

SUMMER SCHOOL 2021

Introduction to Python Programming

Setting up the Python Environment

Binjamin Barsch

Summary of commands used in this lesson

Command	Meaning
<code>python</code>	Starts a Python interpreter from a Linux terminal
<code>print()</code>	A Python function that prints text

Welcome to the second week of the CHPC Summer School 2021. In this week we will be focusing on an introduction to Python programming. We assume that you have no background in Python programming so this course will cover the fundamental principles. We also assume that you have done week 1 that covers the basics of the Linux terminal and bash scripting.

What is programming?

Programming is just a process of using instructions that tells a computer how to perform a task. We use the programming language of Python to create these instructions. Other programming languages also exist like C, C++, and Java. We will be using Python as it is quite versatile, easy to use, and quick to learn.

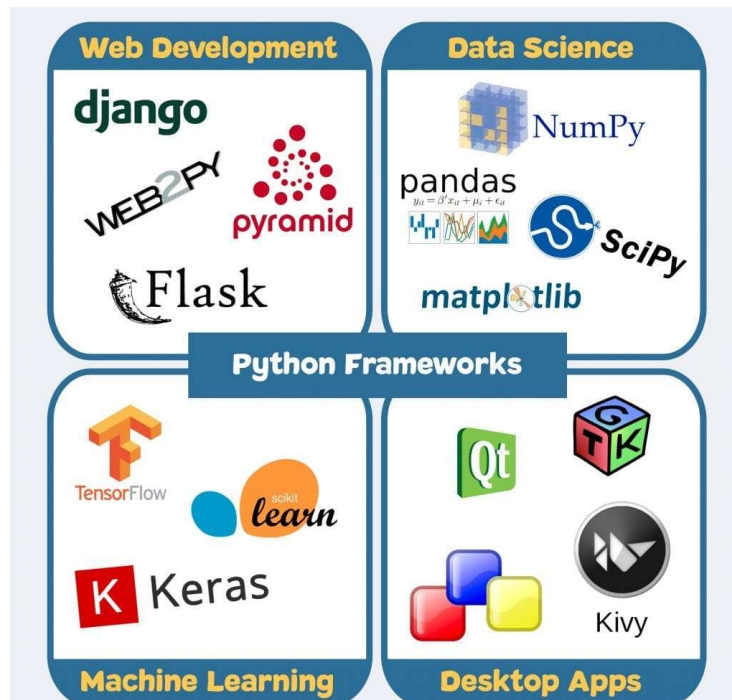
Python Overview

Python can be used to create pretty much any type of application, from data processing to web applications to running embedded systems. Like a Swiss Army Knife, Python may not be the best language for every specific application, but there's a good chance it's the second-best language for almost anything.

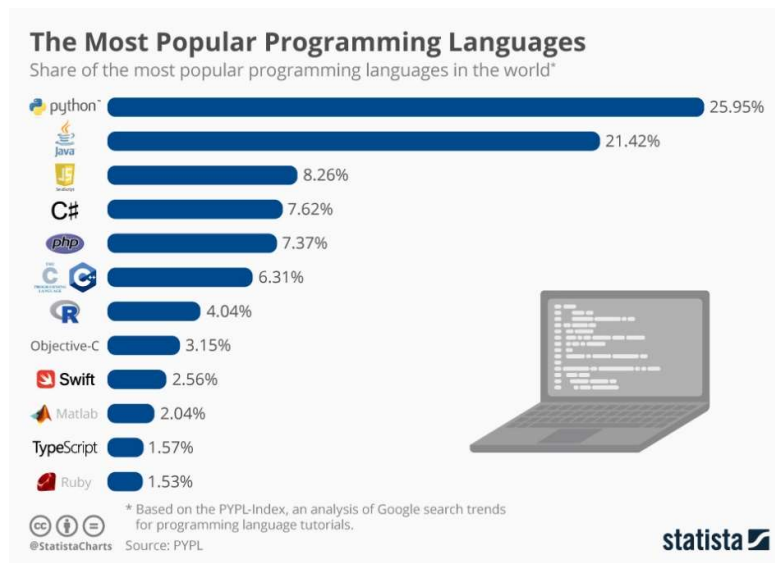
Who uses it?



What can you do with it?



How does it compare to other programming languages (2019):

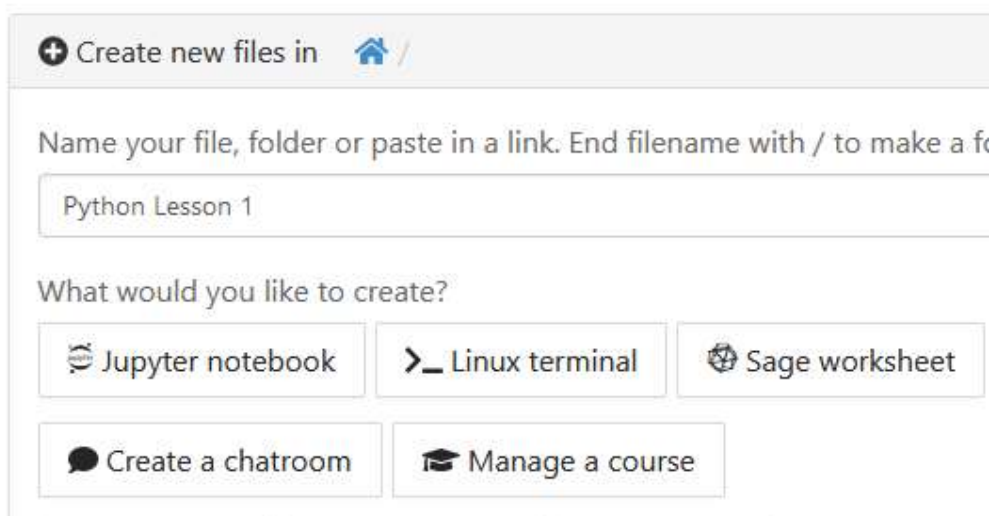


Let's get started

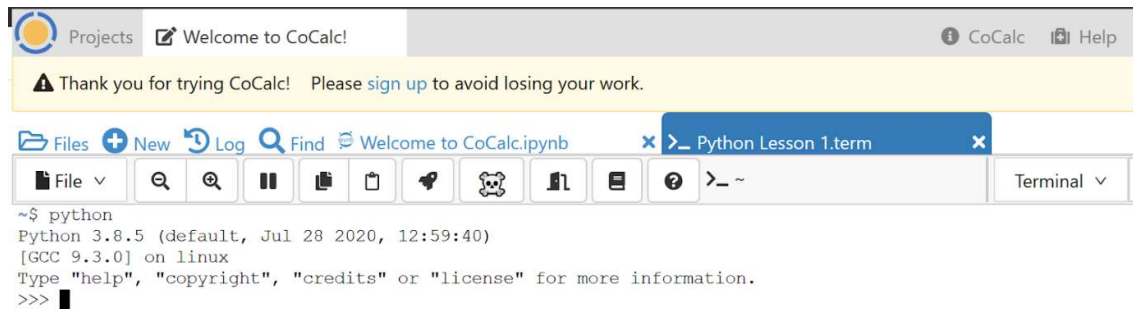
For the first few lessons we are going to use the similar environment that was used in week 1, that is using CoCal. You can also use any terminal environment described in week 1 like for Windows, Linux, or MacOS.

Step 1:

Open CoCalc in your browser and create a new Linux terminal, like you have done in week 1. Call it “**Python Lesson 1**” as shown below:



Now enter in the terminal the command “**python**” and following should appear:



The screenshot shows the CoCalc web interface. At the top, there's a navigation bar with 'Projects', 'Welcome to CoCalc!', 'CoCalc', and 'Help'. Below this is a yellow banner with a warning icon and the text: 'Thank you for trying CoCalc! Please [sign up](#) to avoid losing your work.' The main area has a toolbar with icons for 'Files', 'New', 'Log', 'Find', and a tab for 'Welcome to CoCalc.ipynb'. Below the toolbar is a terminal window titled 'Python Lesson 1.term'. The terminal shows the command '~\$ python' and the output: 'Python 3.8.5 (default, Jul 28 2020, 12:59:40) [GCC 9.3.0] on linux Type "help", "copyright", "credits" or "license" for more information. >>>'. A cursor is visible after the last '>>>'.

```
~$ python
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

You will see the next line saying “**Python 3.8.5** ...” which just tells us the Python version. And the last line shows “>>>” which indicates that you have just started a new python environment called a Python interpreter.

How to use these notes

The previous image is a snapshot of the browser environment that simulates a Python interpreter. The notes that you will be using is a direct copy of the content. The green outlined black text boxes represent code entered into a Python interactive session (which is always in bold - copy or type only these parts) together with the output of the Python interpreter (which you should not copy. Also, don't copy the interpreter prompt ">>>". It will cause errors! Also remember: the interactive mode is only for quick throwaway tasks and testing - anything more than 3 lines should probably go in a script (a what? – we will get to that).

```
~$ python
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on Linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Let's Begin

Since we are focusing on HPC let us start off with some simple math tasks.

Try out the following commands and we will explain afterwards what is happening.

Type out the following commands that are **ONLY** in bold in the Python interpreter and press the “**Enter**” key and you should see the follow responses from the Python interpreter:

```
>>>
>>> 40 + 2
42
>>> print(40+2)
42
>>> name = "wabbit"
>>> print(name)
wabbit
>>> x = 40
>>> y = 2
>>> x
40
>>> y
2
>>>
```

Now let us do a simple sum by entering the following:

```
>>>
>>> sum = x + y
>>> print(sum)
42
>>>
```

Let us take a step back and go through what we have done so far.

First off, we did some basic math by adding two numbers. These types of numbers are also known as integers. You should have seen the answer as “42” which is an integer. We can also print or display that answer as text, with “**print(40+2)** “. We also made a variable called “**name**” which stores the text “ **wabbit** “. Note the two exclamation marks “ ”, makes sure the variable is stored as text, again also known as a string.

We then started assigning some variables like “**x = 40**” and “**y = 2**”. We can print out what those variables represent at any time if you just enter “**x**” and “**y**”. Then we made a sum as “**sum = x + y**” which we then printed out on the terminal using the “**print**” function: “**print(sum)**”.

Well done! That’s it! You have been doing basic Python programming.

Summary

This is the end of this lessons on getting an overview of Python and how to set it up. In the next lesson we will be focusing on variables and data types!

For more information have a look at the following resources:

General Python Resources:

- <https://www.python.org/>
- <https://www.w3schools.com/python/default.asp>
- <https://automatetheboringstuff.com/>
- <https://realpython.com/>

Data Science:

- <https://www.datacamp.com/courses/intro-to-python-for-data-science>
- <https://realpython.com/tutorials/data-science/>
- <https://towardsdatascience.com/tagged/python>

Machine Learning:

- <https://www.tensorflow.org/>
- <https://keras.io/>
- <https://scikit-learn.org/stable/>

GUI Applications:

- <https://wiki.python.org/moin/TkInter> (This is where everyone starts)
- https://www.tutorialspoint.com/python/python_gui_programming.htm
- <https://realpython.com/python-gui-tkinter/>
- https://www.tutorialspoint.com/pyqt/pyqt_hello_world.htm

Web Development:

- <https://www.djangoproject.com/>
- <https://flask.palletsprojects.com/en/1.1.x/>

Mobile App Development

- <https://kivy.org/#home>

Other

- <http://ode.org/> (simulating rigid body dynamics, pendulums, collision dynamics)
 - <https://micropython.org/> (micro-electronics, IoT)
 - <https://opencv.org/> (Image Processing, Computer Vision)
-