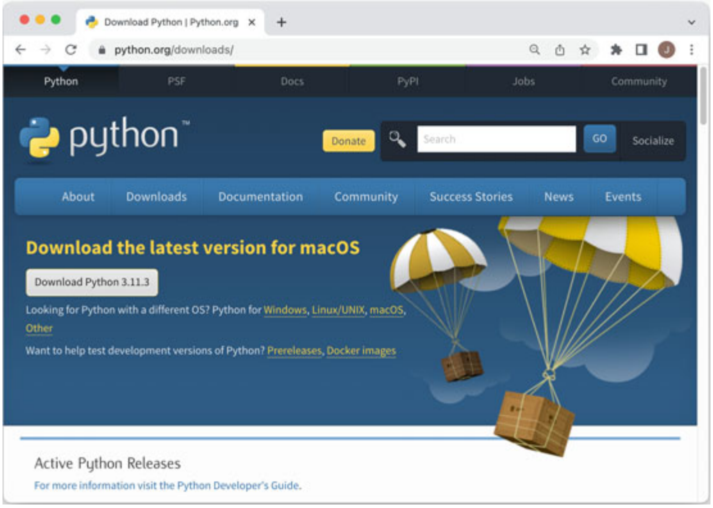
To exit the Python interpreter use exit() or Ctrl-Z plus Return.



```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\phoeb>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World')
Hello World
>>> exit()

C:\Users\phoeb>
```

3.4 Setting up on a Mac

Installing Python on a Mac in similar to installing it on a Windows machine in that you can download the Python installer for the Apple Mac from the Python Software Foundation web site (https://www.python.org). Of course this time you need to make sure you select the macOS version of the download as shown below:



Download the latest version for macOS

Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

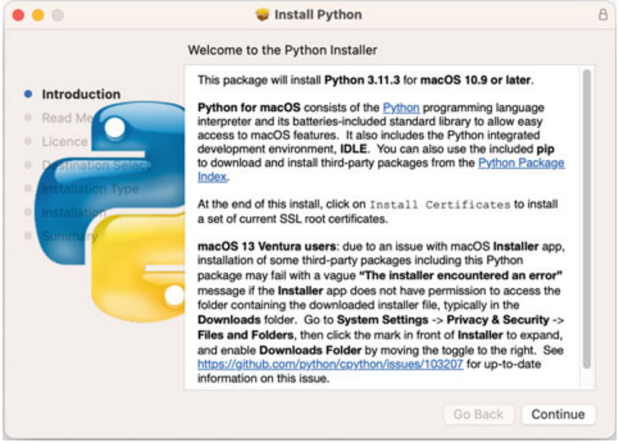
Want to help test development versions of Python? Prereleases, Docker images

Active Python Releases

For more information visit the Python Developer's Guide.

From here you can select the macOS 64-bit installer (make sure you select the appropriate one for your version of the operating system). This will download an Apple package that can be installed (it will have a name similar to `python-3.11.3-macos11.pkg` although the version number of Python and the macOS operating system may be different). You will need to run this installer.

When you run this installer the Python Installer wizard dialog will open as shown below:



Welcome to the Python Installer

This package will install **Python 3.11.3** for **macOS 10.9 or later**.

Python for macOS consists of the Python programming language interpreter and its batteries-included standard library to allow easy access to macOS features. It also includes the Python integrated development environment, **IDLE**. You can also use the included **pip** to download and install third-party packages from the Python Package Index.

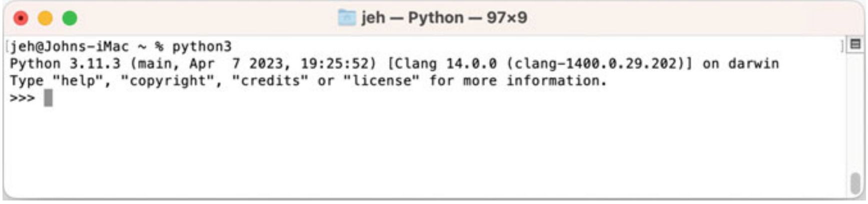
At the end of this install, click on `Install Certificates` to install a set of current SSL root certificates.

macOS 13 Ventura users: due to an issue with macOS **Installer** app, installation of some third-party packages including this Python package may fail with a vague **"The installer encountered an error"** message if the **Installer** app does not have permission to access the folder containing the downloaded installer file, typically in the **Downloads** folder. Go to **System Settings** -> **Privacy & Security** -> **Files and Folders**, then click the mark in front of **Installer** to expand, and enable **Downloads Folder** by moving the toggle to the right. See https://github.com/python/cpython/issues/103207 for up-to-date information on this issue.

Step through the dialogs presented to you by the installer wizard accepting each option until the installation starts. Once the installation has completed you will be presented with a summary screen confirming the installation was successful. You can now close the installer.

This will have created a new folder in your Applications folder for Python 3.11.

Note that on a Mac which already has Python 2 installed (which is installed by default), Python 3 can be installed along side it and can be accessible via the `python3` command (as shown below). You should confirm that Python has been installed successfully by opening a Terminal window and entering the Python 3 REPL:



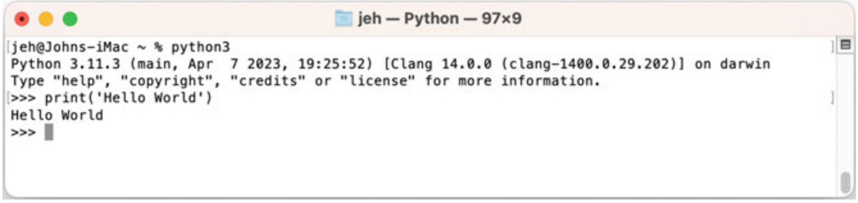
```
jeh@Johns-iMac ~ % python3
Python 3.11.3 (main, Apr  7 2023, 19:25:52) [Clang 14.0.0 (clang-1400.0.29.202)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

Now at the '>>>' prompt type in

```
print('Hello World')
```

Be careful to make sure you use all lowercase letters for the 'print' function; as on a Windows machine Python is very case sensitive which means that as far as Python is concerned `print('Hello World')` and `Print('Hello World')` are completely different things.

The result should be as shown below:



```
jeh@Johns-iMac ~ % python3
Python 3.11.3 (main, Apr  7 2023, 19:25:52) [Clang 14.0.0 (clang-1400.0.29.202)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World')
Hello World
>>> 
```

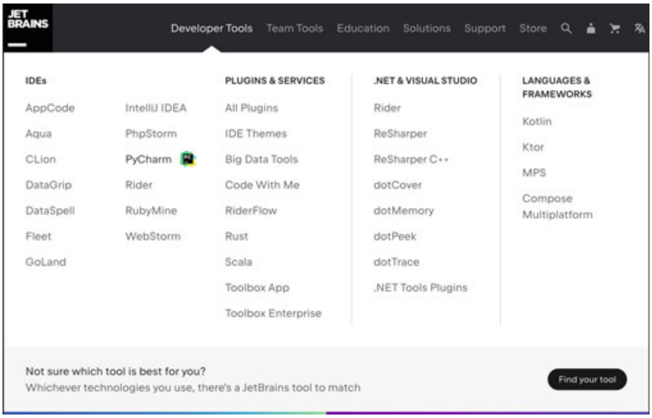
You can now exit the REPL using the `exit()` or `quit()`.

3.5 Setting up an IDE

You can use whatever editor you want along with this book, and most chapters are IDE independent. However, if you wish to use the PyCharm editor to make life easier then this section outlines downloading, installing and getting started with this editor. Almost all other chapters are IDE independent apart from one section in the 'Python Virtual Environments' chapter.

3.5.1 Downloading the PyCharm IDE

PyCharm is provided by JetBrains who make tools for a variety of different languages. The PyCharm IDE can be downloaded from their site—see https://www.jetbrains.com/. Look for the menu heading 'Developer Tools' and select that. You will see a long list of tools, which should include PyCharm.



Select the PyCharm option.

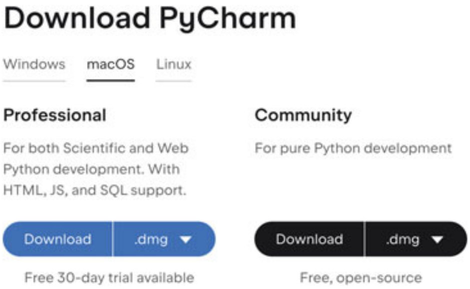
On the next page displayed there should be a large button labelled ‘DOWNLOAD’ select this option:

PC
PyCharm
The Python IDE for Professional Developers
DOWNLOAD
Full-fledged Professional or Free Community

On the next page there are then two download options available: *Professional* and *Community*.

- the Professional version is the charged for option but included several additional elements over the free community version,
- the Community version is free.

For most of the work I do in Python the Community version is more than adequate, and it is therefore the version we will download and install (note with the Professional version you do get a free trial but will need to either pay for the full version at the end of the trial or reinstall the Community version at that point).



For both Scientific and Web Python development. With HTML, JS, and SQL support.

For pure Python development

Free 30-day trial available

Free, open-source

Assuming you selected the Community edition the installer will now download, and you will be prompted to run it. Note you can ignore the request to subscribe if you want). Note that the web browser should select the correct operating system for you however, if it does not then you can manually select it above the ‘Professional’ and ‘Community’ headings.

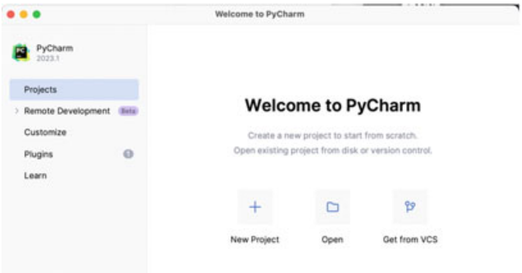
You can now run the installer and follow the instructions provided.

3.5.2 Setting up the IDE

You need to first start the PyCharm IDE. Once started the first dialog shown to you asks if you want to import any settings you may have had for another version of PyCharm. At this point select ‘Do not import settings’.

Step through the next set of dialogs selecting the look and feel (I like the light version of the IDE), whether you want to share data with JetBrains, etc. Once you have completed this click the ‘Start PyCharm’ option.

You should now be presented with the landing screen for PyCharm:



Welcome to PyCharm

Create a new project to start from scratch.
Open existing project from disk or version control.

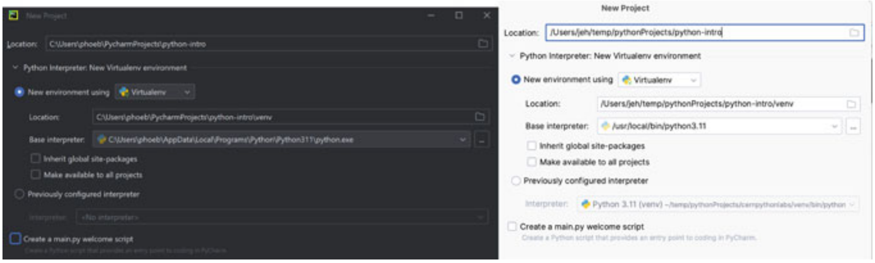
We will now create a project for you to work in. A Project in PyCharm is where you write your programs and how you configure what version of Python you are using and any libraries that you might need (such as graphics libraries etc.).

Click on the ‘New Project’ option in the landing dialog.

You will now be asked where you want to create this new project. Again you can use the default location but you will need to give it a name, we will call ours *python-intro*. This is done by changing the last element of the ‘Location’ field to read ‘python-intro’.

It is also worth at this stage making sure that the Python interpreter you installed has been picked up by the IDE. You can do this by opening the ‘Project Interpreter: New Virtualenv environment’ option and making sure that the Base interpreter field is populated with the installation of Python on your machine (this should be picked up automatically for you). If all is OK then select ‘Create’; if the Base Interpreter is not specified or is incorrect then click on the ‘…’ button to the right of the field and browse to the appropriate location.

The ‘New project’ dialog is shown below on the left for Windows 11 and on the right for an Apple Mac:



New Project

Location: C:\Users\phoeb\PycharmProjects\python-intro

Python Interpreter: New Virtualenv environment

● New environment using Virtualenv

Location: C:\Users\phoeb\PycharmProjects\python-intro\venv

Base interpreter: C:\Users\phoeb\AppData\Local\Programs\Python\Python311\python.exe

□ Inherit global site-packages

□ Make available to all projects

○ Previously configured interpreter

Interpreter: <No interpreter>

□ Create a main.py welcome script

Create a Python script that provides an entry point to coding in PyCharm.

New Project

Location: /Users/jeh/temp/pythonProjects/python-intro

Python Interpreter: New Virtualenv environment

● New environment using Virtualenv

Location: /Users/jeh/temp/pythonProjects/python-intro/venv

Base interpreter: /usr/local/bin/python3.11

□ Inherit global site-packages

□ Make available to all projects

○ Previously configured interpreter

Interpreter: Python 3.11 (venv) ~/temp/pythonProjects/campythonlabs/venv/bin/python

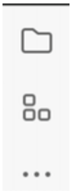
□ Create a main.py welcome script

Create a Python script that provides an entry point to coding in PyCharm.

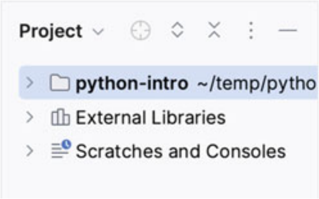
Next make sure that the ‘Create a main.py welcome script’ is **NOT** selected as this will create a sample program for you which contains far more than a simple ‘Hello World’ message.

On opening the PyCharm project you might see a Welcome message; click ‘Close’ and the project will be set up for you.

When you open the project, you will be shown a blank work space. On the left-hand side you should see at the top of the thin bar three icons:



Select the top one which looks a bit like a folder—this is the project view. This will look something like:



Project

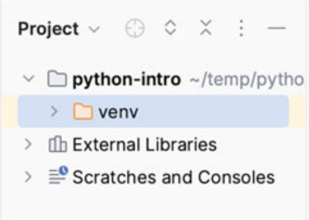
python-intro ~/temp/pytho

External Libraries

Scratches and Consoles

This illustrates the structure of your project. You should ignore ‘External Libraries’ and ‘Scratches and Consoles’ and open the top node. If you called your project something other than python-intro you will find that this node has the name you entered rather than python-intro but that is fine.

When you open the top node you will see that it currently contains a directory called venv—ignore this one as well and close this node so that you should now see:



We are now ready to get started.

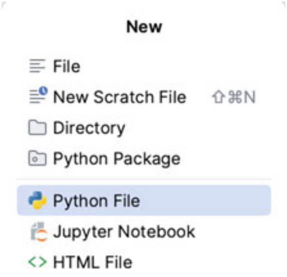
3.6 Writing Python Programs

In this section we will use PyCharm to write our first Python program. If you are using a different IDE then the basic ideas will be the same—create file with a.py extension and run it—however the menu options and buttons may be (will be) different.

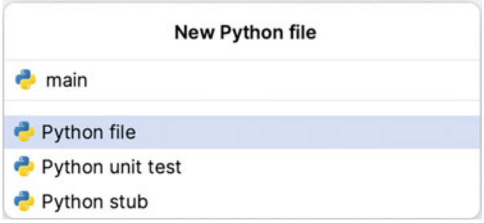
The first step will be to create a new file within PyCharm to store our 'Hello World' program.

First make sure that the selected node in the project tree is the top level node with the name of your project.

Next select the PyCharm File->New menu option. This will display a 'New' menu with the option to create a new 'Python File'. Make sure that this is the option you select, do not select the plain File option. If you don't select the Python File option PyCharm will not realize that you want to run the associated file as a Python program. For example:

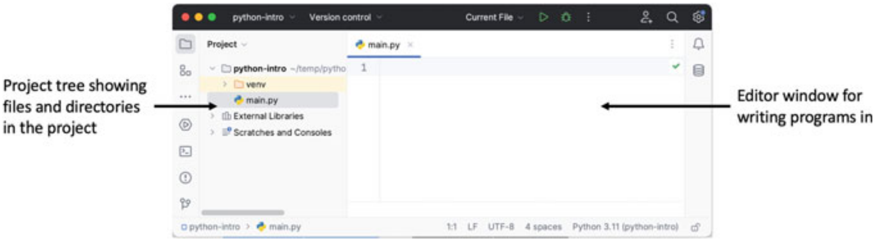


When you select the 'Python File' option PyCharm will display a new dialog allowing you to enter the name of the file you want to create. All Python files must end with a.py extension thus you could enter the name 'main.py' into this dialog, however the '.py' is optional here as PyCharm will add that extension for you, thus you can merely enter 'main' if you wish:



Again make sure you have selected the ‘Python file’ option and neither of the other options as they will not work with the simple Hello World program we are creating. Now press enter.

Once you have entered a filename for your program, the enter will create the file for you, for example called `main.py`, and display it on the right-hand side of your editor as an empty file. Note that the editor is divided into two areas, one is the project tree on the left and one is the editor window on the right, for example:



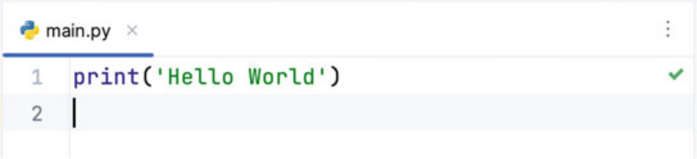
Project tree showing files and directories in the project

Editor window for writing programs in

Within the editor window enter the code to print out ‘Hello World’, for example:

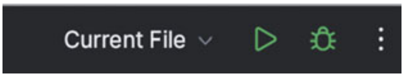
```
print('Hello World')
```

Now save the file. For example,

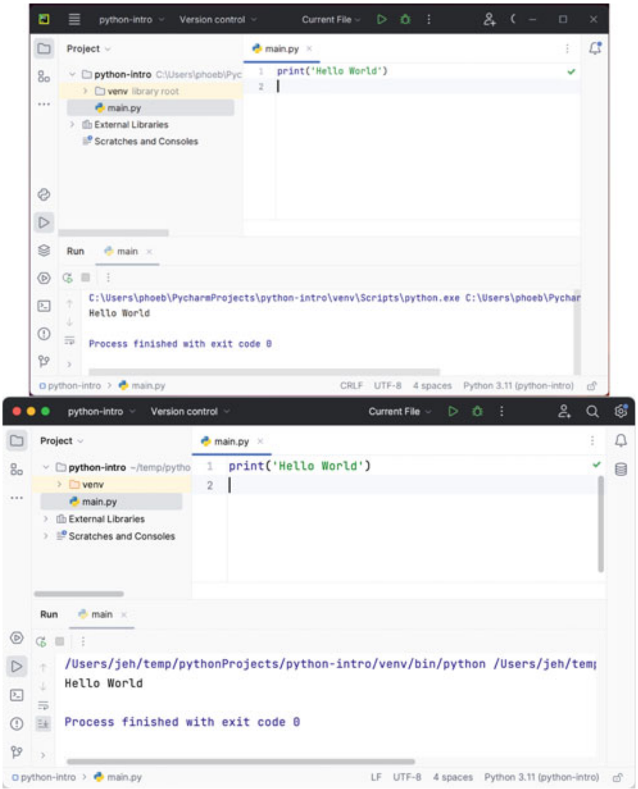


```
print('Hello World')
```

You can now run the file. To do this look above the name of the file in the editor and check that the field next to the little green arrow head say ‘Current File’. This indicates that the IDE will run the current file when you click on the green arrow, for example:



Now click on the green arrow head and your program should now run. Below the left hand screen shot is from running PyCharm on a Windows 11 machine and the right-hand one is from an Apple Mac. As you can see they are very similar:



```
print('Hello World')
```

```
C:\Users\phoeb\PycharmProjects\python-intro\venv\Scripts\python.exe C:\Users\phoeb\Pychar
Hello World

Process finished with exit code 0
```

```
print('Hello World')
```

```
/Users/jeh/temp/pythonProjects/python-intro/venv/bin/python /Users/jeh/tem
Hello World

Process finished with exit code 0
```

The output from the program is displayed in a ‘Run’ window at the bottom of the IDE. This output shows you what was run (Python and the `main.py` file), the output from the program ‘Hello World’ and that the process (that ran the program) exited successfully with an exit code of ‘0’.

You have now run your first Python program!

3.7 A Note on Filenames

One point to note is that file names in Python need a little care.

Firstly, you should always name your files using all lowercase letters. For example the filename `utils.py` is fine but the file Utils.py should be avoided (by convention).

Secondly, you should try to avoid splitting file names up into constituent words, such as `my_utils.py`, this is a style thing mostly, and one that not everyone (including myself) adheres to. However, if you do need to split a filename up to make it more readable always use an underbar '_', never use a hyphen '-'. If you do use a hyphen this will not cause a problem right now, but at some point in the future it may come back to bite you. This is because if you ever need to reference one Python file from another the hyphen '-' will break such an operation as the 'import' command parses the '-' as a minus sign and thus will try to subtract part of the file name from another part resulting in an error instead of just being the file name.

3.8 Online Resources

See the Python Standard Library documentation for:

- https://docs.python.org/3/using/index.html with documentation for Python setup and usage.
- https://docs.python.org/3/faq/windows.html Python on Windows FAQ.
- https://www.jetbrains.com/pycharm/ The PyCharm IDE home page.