

Intro to Python

Intro to Python



What is Python?

Python is a programming language. With it, we will write plain text documents that do things. Keep in mind that *you have already been programming in this course!*

We Programmed with HTML and CSS

Recall the steps we went through to write a website:

- We defined what kind of document we were writing with the `<DOCTYPE>`
- We imported additional code, in the form of CSS, into the HTML document.
- We wrote out instructions for how the text was supposed to look with commands like

``

Then, we used a web browser to interpret the code in the HTML document, and the browser returned a result in the form of a website.

We programmed with LaTeX

Here were the steps we used to program with LaTeX:

- We defined what kind of document we were writing with `\documentclass{}`
- We imported additional code, with `\usepackage{}`
- We wrote out instructions for how the document was supposed to look with commands like

`\textbf{}`

Then, we used a LaTeX engine (being run by OverLeaf) to interpret the LaTeX code, which returned a result in the form of a typeset pdf.

We even programmed with RegEx

Regular expressions are much simpler than HTML or LaTeX, but one way to think of a regex like `\b\w+ly\b` is that it is a very tiny program, that we then used `egrep` to interpret, resulting in all of the matching strings.

Programming in python will be a lot like this

When we write a python program, we will

- Define what kind of document it is at the top with `#!/usr/bin/python3`
- We will import additional code, using `import`
- We will write out instructions for what the document is supposed to do with code.

We will then use the python interpreter to interpret our code, and do what we asked it to do.



Getting Set Up

To get set up for writing python scripts, I recommend you create a folder inside the course folder called `python`. This is where you should save your python scripts, and where you should navigate to in your terminal when we run python scripts.



A first look at python

We can get a first look at python by opening your terminal and running `python3`.

```
python3
```

Your ordinary terminal should be replaced by some text about the python version, and then `>>>`. The `>>>` is there to indicate that anything we type in will be interpreted as python code.

Using python like a calculator

One way we could use python is as a calculator. Back in the [Textual Analysis](https://uk.instructure.com/courses/2051722/pages/textual-analysis) (<https://uk.instructure.com/courses/2051722/pages/textual-analysis>) module, I found that Jane Austen used "the" 29,696 times, and used a total of 798,954 words overall. To find the proportion of here words that were "the", I need to do division in python. (Don't copy the >>> from the code below.)

```
>>> 29696 / 798854  
0.03717325068160139
```

The `/` symbol is the code for division in python. Note that we didn't include the commas when we typed in the numbers. An important thing about python, and all programming languages, is that we need to enter all of our numbers without commas.

Getting out of python!

To get out of python and back to your usual terminal, run `quit()`

```
quit()
```



Writing our first script

To write our first python script, open SublimeText (or whichever text editor you're using) and create a new blank document. While the document is still blank save it as `hello.py` in your python folder. It might seem silly to save a blank document, but this will let SublimeText know that we are writing a python script, and it will give us nice syntax highlighting.

Writing Hello World

The traditional first python script to write is a "Hello world!" script. Inside your `hello.py` document, write the following.

```
print("Hello world!")
```

It is important that you enter it in *exactly* as it appears here.

- The command `print` must be lower case.
- `"Hello world!"` must be inside quotation marks.
 - It's less important what exactly you put *inside* the quotation marks.

- The opening `(` must be matched with a closing `)`.

Save your script!

Running `hello.py`

To run the script, [navigate in your terminal](#)

(<https://uk.instructure.com/courses/2051722/files/102817016?wrap=1>) to the `python` directory where you saved the script. Then, run the following.

```
python3 hello.py
```

What should happen next is the terminal will print out `Hello world!`, or whatever it was you put in quotes.

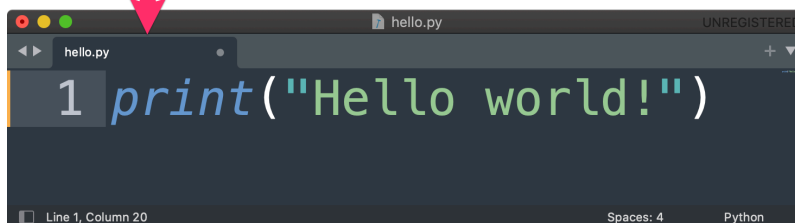


What just happened?

The Script.

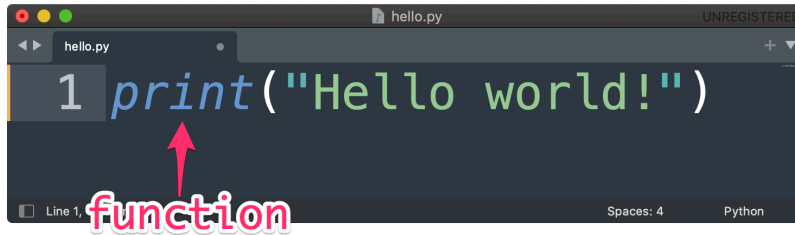
Our document, `hello.py` is a *script*. All a script is is a plain text document. This specific script is a python script because it contains text that python can interpret. It doesn't *have* to end in `.py`, but its helpful for keeping things straight.

Script



Functions

Inside of the script is code. Specifically, the `print()` function. Functions are the pieces of code that *do* things. In this case, the `print()` function prints its arguments.



Arguments

Functions take *arguments*. Arguments are the pieces of information the functions use to do what they do. In this case, `print()` has to print something! The *value* we pass to

`print()`'s *argument* is the string `"Hello world!"`



Running the script

For the script to do anything, we need our code to be *interpreted*. Sort of like how we had to open our HTML documents in a web browser for the markup to be interpreted, we need to tell python to interpret the code in `hello.py` as python code, and do something about it.

We do this by telling our computer to run `python3`, but instead of launching the interactive prompt, to interpret the text in `hello.py` as code as if we had just typed it all in.

"run python3"



A terminal window with a dark background. The title bar shows 'python — joseffruehwald@Josef-Fruehwalds-MacBook-Pro — ..1/2022/p...'. The prompt is a green arrow. The first command is 'python python3 hello.py' in cyan and white. The output is 'hello world' in white. The second command is 'python' followed by a white cursor box. A red arrow points from the text 'run python3' to the 'python3' in the first command. Another red arrow points from the text 'interpret the code in this document' to the 'python' in the second command.

```
python python3 hello.py
hello world
python
```

"interpret the code in this document"