EXPLORE | DIGITAL SKILLS

# Introduction to Python Scripting

## Outline

This train is structured as follows:

- An introduction to python scripting;
- Using python in command terminal and the use of python scripts;
- Installation of VSCode as an IDE;
- Writing and executing your first python script;
- Import and execute a python script in Jupyter notebook.

## Objectives

In this train you will learn how to:

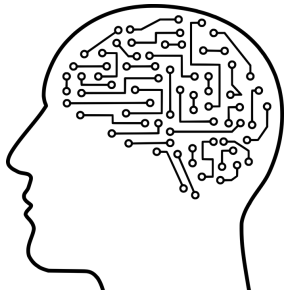
- Write and execute your own python scripts;
- Understand the different tools used for python scripting;
- Installing an IDE to write python scripts;
- Execute python scripts in a command terminal; and
- Import python scripts and run them in a Jupyter notebook.

EXPLORE | DIGITAL SKILLS

# An Introduction to Python Scripting

Python scripting is often used to automate a specific series of tasks, leading to a more efficient process. Python scripts can be used for:

- Developing both desktop and web applications;
- Development of complex scientific and numeric applications;
- To facilitate data analysis and visualization;
- Machine learning and artificial intelligence; and
- Business applications.



Running Python in Jupyter notebooks is not the only way to interact with code. You can run scripts through IDEs (Integrated Development Environments) such as PyCharm or VSCode or even through the command line. Through the rest of this train we will expand on the different tools used for compiling and executing python scripts.

# Tools used to compile and execute Python scripts

## Command line 

There are several reasons you would want to use Python in a command terminal.
These may include:

- Wanting to do quick one-liner calculations
- Executing a python script with parsed arguments
- Running simple, code.

## Integrated Development Environment (IDE)

An IDE  is an application that can be used to write code into a script. Python scripts  allow you to write much longer code that can then subsequently be executed in a command terminal  or imported to a jupyter notebook. IDE's facilitate this process. Examples of IDE's include VS code, Pycharm and ATOM

## Jupyter notebook

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain code, equations, visualizations and narrative text. Applications include:

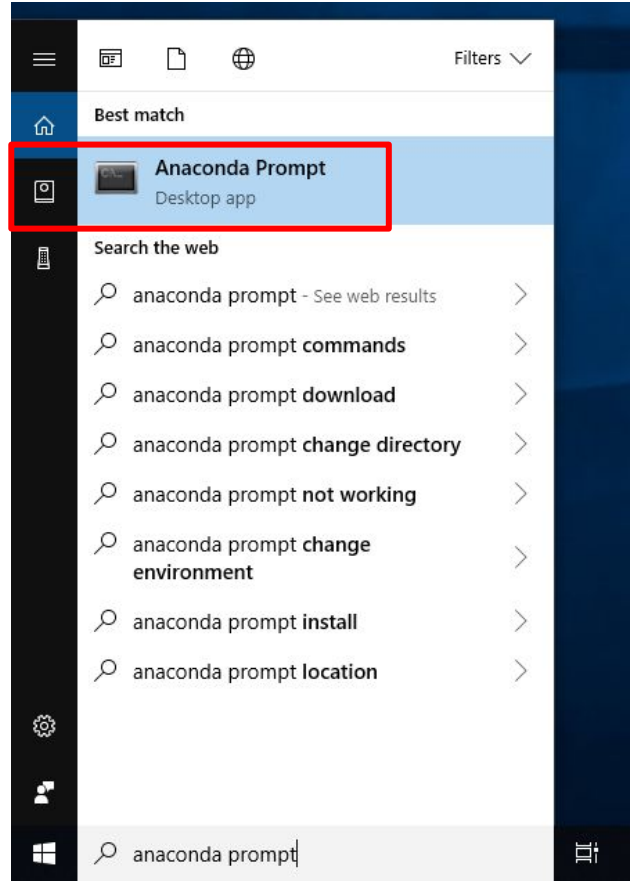
- Data exploration
- Machine learning
- Aids in explanation and interactivity of code

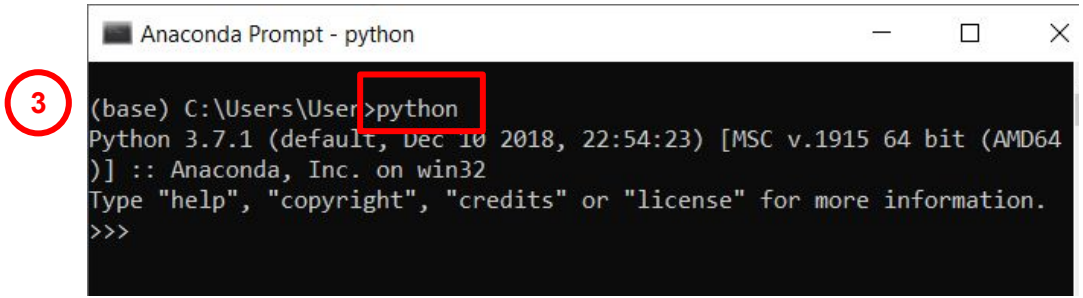
# Using Python in a command terminal on Windows

For Mac, head to the next slide.

Let's see how we can use Python in a command terminal on Windows:

1. Click the Start button;
2. Search for "Anaconda prompt" and select from menu;
3. Enter `python` at the blinking cursor and hit **Enter**. This will start an instance of python.




EXPLORE | DIGITAL SKILLS

# Using Python in command terminal on Mac

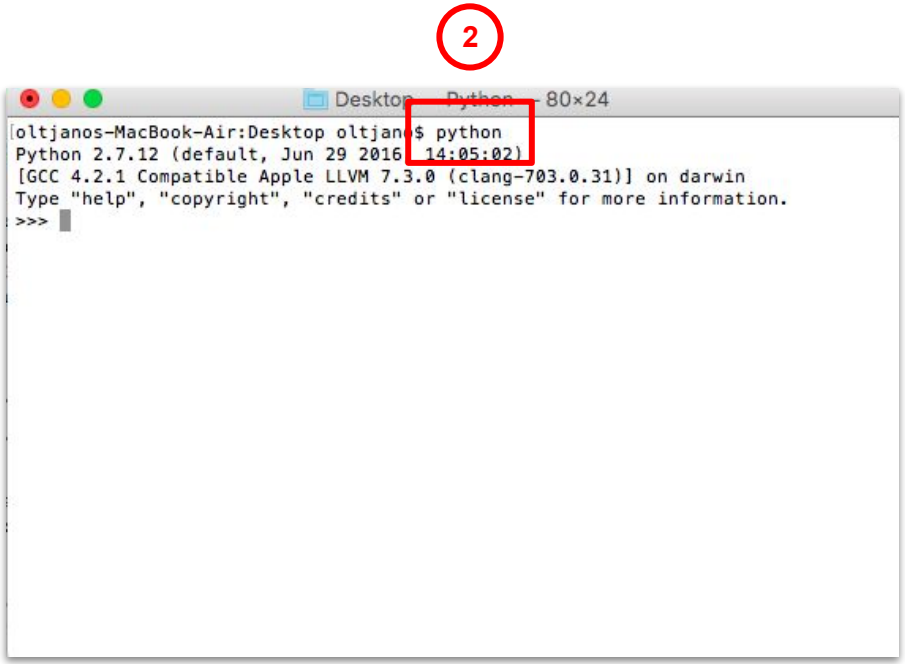
For Windows, head to the previous slide.

Let's see how we can use Python in a command terminal on Mac:

1. In Finder, navigate to **Applications > Utilities > Terminal.app;**
2. Start a python instance by typing the following at the blinking cursor and hitting **Enter:**

   `python`

3. You should get the output shown to the right.




EXPLORE DIGITAL SKILLS

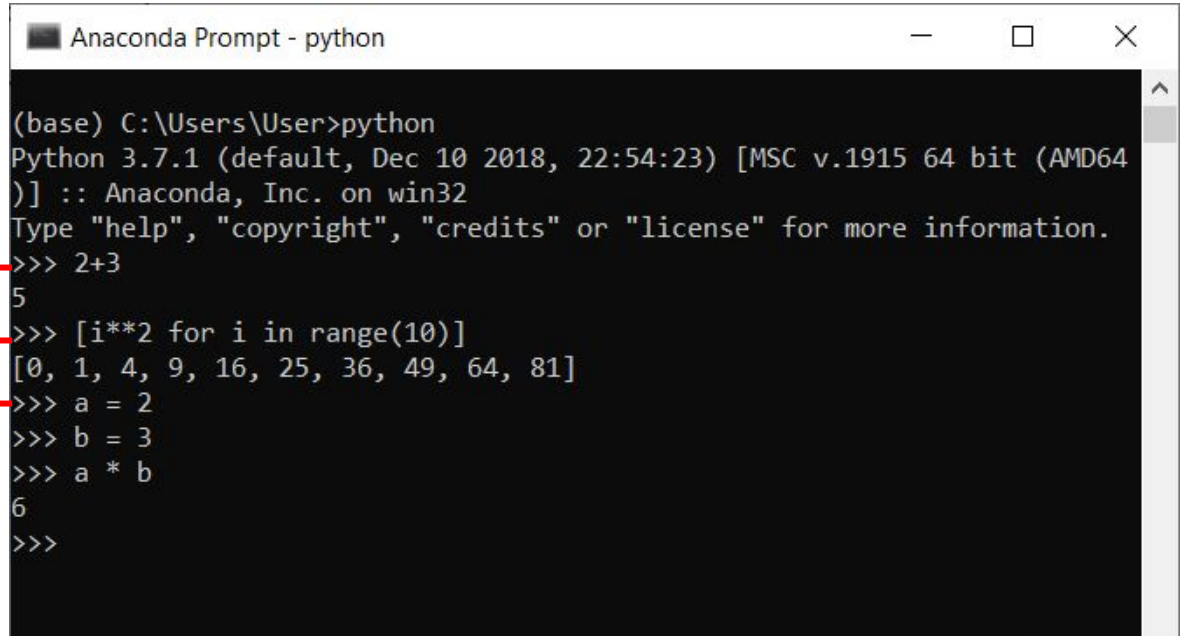
# Using Python in a command terminal

You can now run python commands in your terminal. Since python is a universal language, the commands will be the same whether you are using a Mac or Windows O.S. You can input code at each line starting with **>>>**. The output of your code is displayed below this line. Some examples:

Simple addition.

List comprehension of all square numbers below 10.

Setting variables, then multiplying them together.



# What are python scripts?

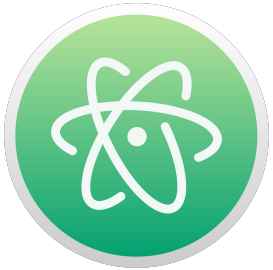
Python scripts allow you to write much longer code that can then be executed in a command line, or imported to a jupyter notebook.

Though you can start up Notepad or a word processor and write your own python scripts, it's much more advisable to use an **IDE** (**I**ntegrated **D**evelopment **E**nvironment) to write python scripts. These come with a number of features that make writing and executing your scripts far easier.

There are a number of IDEs available for free, including:

- PyCharm
- Atom
- VSCode
- Spyder

None of these are perfect. Each comes with its own advantages and disadvantages. As a growing data scientist, you will have your own preferences. For the purposes of this tutorial though, we'll be using **VSCode**.



# Installing Visual Studio Code on Windows

For Mac, head to the next slide.

## Step 1

Install the Windows version of VS code

## Step 2

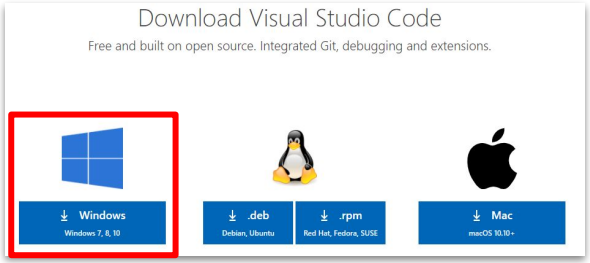
Download and install the Python extension for VS code

## Step 3

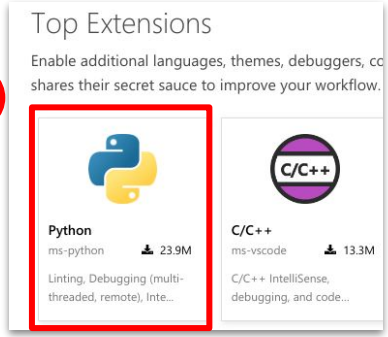
Open VS Code.
Click 'New file' or add a workspace you already use.

## Step 4

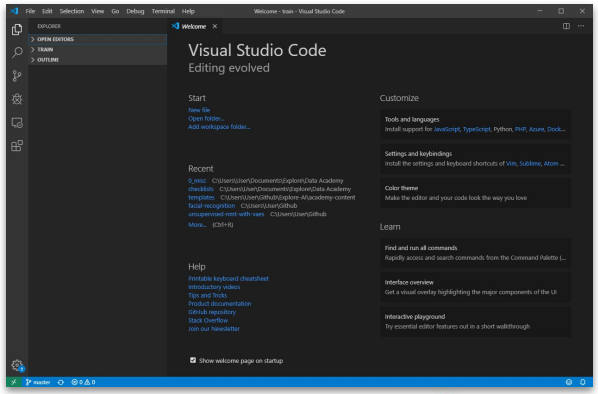
Start coding!




1




2




EXPLORE DIGITAL SKILLS

# Installing Visual Studio Code on Mac

For Windows head on to the previous slide

## Step 1

Install the Mac version of VS code

## Step 2

Locate downloaded archive file.

Move the visual studio code.app to the application folder

## Step 3

Download and install the Python extension for VS code.

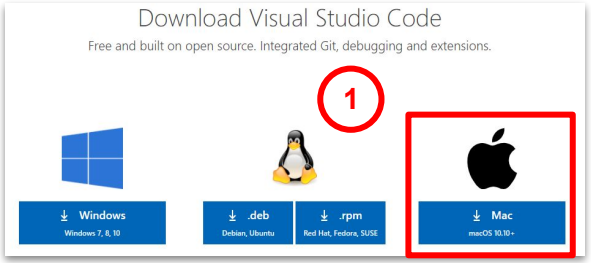
Be sure to take note of the Python requirements!

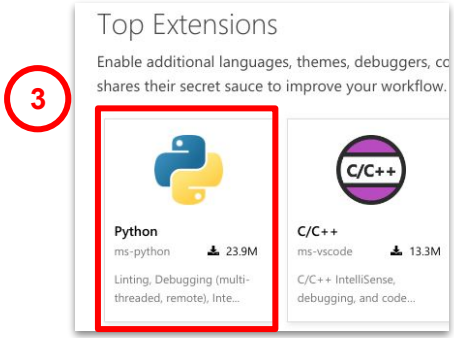
## Step 4

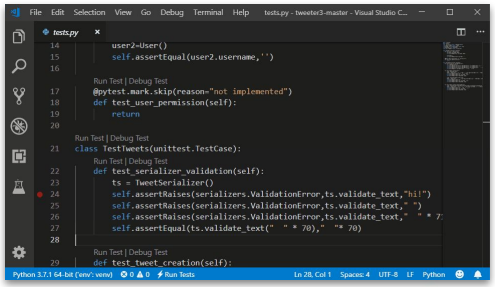
Open VS Code.

Click ‘New file’ or add a workspace you already use.

Start coding!



EXPLORE DIGITAL SKILLS

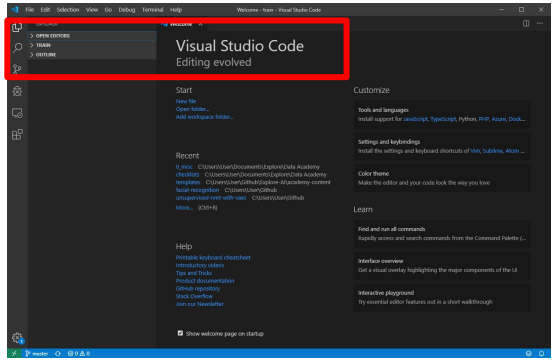
# Writing your first python script

Now, that you're ready to write your first python script. You can use a text editor or IDE of choice to write your script. Let's have a look at the process of writing a script.

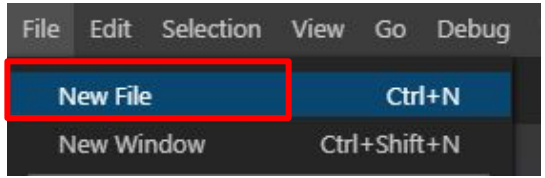
**Process of writing a script:**

1. Open your text editor or IDE of choice; for this train we will be using VS code;
2. File > New File;
3. To be able to execute your new script you will have to save your file with a **.py** extension.

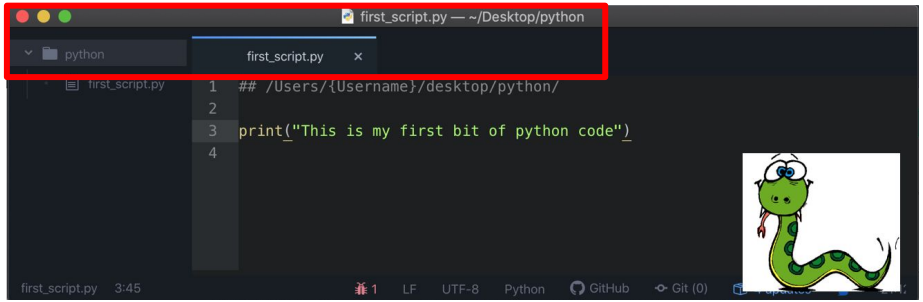
1




2




3




EXPLORE | DIGITAL SKILLS

# Executing your python script on Windows

For Mac head on to the next slide

## To execute your python script:

1. Open a new instance of an anaconda prompt.

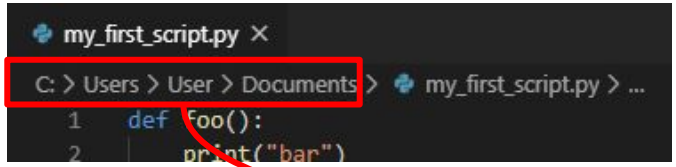
2. Navigate to the path where you saved your script. To do this, first take note of the path you saved your file in in VSCode. Then, convert the '>' symbols to '\' symbols (in my case **C:>Users>User>Documents>** becomes **C:\Users\User\Documents**).

3. Type the path from step 2 into your anaconda prompt as `cd <your path here>` which in my case becomes `cd C:\Users\User\Documents`

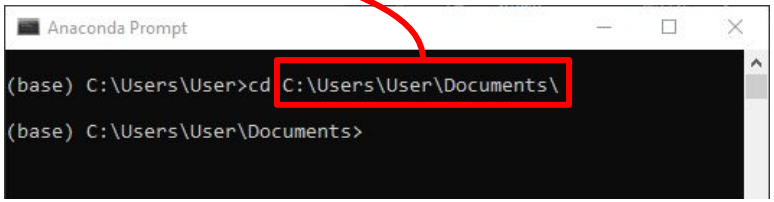
4. You can now execute your script by running `python my_first_script.py`. The output should be the printed word 'bar'.

1

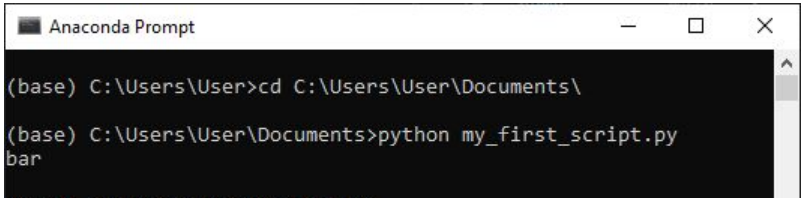

2




3




4




EXPLORE DIGITAL SKILLS

# Executing your python script on Mac

For Windows, head to the previous slide

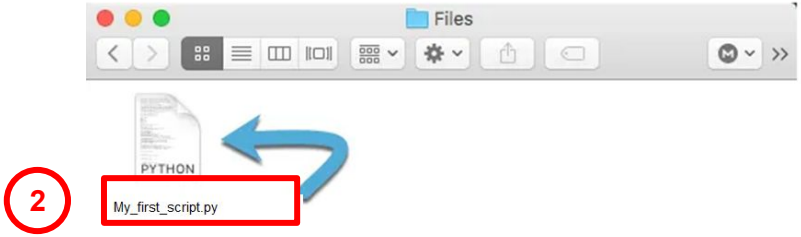
## To execute your python script:

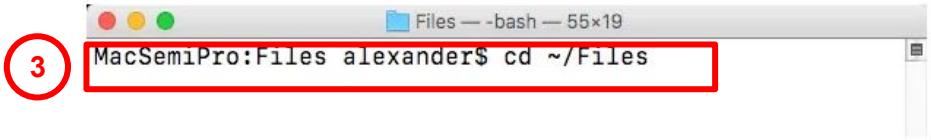
1. Open a new Terminal window;

2. Copy the path to your script's .py file. A shortcut: Right-click on the file while holding down the **Option** key and select **"Copy … as Pathname"** to copy the path to your clipboard;

3. In your Terminal window, type `cd` and then paste the path that you copied in Step 2;

4. You can now execute your script by typing `python my_first_script.py` and hitting Enter. The output should be the printed word 'bar'.




EXPLORE | DIGITAL SKILLS

# Importing a python script into a Jupyter notebook

Let's say you've written a very useful python script that you'd like to run in a jupyter notebook. You can run it by doing the following:

1. Open a command prompt in the same directory your script is in and type `jupyter notebook`, then hit **Enter**. This will open an instance of jupyter in your browser.

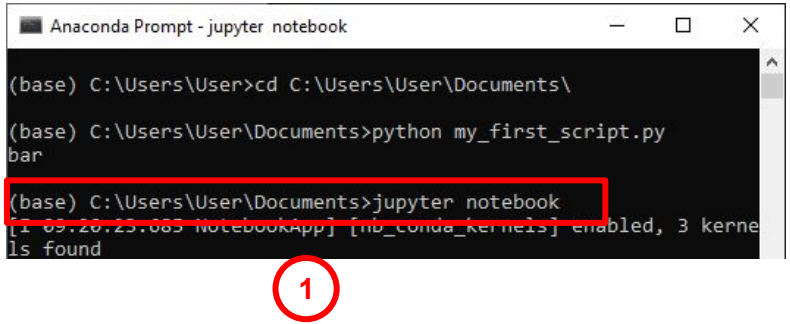
```
In [1]: import my_first_script
```

2. Start a new notebook. In the first cell of the new notebook, type `import my_first_script` and execute this cell by hitting "Shift+Enter".

```
In [2]: my_first_script.foo()

        bar
```

3. You can now access the foo function in this script by typing `my_first_script.foo()`. Executing this in a new cell will print the word 'bar'.




EXPLORE | DIGITAL SKILLS

# Differences between notebooks, scripts, and the command line



| Notebooks | Scripts | Command Line |
|---|---|---|
| Medium scale coding operations. About 100-200 lines of code as longer length notebooks become unmanageable | Large scale coding operations. Can cover thousands of lines of code. | Small scale coding operations. Usually only a couple of lines |
| Can import python scripts | Can import other python scripts. | Can import python scripts and execute python scripts. |
| Not recommended for a production pipeline. | Can form the backbone of a production pipeline. | Not used in production outside of debugging and monitoring. |
| Useful for presenting and exploring data | Useful for writing large amounts of code | Useful for executing python scripts and debugging. |
| Can execute cells in any order. | Always executes from top to bottom | Always executes from top to bottom |
| Does not take parsed arguments | Can work with parsed arguments | Can parse arguments to a script. |

# Conclusion

- In this train we were introduced to python scripting and the different tools that can be used to perform scripting;
- We learned how to use the command terminal to execute short python code and went on to  install an IDE to compile and execute our own python scripts;
- In this train we learned how to import and run a python script in jupyter notebook and learned the difference uses of each python scripting tool;
- You are encouraged to apply all that you have learned from this train, to write and execute your own code.


EXPLORE | DIGITAL SKILLS

# Extra Reading

Extending python scripts with arguments:

- Argument parsing
- Argument-parsing-in-python

Python scripting:

- Running python scripts
- Python scripts

Visual Code Tips and Tricks:

- Visual code tip and tricks

Python scripts vs modules:

- Python modules vs scripts
- Reusing code