Computer Lab 1
Part One: Intro to Computing in Python

Computing in Python

Simple

Readable

Easy to explain

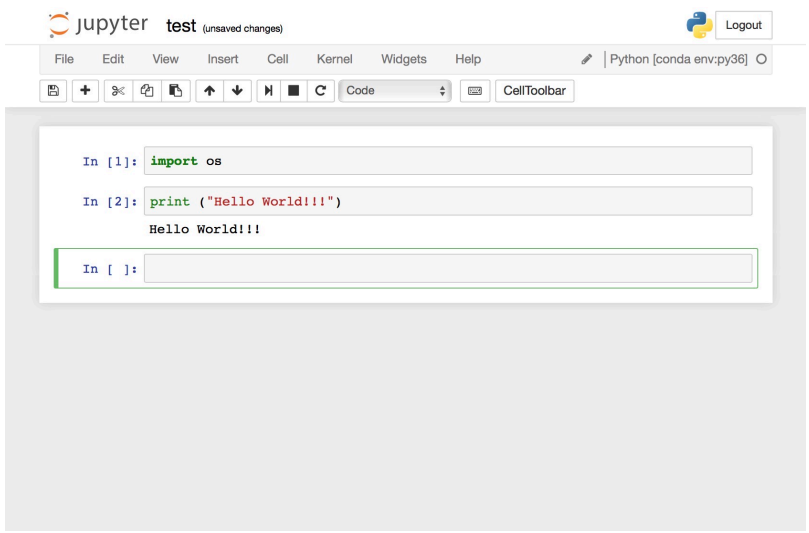
Instant to execute

Python is an extremely popular and versatile high-level programming language that was created by Guido Van Rossum in the 90.

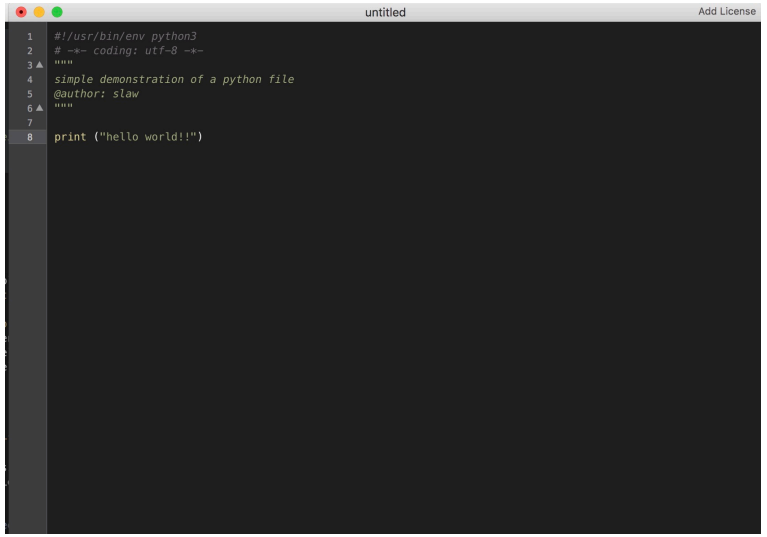
It is a general-purpose, procedural and object oriented programming language. (more next class!)

In this class we will learn by using i-Python Jupyter-notebooks.

Computing in Python



Interactive computing (Ipython, jupyter-notebook)



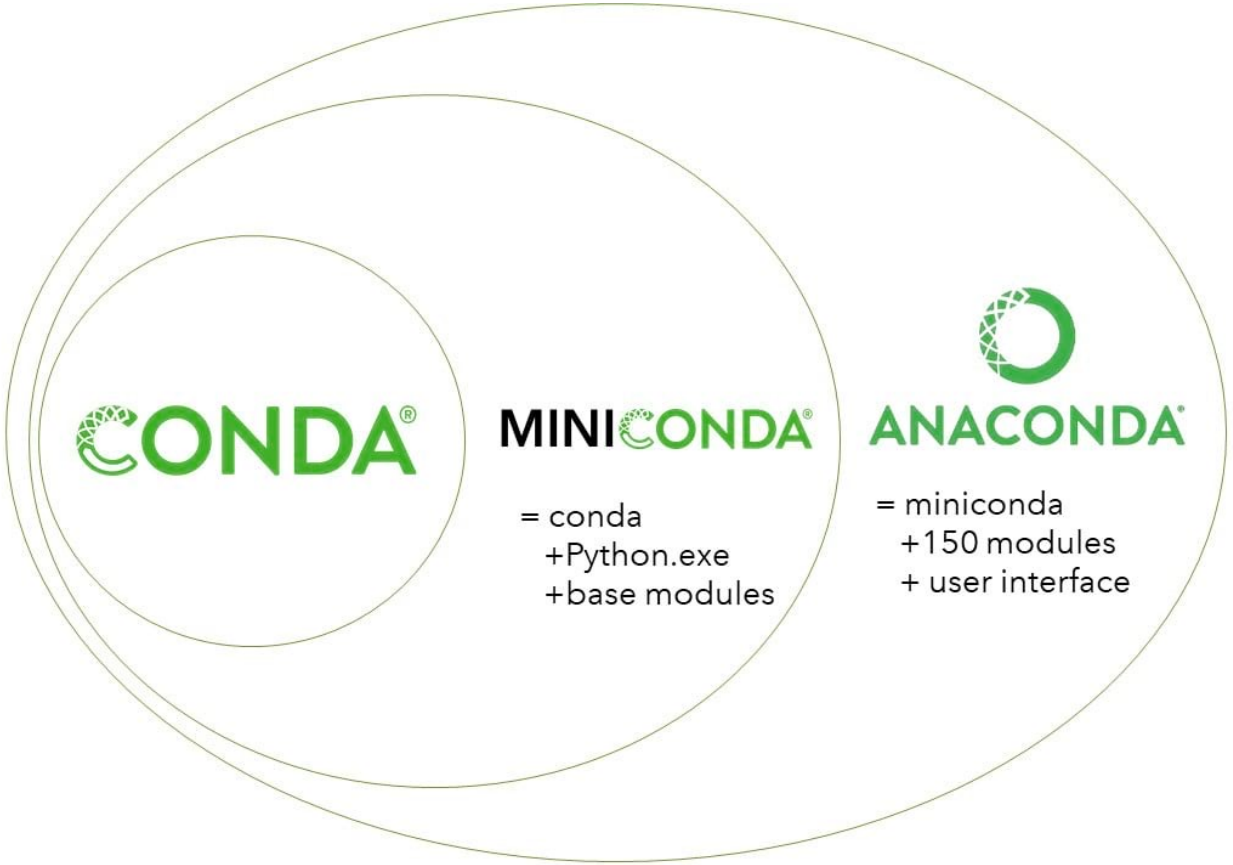
IDE (Pycharm, VScode) Text Editor (atom,vim,notepad++)

Conda

Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. It is a package manager that lets you install and update your packages.

https://conda.io

Conda/Miniconda/Anaconda (Easiest)

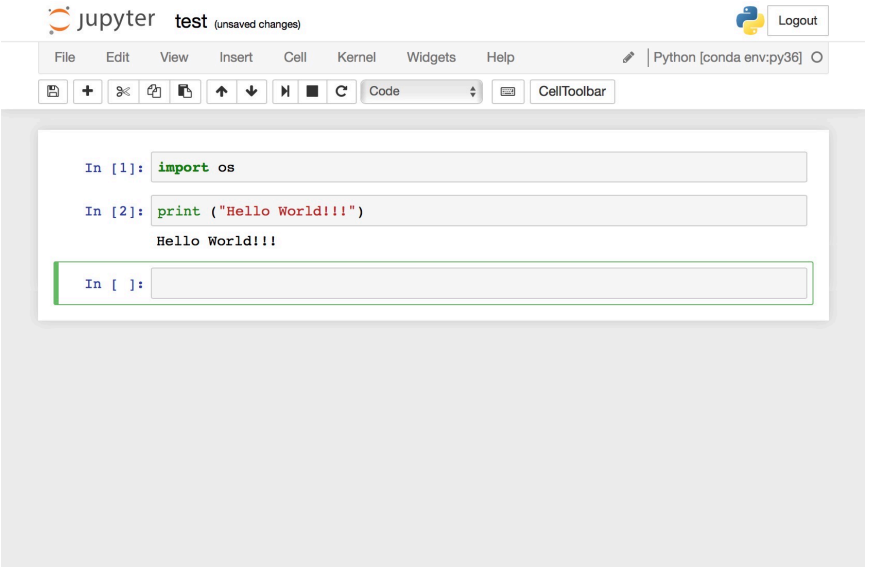


python™

IP[y]:

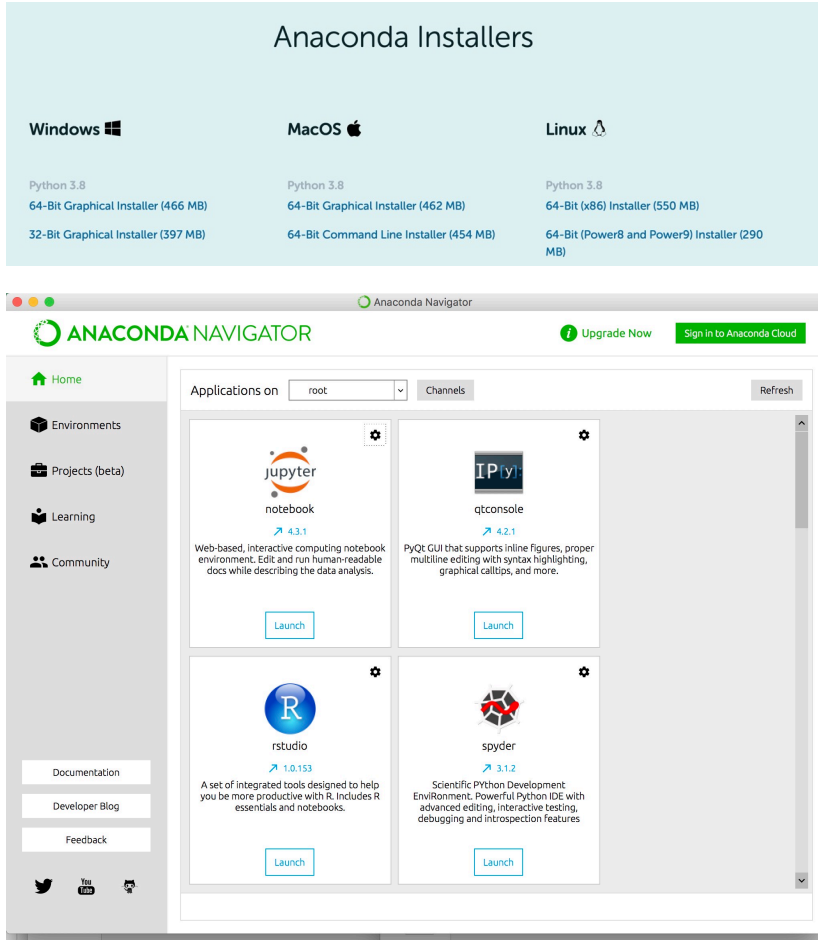
jupyter

Ipython Jupyter Notebook



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

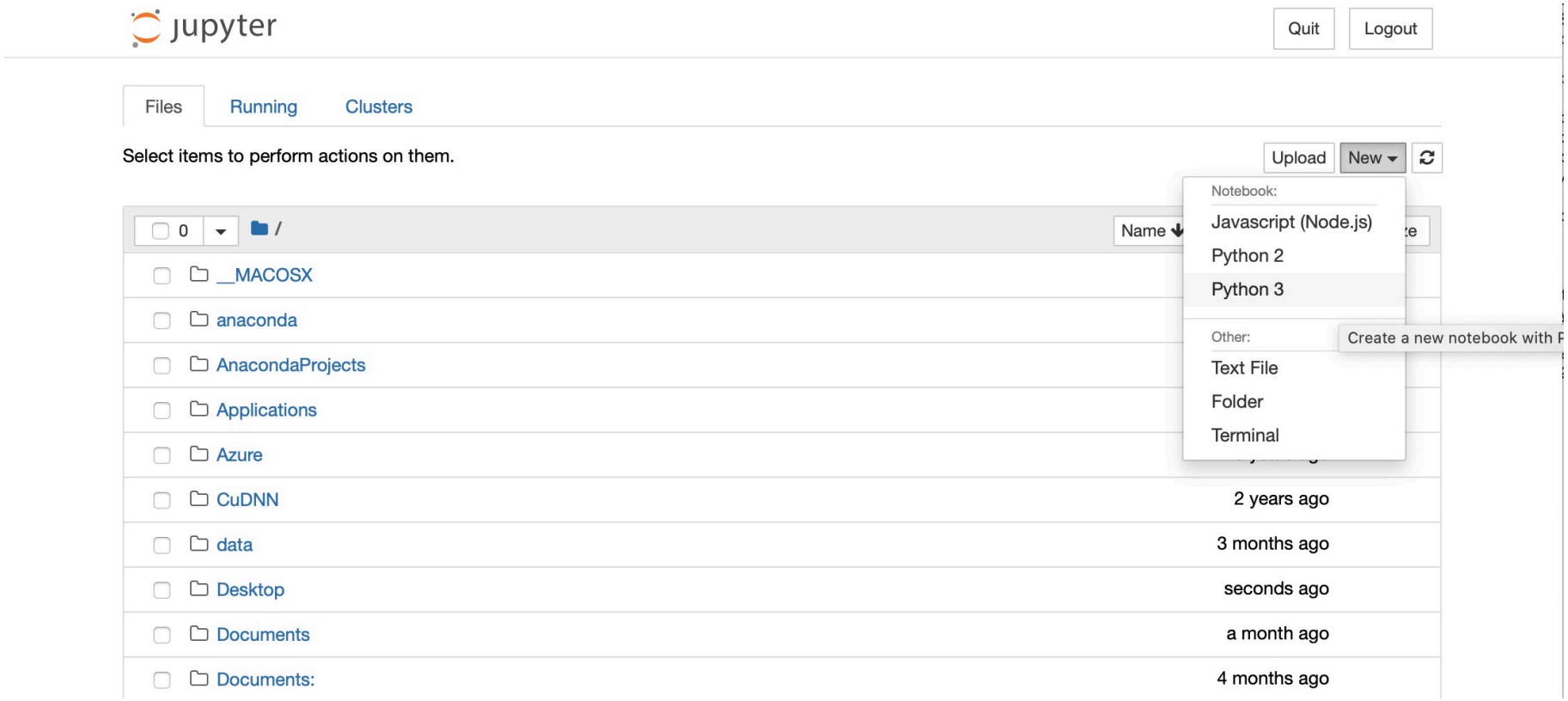
Software Installation



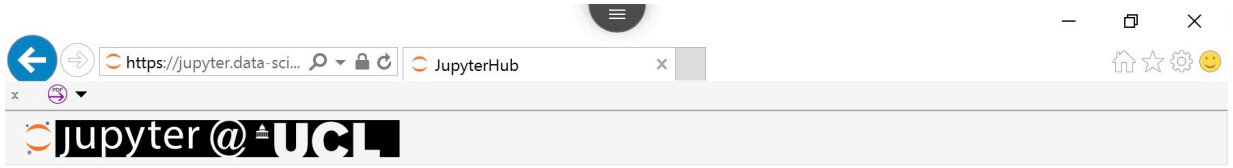
Suggested ways to install Jupyter-notebook

- *Download and install Anaconda with Jupyter-notebook locally.* www.anaconda.com/products/individual
- *Access Jupyter-notebook on UCL server.*
- *Install Docker-image with Jupyter-notebook installed.*
- *Google Colab Notebooks*

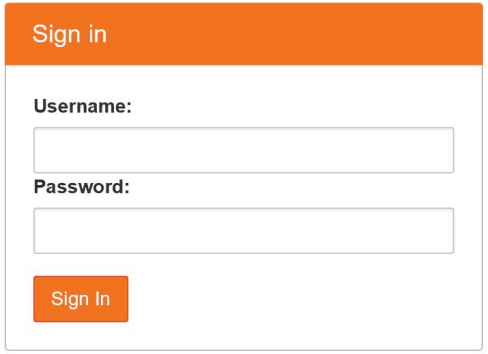
Jupyter Notebook (locally)



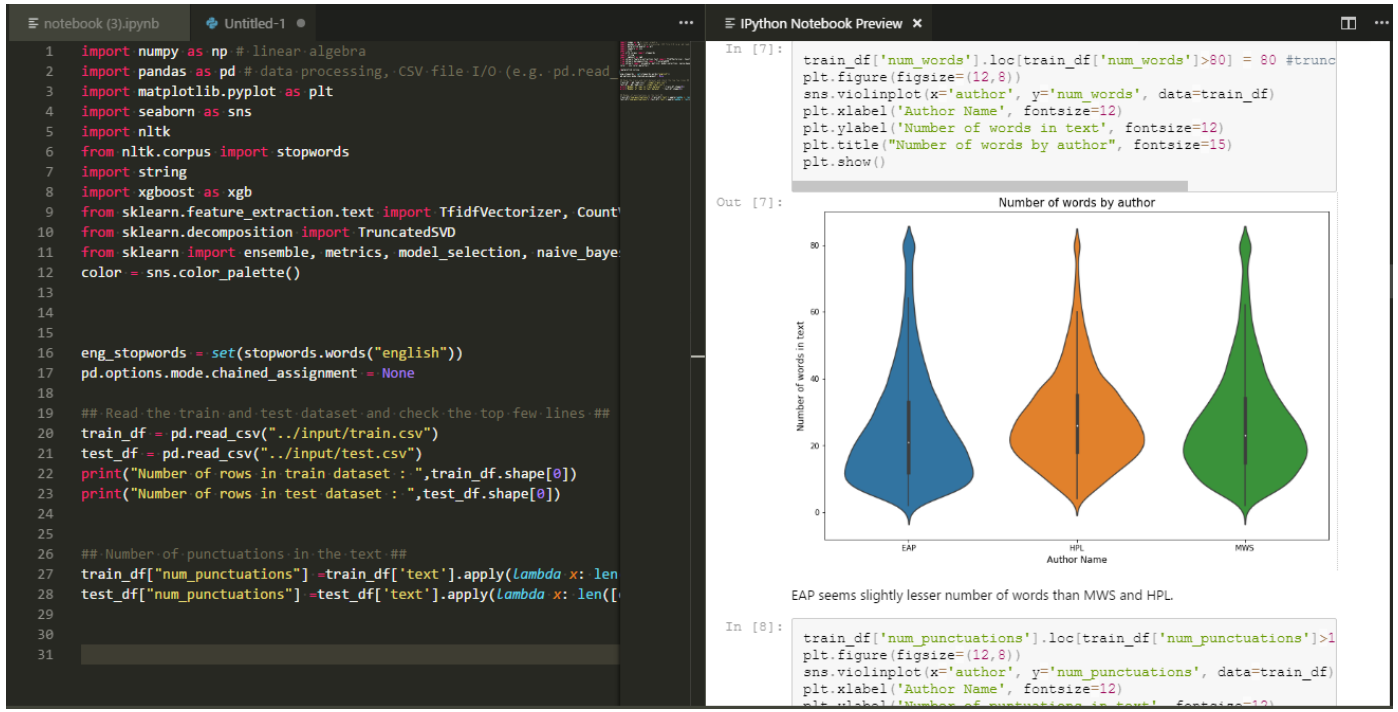
Jupyter-Hub (UCL)



https://jupyter.data-science.rc.ucl.ac.uk/

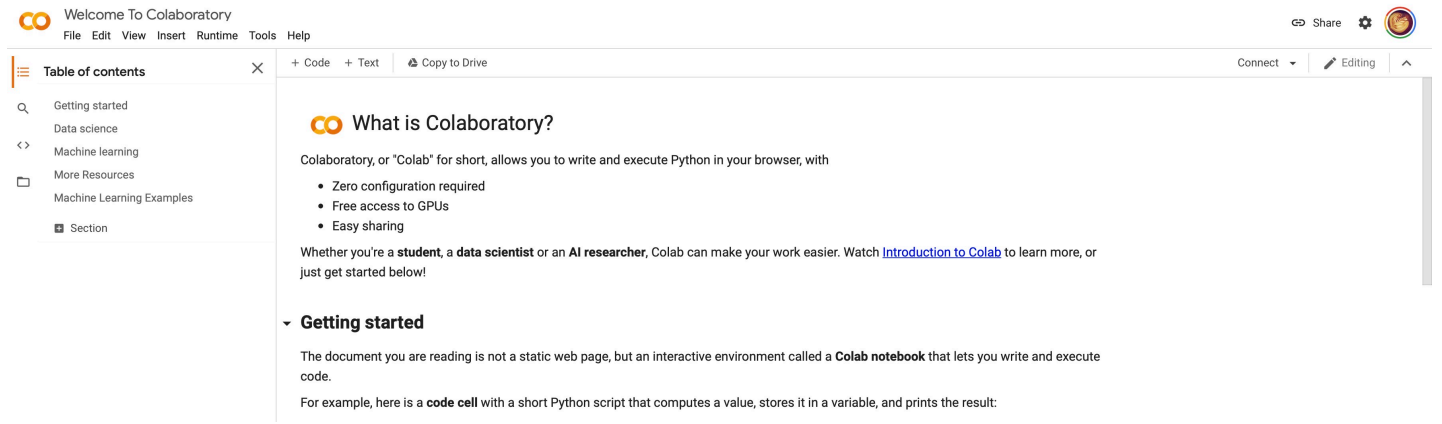


Visual Studio Code (VScode)



Visual Studio Code is a source-code editor made by Microsoft which include more advance use ie. debugging, syntax highlighting, code completion, refactoring, and embedded Git.

Google Colab



Google Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser with free access to GPU. Use constraint for a max time of 12 hours per session.
https://colab.research.google.com/

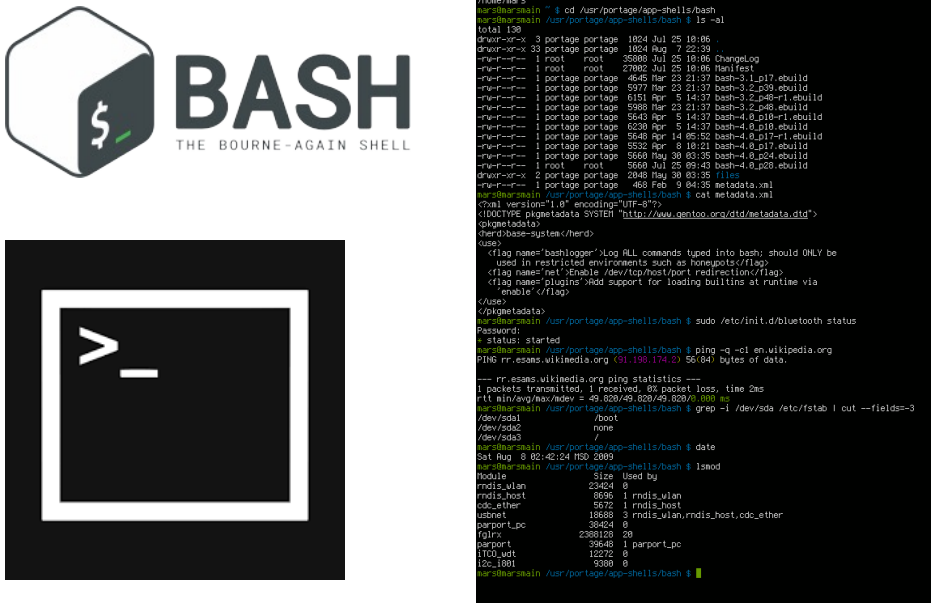
Virtual Environment

A virtual environment is an isolated environment that can help to keep interpreters and dependencies required by different projects separate.

The benefit is you can create and delete these environments without affecting the root one. It also allows you to freeze the environment specifically for the project/module.

Even though you can use the environment in Anaconda, We suggest here to setup a virtual environment specifically for GEOG0051.

Extra Learning: Command Line Shell



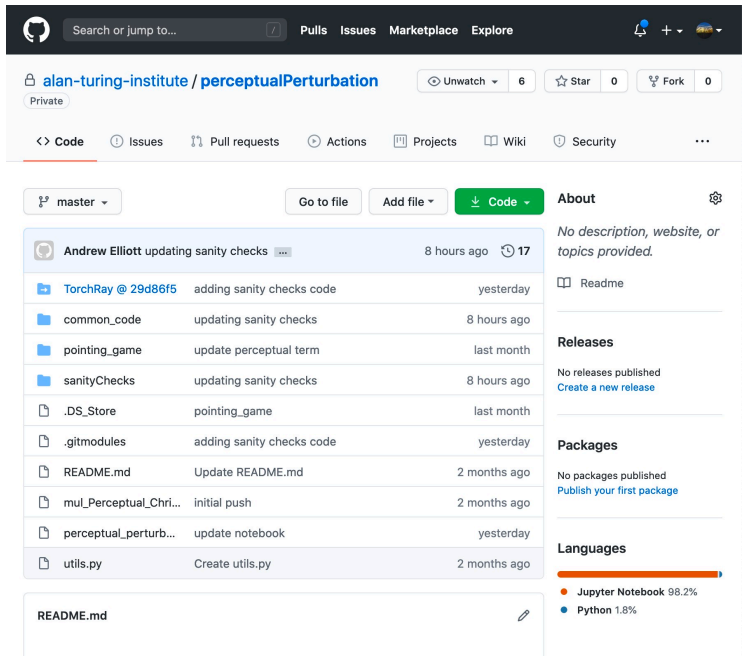
Command Line Shell (CLI)
Command-Prompt, Terminal/BASH

BASH Commands (Terminal)

Print	$echo “Hello World”
Change directory	$cd
List the files in dir	$ls
Move files	$mv file.txt ../
Copy files	$cp file.txt ../
Remove files	$rm file.txt
Remote access	$ssh
Install module in python	$pip install plotly

Extra Learning: Github

GitHub



Github : a hub for hosting, sharing, managing, bug testing, documenting and version controlling code with Git. Effectively this is the library for code.

https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners

Computer Lab 1
Part Two: Installing Anaconda/Jupyter for the first time

Lab 1 Preparation

1. For students that have Anaconda/Miniconda installed, you can skip to step 3.

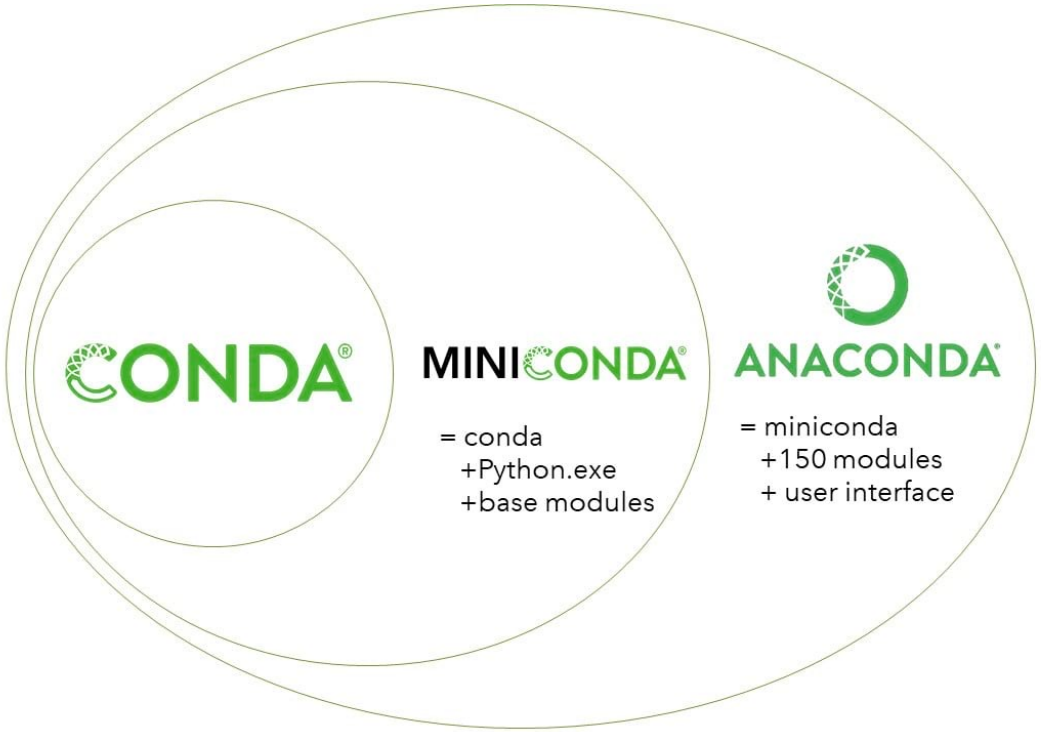
2. Download and Install Anaconda for your operating system from (https://www.anaconda.com/products/individual).

3. In the terminal (Mac), make a new virtual environment for the module=envGEOG0051 with the following script that contains various libraries. (N.B. For Windows users, you can type "Anaconda Prompt" in the Windows search box to launch the terminal to type the code below.)

```
conda create -n envGEOG0051 -c conda-forge geopandas scipy jupyter scikit-learn nltk networkx osmnx
matplotlib seaborn spint contextily spyder folium gitpython gensim scikit-surprise
```

Note: make sure the whole script is copied into terminal when running so you don't miss any packages.

Step 1: Anaconda Installation

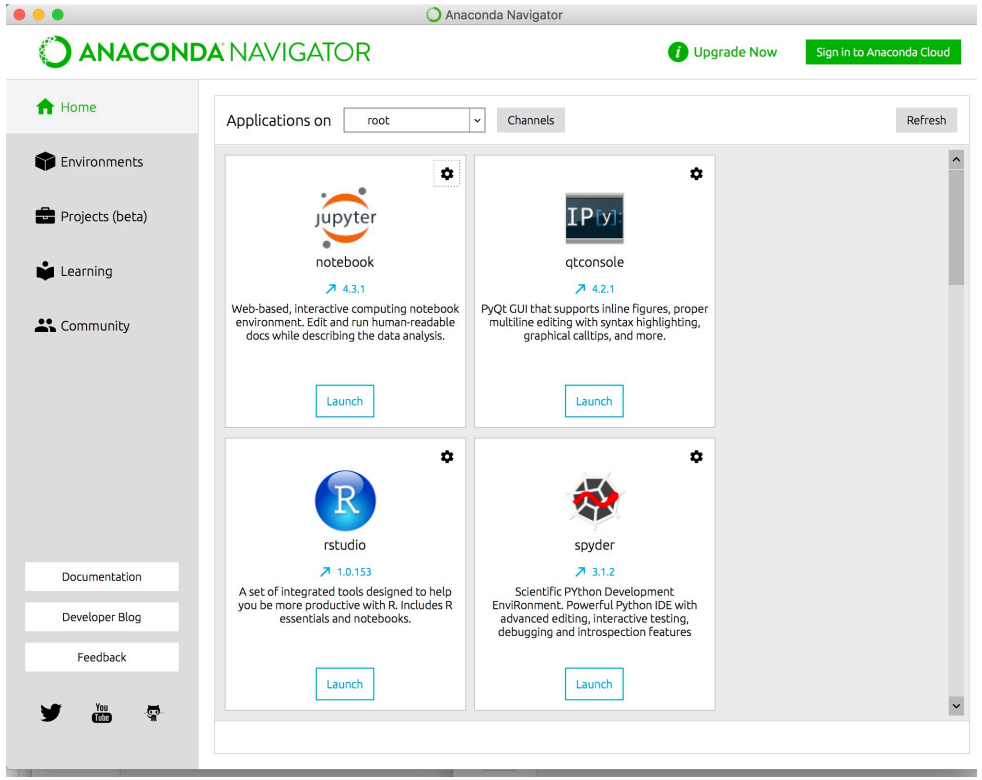


python™

IP[y]:

Download and Install Anaconda for your operating system from (https://www.anaconda.com/products/individual). Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. It is a package manager that lets you install and update your packages.

Step 1: Anaconda Installation



After installing Anaconda, you will have access to the Anaconda-navigator which would allow you to access open Jupyter-notebook that have been installed as part of Anaconda.

However, what we recommend is for you to create a new virtual environment explicitly for the module.

Step 2: Create Virtual Environment

Mac/Linux users: Open a Terminal

Windows users: Open Anaconda prompt

Then run the following command:

conda create -n envGEOG0051 -c conda-forge geopandas scipy jupyter scikit-learn nltk networkx osmnx matplotlib seaborn spint contextily spyder folium gitpython gensim scikit-surprise

Note: make sure the whole script is copied into terminal when running so you don't miss any packages.

(If you get stuck at "solving environment' for a long time, try to run 'conda update conda' to update your conda. Or uninstall your anaconda and reinstall it)

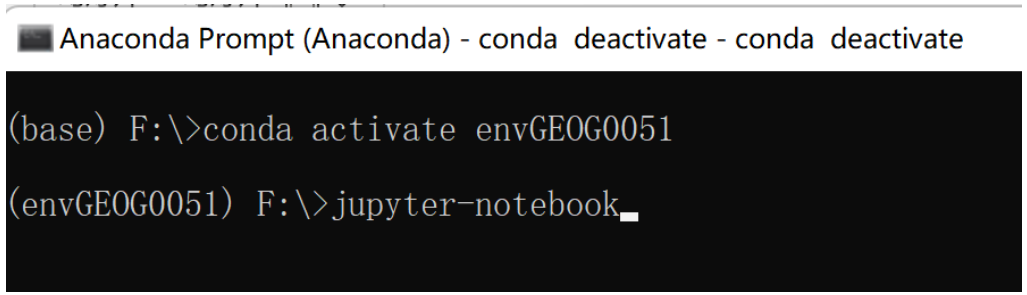
Step 3: Activate Virtual Environment

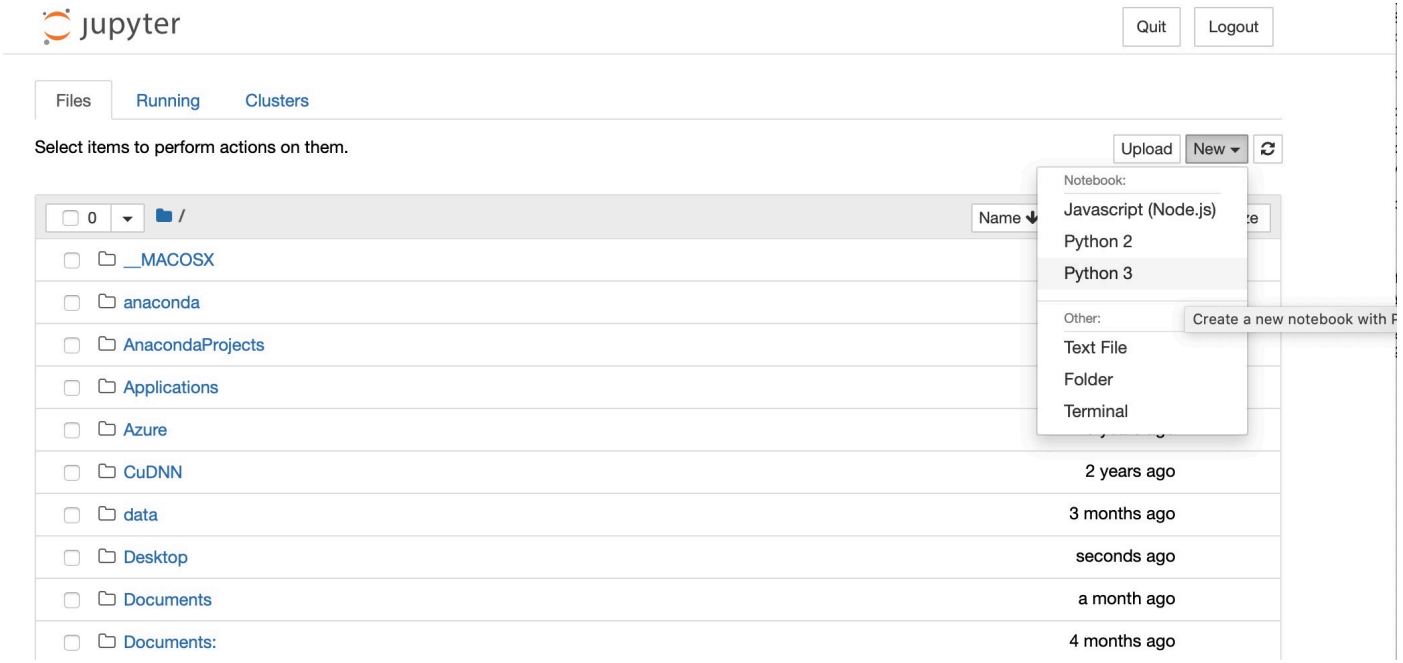
4. Activate the envGEOG0051 virtual environment

```
conda activate envGEOG0051
```

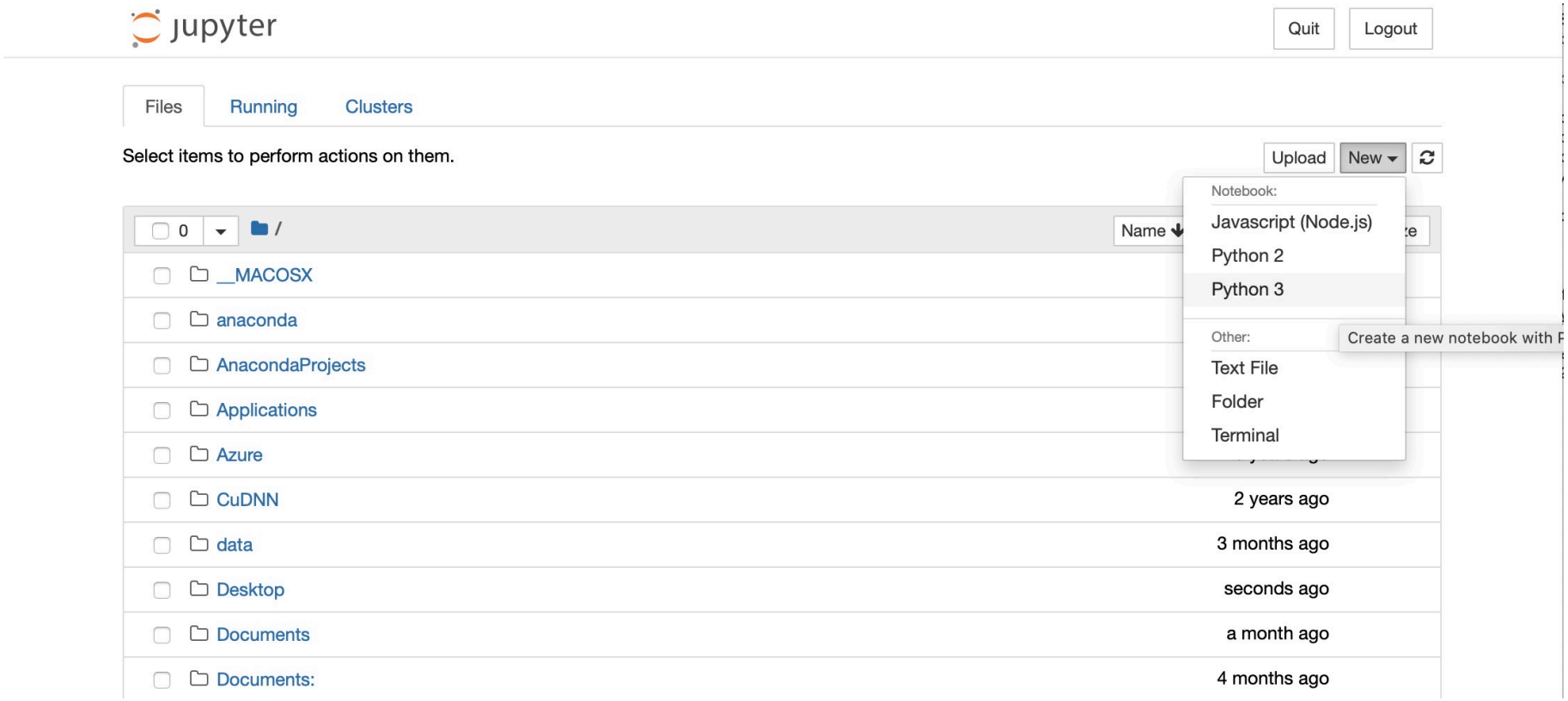
5. Load jupyter-notebook

```
(envGEOG0051)jupyter-notebook
```

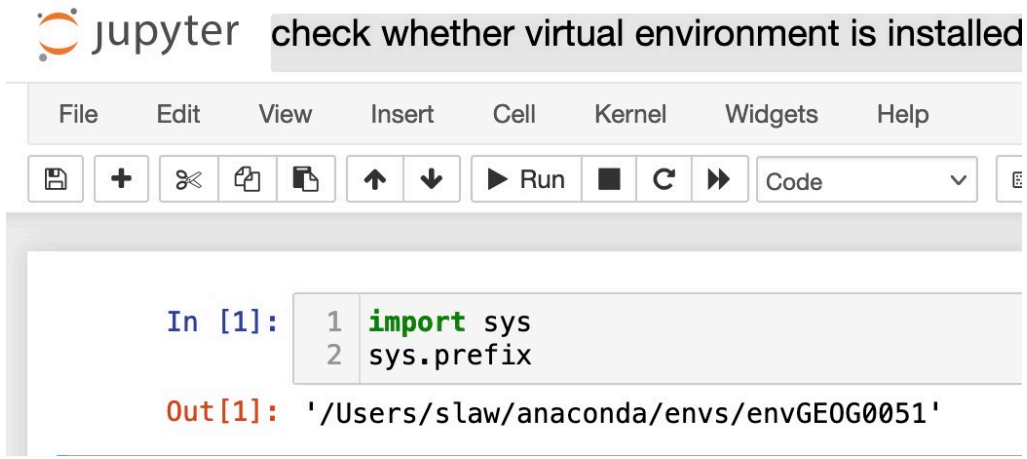




Jupyter Notebook (locally)



Check whether you are using the right env



6. Check whether you are using the right version of jupyter by typing the following

```
Import sys
sys.prefix
```

Step 4: Download Practical Materials

1. Go to Moodle and download the Week1 materials
2. Save materials into an accessible location (e.g. in a folder in Documents called GEOG0051)

Jupyter Notebook Shortcuts

- Shift + Enter run the current cell, select below
- Ctrl + Enter run selected cells
- Alt + Enter run the current cell, insert below
- Ctrl + S save and checkpoint

Stephen Law
Department of Geography | University College London

Computer Lab 1
Part Three: Data-types and Operators in Python

Built-in data-types

Strings: an ordered sequence of characters.

```
>>>Sample_string=“hello world”
```

Integers : Positive/Negative whole numbers

```
>>>sample_int = 3619
```

Floats: Real number with floating point.

```
>>>Sample_float= 1.126
```

Boolean: either True or False

```
>>>Sample_bool = True
```

Composite Built-in data-types

Lists : an ordered mutable collection of one or more items.

```
>>>sample_lists =[1,2,3]
```

Tuples: an ordered immutable collection of one or more items.

```
>>>sample_tuples = ("apple", "banana", "cherry")
```

Dicts : an unordered mutable collection of data in a key:value pair.

```
>>>sample_dict = {“key1”:10}
```

Basic Operators

Operators are constructs which manipulate a set of inputs in Python;

Math Operators

+ (addition)
- (subtraction)
** (power)
* (multiplication)
/ (division)
% (modulus)

Comparison Operators

== : if two values are equal,
!= : if two values are unequal,
< : greater than
> : less than

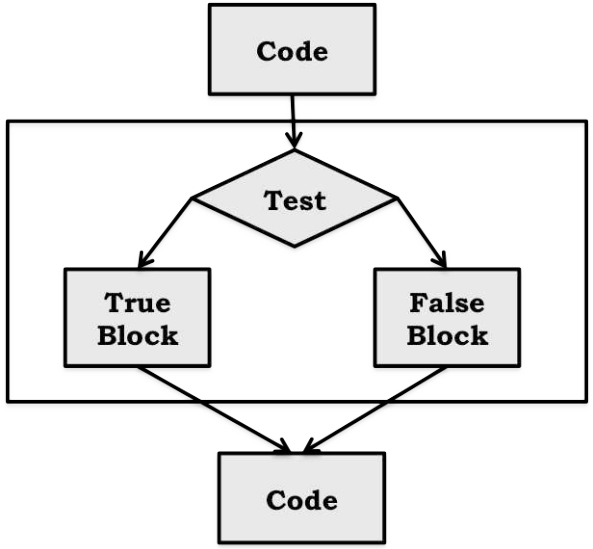
Computer Lab 1
Part Three: Control Structure, Functions and Modules

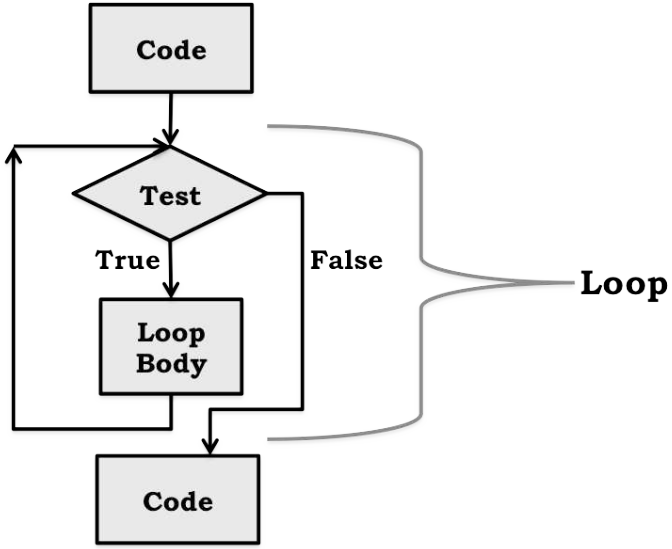
Control Structure

Control Structures is basically a way to specify the flow in programmes.

Sequential, Selection,Repetition

1. if/elif/else
2. for/in
3. while



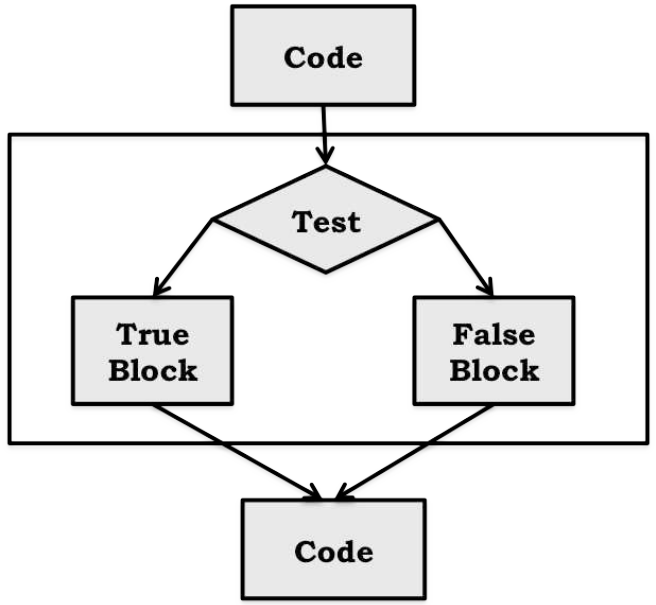


John V. Guttag (2013)

The if Statement

Conditional Statement in Python are handled by IF statement.

```
if expression:
        statement(s)
else:
        statement(s)
```



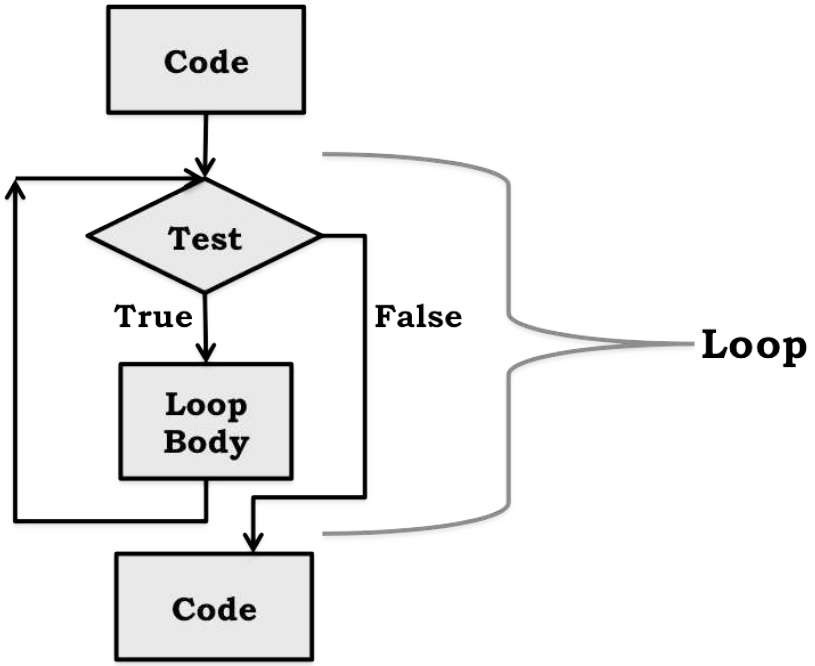

```
x = 10
if x>0:
    print ”The number is positive”
elif<0:
    print “The number is negative”
else:
        print ”The number is zero”

The number is positive
```

John V. Guttag (2013)

The for Statement

For loops is used to iterate over statements for a certain times

```
for i in range(n):
    statement(s)
```




```
for i in range(0,7,1):
    print (i)
```

```
0
1
2
3
4
5
6
```

John V. Guttag (2013)

The while statement

While loops is used to iterate over statements, as long as the condtion is true.

```
while expression:
      statement(s)
```

```
x = 0
while x<5:
      print x
      x = x + 1

0
1
2
3
4
```

Combined Flow and Controls Statements

```
for i in [0,1,2,3]:
    if i!=0:
        print (i)
    else:
        continue
```

For loop

If branch

else branch

```
1
2
3
```

Algorithm in plain-english

For all images

if the images has a dog

run function X

elif the image has a cat

run function Y

else

next

Breaking Code into Parts Functions

Function is a block of code that is executed when it is called.

```
def functionName(parameters):
      statement
      return expression
```

```
def sum2numbers(first, second):
      mysum = first + second
      return mysum

s = sum2numbers(3, 4)
print (s)

[program will print 7]
```

Breaking Code into Parts Functions

https://www.chinasichuanfood.com

1. Transfer the dough to a board.
2. Roll it into a thick wrapper
3. Roll the dough into or round wrapper with around 0.5 cm in thick, keeping remaining dough covered with flour.
4. Fold the large dough wrapper around 5cm wide.
5. Each time after folding up, spread the surface with flour. Cut the remaining part off.
6. Cut the folded dough into thin strips with a sharp and dry knife.

Built-in functions

print(): prints object as a stream of text.

```
>>>print (“hello world”)
```

len() : return the length of an object.

```
>>>sample_lists =[1,2,3]
>>>len(sample_lists)
```

sum() : returns the total of an iterable.

```
>>>sample_list.sum()
```

https://docs.python.org/3.8/library/functions.html

Python Modules

There are modules that needs to be gotten from a package manager/installer such as pip official python packaging manager that links to PyPI, or conda in terminal.

In terminal

pip install plotly

In python

>>>import plotly

https://en.wikipedia.org/wiki/Pip_(package_manager)

Python Modules

Numpy is an open source lib that provides numerical functions on multi-dimensional arrays.
https://numpy.org

```
>> import numpy as np
>> np.mean([1,2,3])
```

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$

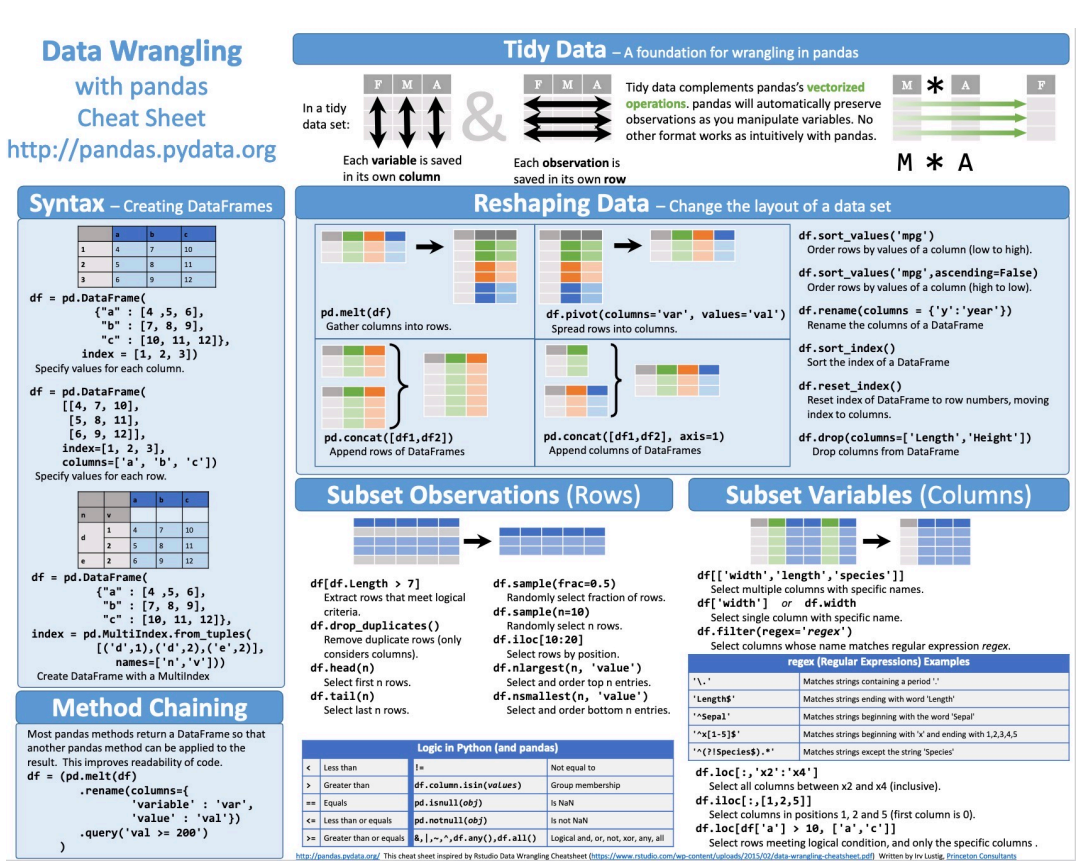
Pandas is an open source lib that provides high-performance data structures for Python.
https://pandas.pydata.org

```
>> import pandas as pd
>> pd.DataFrame([1,,2,3])
```

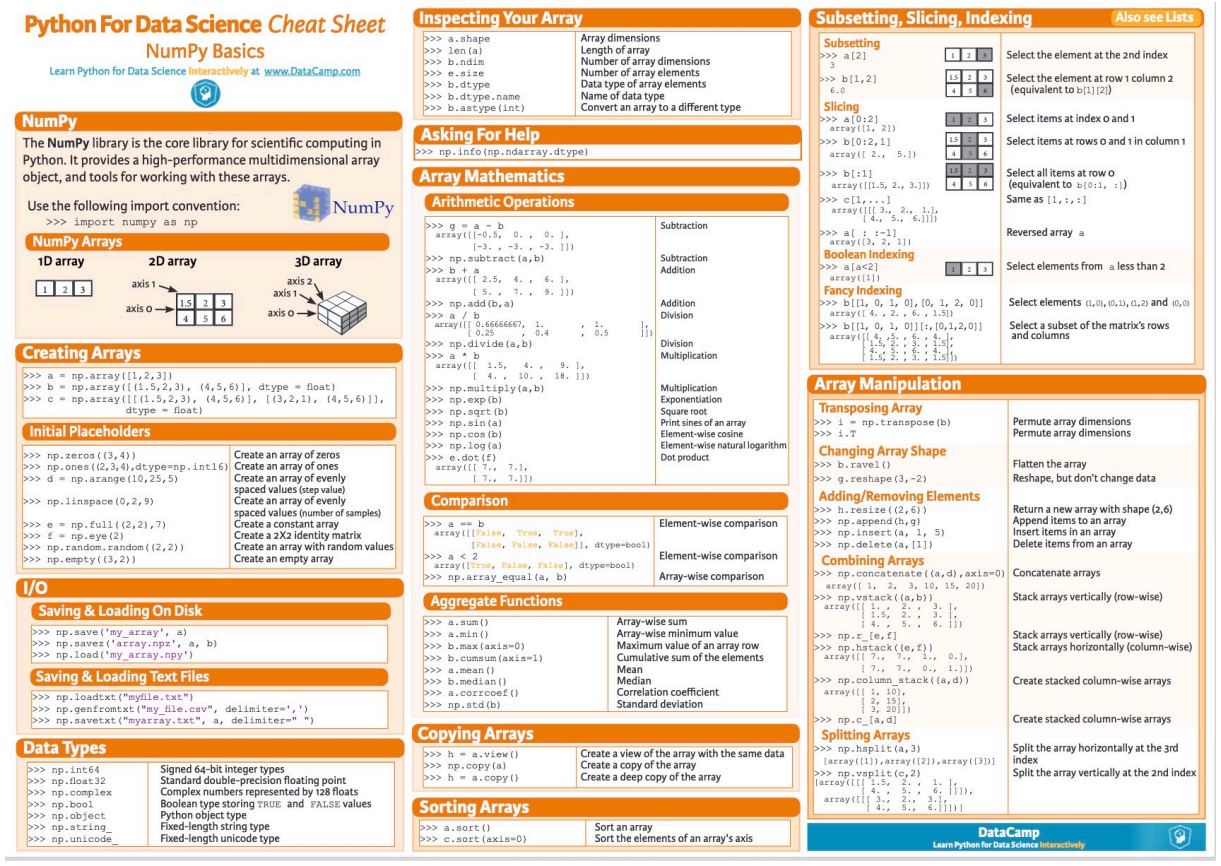
Matplotlib is an open source plotting lib that produces publication quality figures.
https://matplotlib.org/

```
>> import matplotlib.pyplot as plt
>> plt.hist([1,1,1,2,2,3])
```

References



https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf



https://assets.datacamp.com/blog_assets/Numpy_Python_Cheat_Sheet.pdf

References

Python Official Tutorial https://docs.python.org/3/tutorial/index.html

w3 schools https://www.w3schools.com/python/

John V. Guttag (2013). Introduction to Computation and Programming Using Python. MIT Press 2013. Chapter 2&3

McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc."

VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc.".