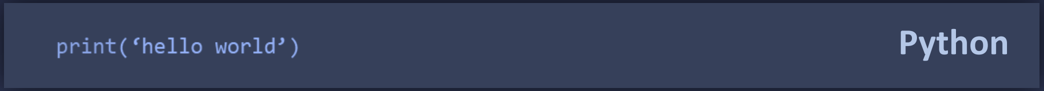
Introduction to Programming in Python

BMEG 591M Workshop Series

Shekoofeh (Shek) Azizi, UBC, November 2017

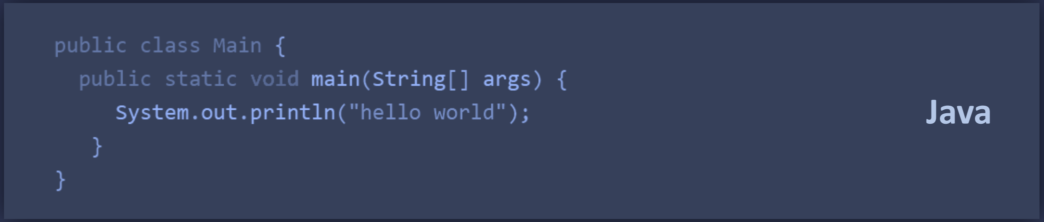
Why Python?

- The most popular introductory teaching language in U.S. universities.
- The 4th most popular language according to an IEEE survey, behind old classics Java, C, and C++.



```
print(‘hello world’)
```

Python



```
public class Main {
  public static void main(String[] args) {
     System.out.println("hello world");
   }
}
```

Java

Why Python?

- Open source general-purpose language.
- Object oriented, procedural, functional.
- The community provides many introductory resources.
- Device programming such as *Raspberry Pi*

- Downloads: http://www.python.org
- Documentation: http://www.python.org/doc/
- Community: https://www.python.org/community/

What I am going to cover?

- Setup
- Input and Output
- Basic Data Types
- Containers
- Loops (For and While)
- Conditions
- Functions
- Modules and Packages

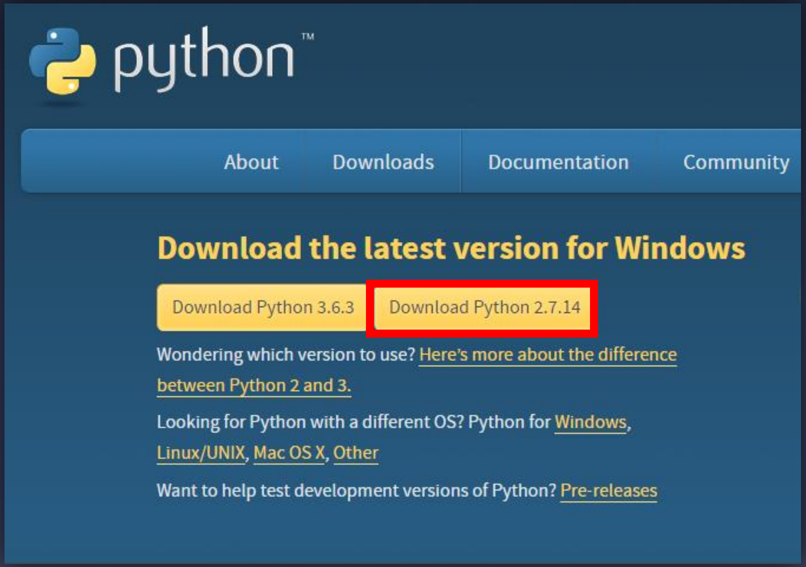
Our Shared Space

- Slides
- Codes
- Assignments

https://goo.gl/XY72t6

Installing Python

- Python comes pre-installed with Mac OS X and Linux.
- https://www.python.org/downloads/
- You might not have to do anything!
- 2.7.x / 3.x ???



Download the latest version for Windows

Wondering which version to use? Here's more about the difference between Python 2 and 3.

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

Want to help test development versions of Python? Pre-releases

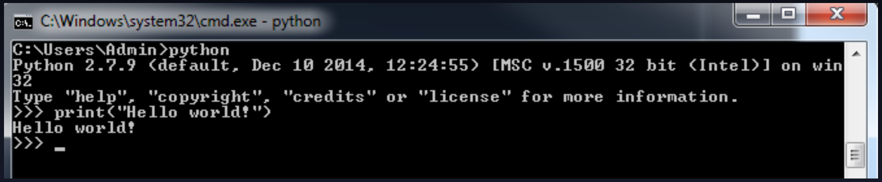
Using Python

- Terminal or Command Prompt

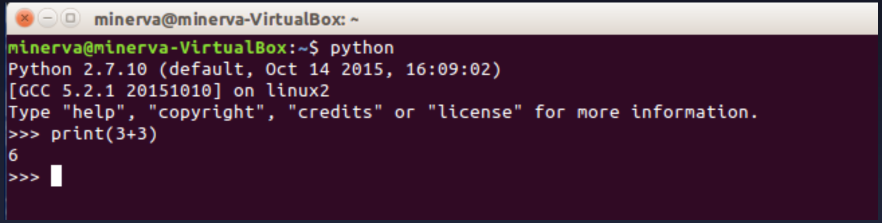
- Jupyter/IPython

- Python IDE: PyCharm

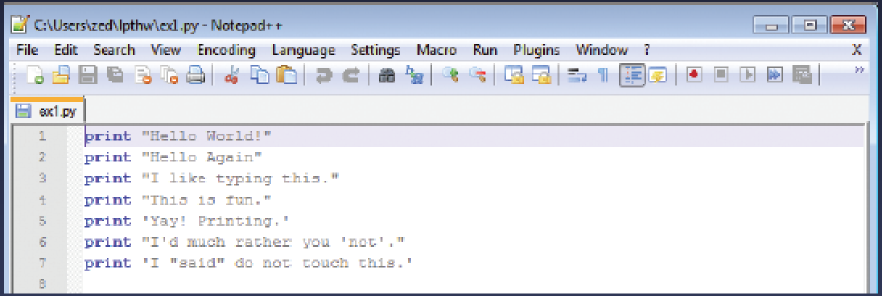
Using Python: Terminal/Command



```
C:\Users\Admin>python
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello world!")
Hello world!
>>> _
```



```
minerva@minerva-VirtualBox:~$ python
Python 2.7.10 (default, Oct 14 2015, 16:09:02)
[GCC 5.2.1 20151010] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print(3+3)
6
>>> 
```


C:\Users\zed\lpthw\ex1.py - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
ex1.py

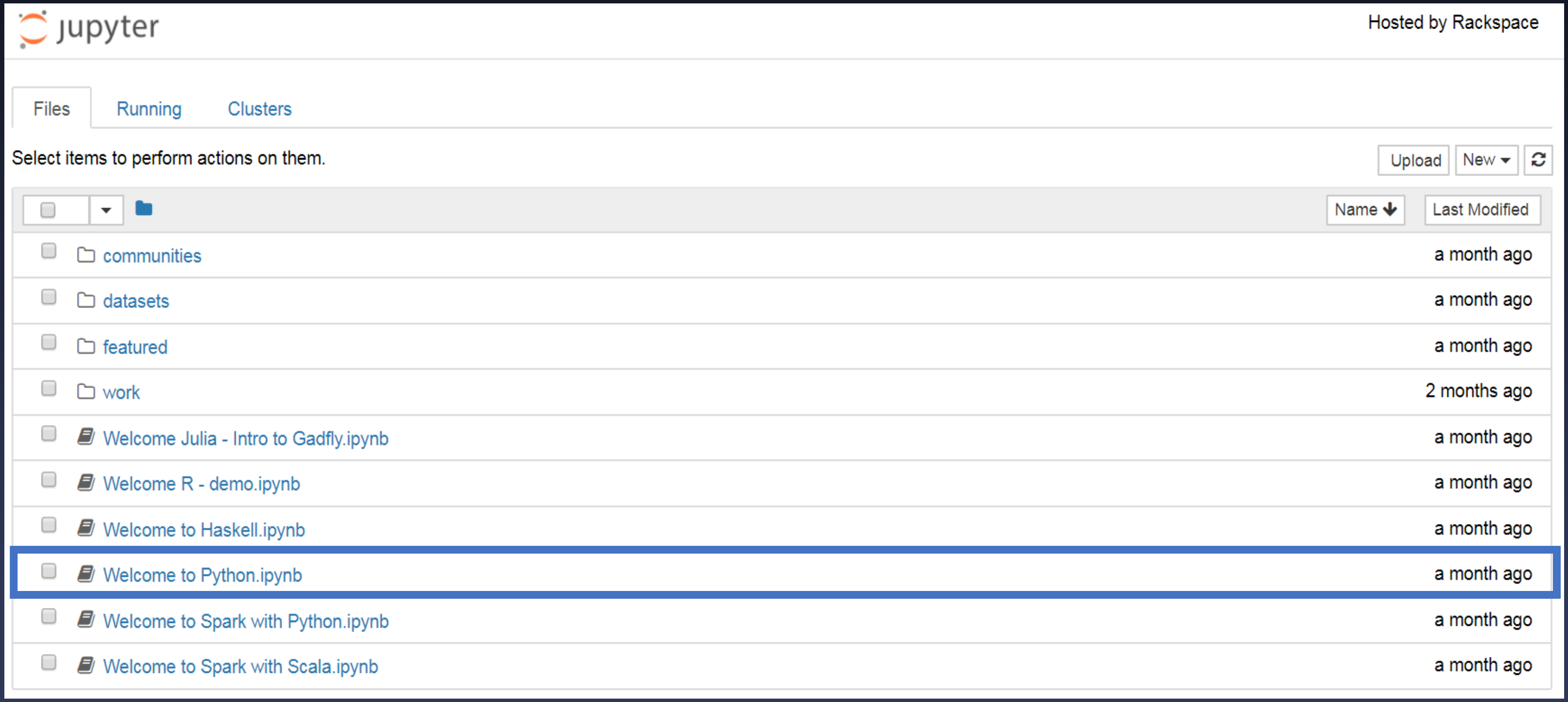
```
print "Hello World!"
print "Hello Again"
print "I like typing this."
print "This is fun."
print 'Yay! Printing.'
print "I'd much rather you 'not'."
print 'I "said" do not touch this.'

```

python filename.py

- Python prompts with '>>>'.
- To exit Python:
  - CTRL-D
  - exit()

Using Python: Jupyter

- https://try.jupyter.org/

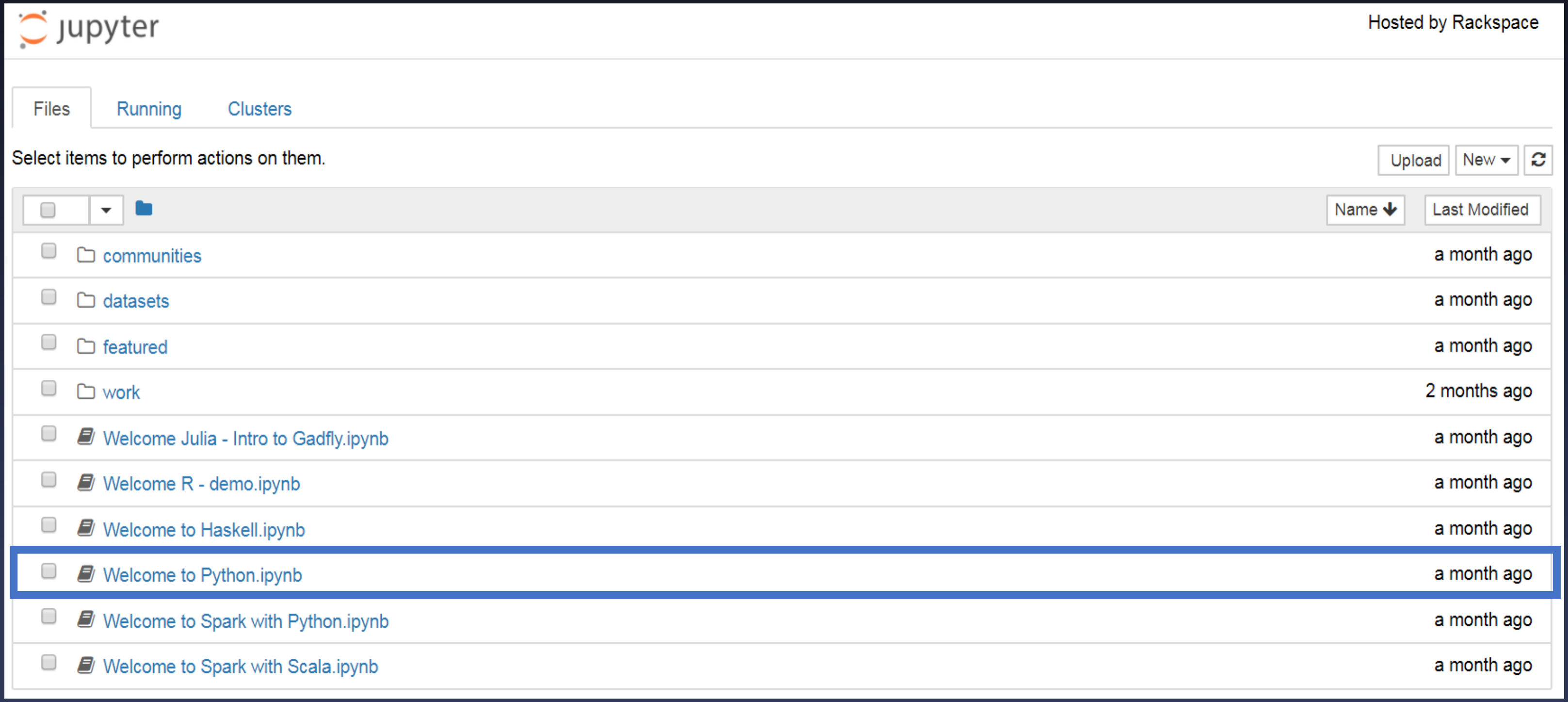


Select items to perform actions on them.

□ communities	a month ago
□ datasets	a month ago
□ featured	a month ago
□ work	2 months ago
□ Welcome Julia - Intro to Gadfly.ipynb	a month ago
□ Welcome R - demo.ipynb	a month ago
□ Welcome to Haskell.ipynb	a month ago
□ Welcome to Python.ipynb	a month ago
□ Welcome to Spark with Python.ipynb	a month ago
□ Welcome to Spark with Scala.ipynb	a month ago

Using Python: Jupyter

- https://try.jupyter.org/

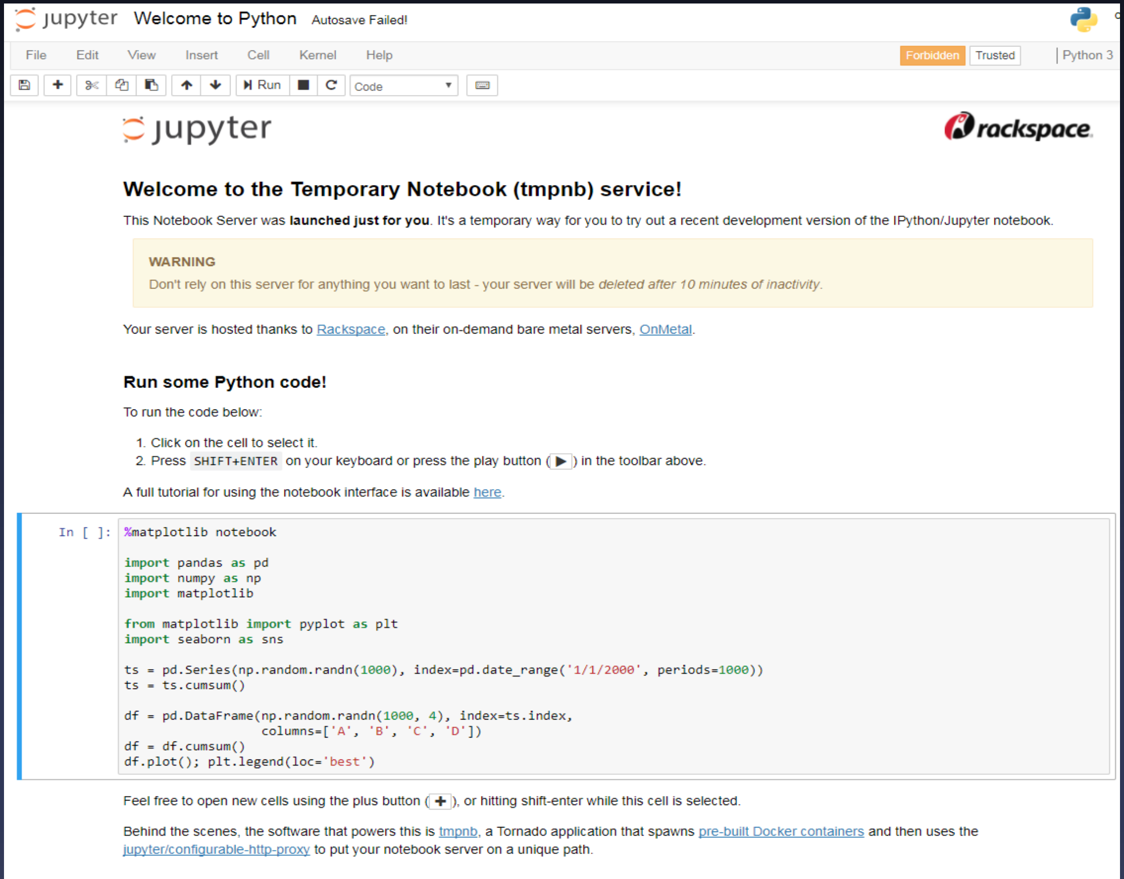


Select items to perform actions on them.

communities	a month ago
datasets	a month ago
featured	a month ago
work	2 months ago
Welcome Julia - Intro to Gadfly.ipynb	a month ago
Welcome R - demo.ipynb	a month ago
Welcome to Haskell.ipynb	a month ago
Welcome to Python.ipynb	a month ago
Welcome to Spark with Python.ipynb	a month ago
Welcome to Spark with Scala.ipynb	a month ago

Using Python: Jupyter

- https://try.jupyter.org/



Welcome to the Temporary Notebook (tmpnb) service!

This Notebook Server was **launched just for you**. It's a temporary way for you to try out a recent development version of the IPython/Jupyter notebook.

> WARNING
>
> Don't rely on this server for anything you want to last - your server will be *deleted after 10 minutes of inactivity*.

Your server is hosted thanks to Rackspace, on their on-demand bare metal servers, OnMetal.

Run some Python code!

To run the code below:

1. Click on the cell to select it.
2. Press `SHIFT+ENTER` on your keyboard or press the play button (▶) in the toolbar above.

A full tutorial for using the notebook interface is available here.

In [ ]:

```
%matplotlib notebook

import pandas as pd
import numpy as np
import matplotlib

from matplotlib import pyplot as plt
import seaborn as sns

ts = pd.Series(np.random.randn(1000), index=pd.date_range('1/1/2000', periods=1000))
ts = ts.cumsum()

df = pd.DataFrame(np.random.randn(1000, 4), index=ts.index,
                  columns=['A', 'B', 'C', 'D'])
df = df.cumsum()
df.plot(); plt.legend(loc='best')
```

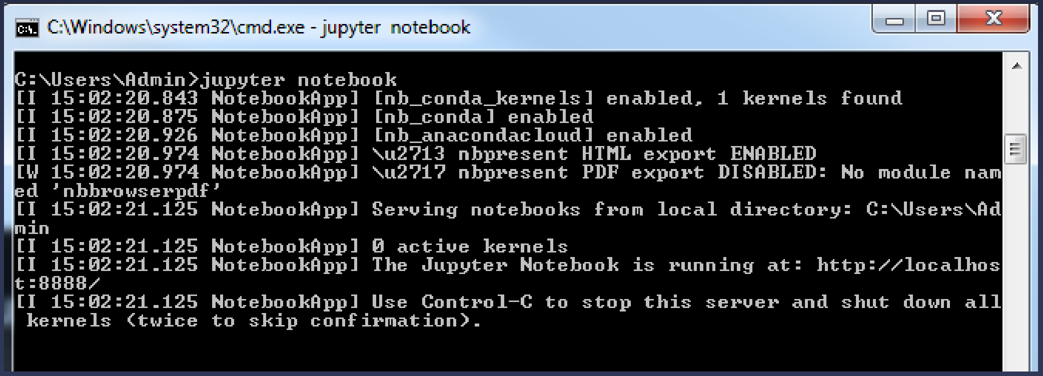
Feel free to open new cells using the plus button (✚), or hitting shift-enter while this cell is selected.

Behind the scenes, the software that powers this is tmpnb, a Tornado application that spawns pre-built Docker containers and then uses the jupyter/configurable-http-proxy to put your notebook server on a unique path.

Using Python: Jupyter

- http://jupyter.org/install.html
- Prerequisite: Python
- To run the notebook, run the following command at the Terminal (Mac/Linux) or Command Prompt (Windows):

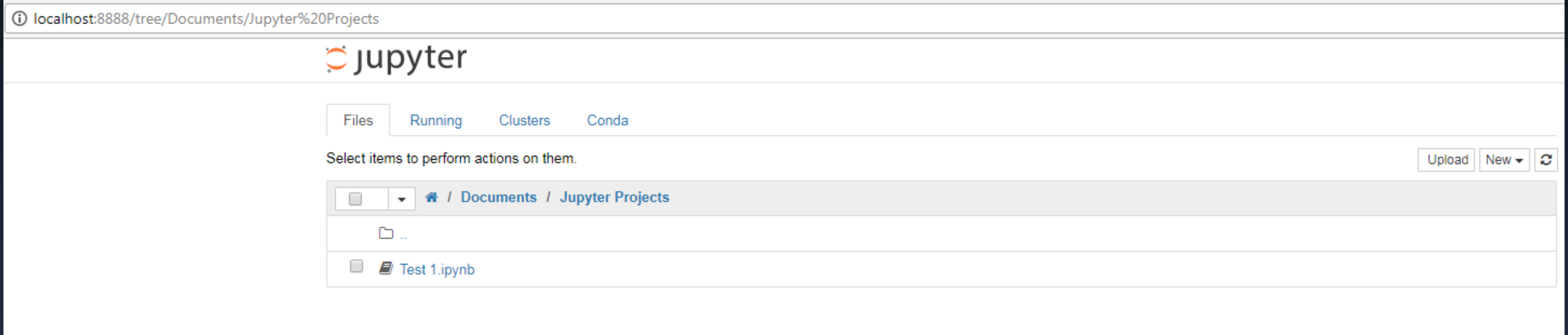
```
jupyter  notebook
```



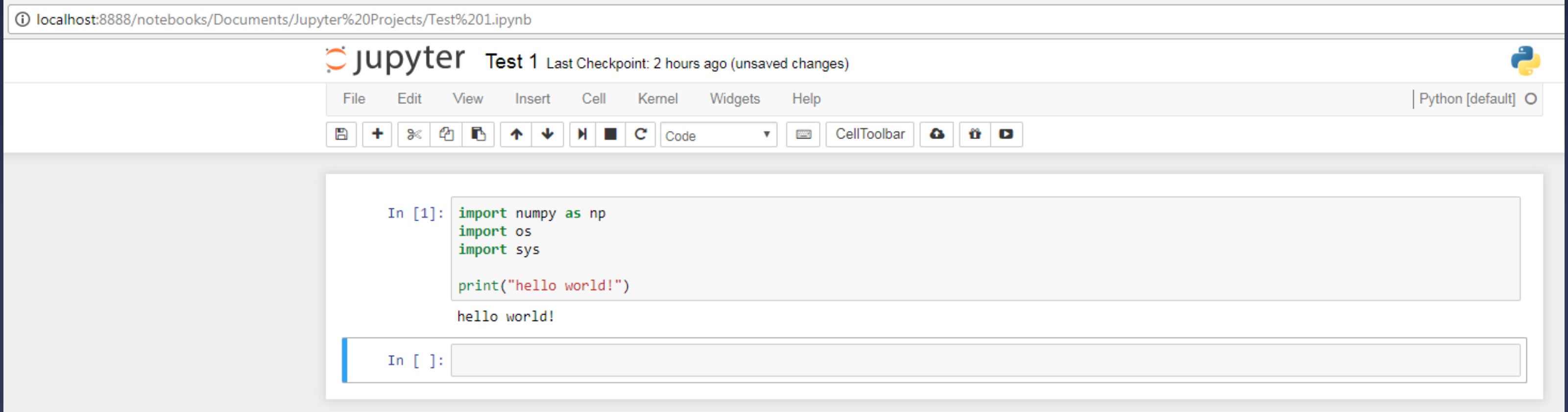
```
C:\Users\Admin>jupyter notebook
[I 15:02:20.843 NotebookApp] [nb_conda_kernels] enabled, 1 kernels found
[I 15:02:20.875 NotebookApp] [nb_conda] enabled
[I 15:02:20.926 NotebookApp] [nb_anacondacloud] enabled
[I 15:02:20.974 NotebookApp] \u2713 nbpresent HTML export ENABLED
[W 15:02:20.974 NotebookApp] \u2717 nbpresent PDF export DISABLED: No module nam
ed 'nbbrowserpdf'
[I 15:02:21.125 NotebookApp] Serving notebooks from local directory: C:\Users\Ad
min
[I 15:02:21.125 NotebookApp] 0 active kernels
[I 15:02:21.125 NotebookApp] The Jupyter Notebook is running at: http://localhos
t:8888/
[I 15:02:21.125 NotebookApp] Use Control-C to stop this server and shut down all
 kernels (twice to skip confirmation).
```

Using Python: Jupyter

- http://jupyter.org/install.html


localhost:8888/tree/Documents/Jupyter%20Projects
jupyter
Files
Running
Clusters
Conda
Upload
New ▾
Documents / Jupyter Projects
..
Test 1.ipynb

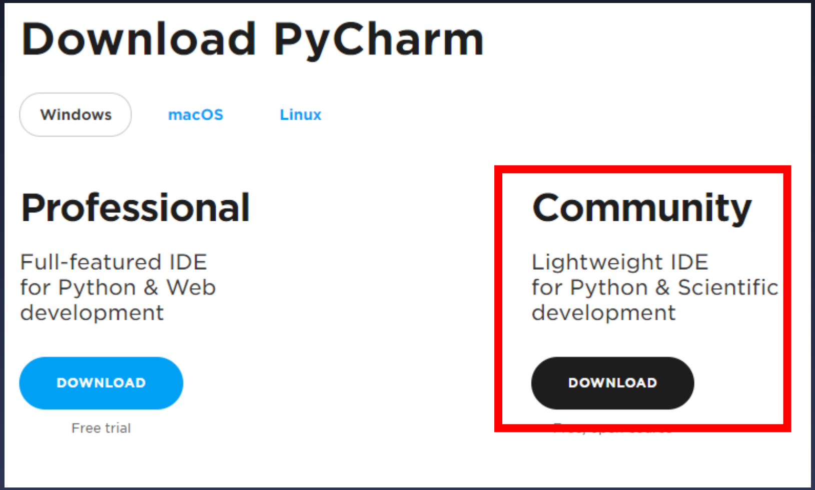
Select items to perform actions on them.


localhost:8888/notebooks/Documents/Jupyter%20Projects/Test%201.ipynb
jupyter
Test 1
Last Checkpoint: 2 hours ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Python [default]
Code
CellToolbar

```
In [1]: import numpy as np
        import os
        import sys

        print("hello world!")

        hello world!

In [ ]:
```

Using Python: PyCharm

- https://www.jetbrains.com/pycharm/download



Download PyCharm

Windows | macOS | Linux

Professional

Full-featured IDE for Python & Web development

DOWNLOAD

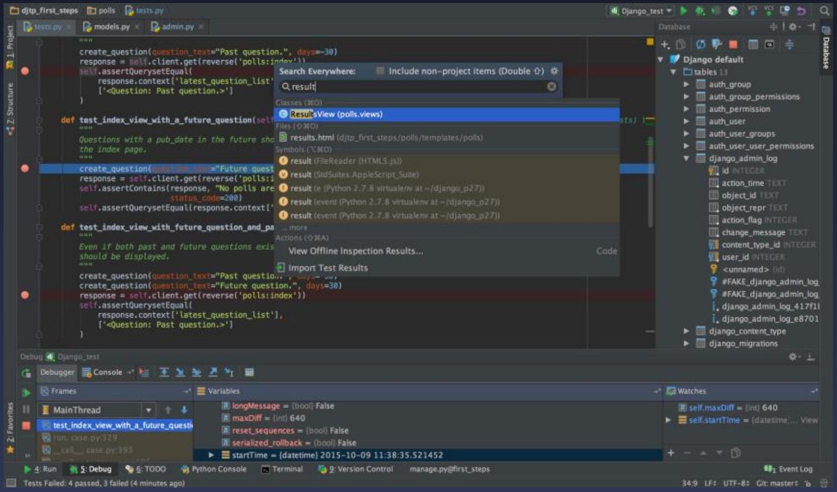
Free trial

Community

Lightweight IDE for Python & Scientific development

DOWNLOAD

Free, open-source



```
        """
        create_question(question_text="Past question.", days=-30)
        response = self.client.get(reverse('polls:index'))
        self.assertQuerysetEqual(
            response.context['latest_question_list'
            ['<Question: Past question.>']
        )

    def test_index_view_with_a_future_question(self
        """
        Questions with a pub_date in the future sho
        the index page.
        """
        create_question(question_text="Future quest
        response = self.client.get(reverse('polls:i
        self.assertContains(response, "No polls are
                            status_code=200)
        self.assertQuerysetEqual(response.context['

    def test_index_view_with_future_question_and_pa
        """
        Even if both past and future questions exis
        should be displayed.
        """
        create_question(question_text="Past questio
        create_question(question_text="Future question.", days=30)
        response = self.client.get(reverse('polls:index'))
        self.assertQuerysetEqual(
            response.context['latest_question_list'],
            ['<Question: Past question.>']
        )
```

Search Everywhere: ☐ Include non-project items (Double ⇧)

result

Classes (⌘O)
- ResultsView (polls.views)

Files (⇧⌘O)
- results.html (djtp_first_steps/polls/templates/polls)

Symbols (⌥⌘O)
- result (FileReader (HTML5.js))
- result (StdSuites.AppleScript_Suite)
- result (e (Python 2.7.8 virtualenv at ~/django_p27))
- result (event (Python 2.7.8 virtualenv at ~/django_p27))
- result (event (Python 2.7.8 virtualenv at ~/django_p27))
- ... more

Actions (⇧⌘A)
- View Offline Inspection Results... Code
- Import Test Results

Debug Django_test

```
MainThread
test_index_view_with_a_future_questi
run, case.py:329
__call__, case.py:393
```

```
longMessage = {bool} False
maxDiff = {int} 640
reset_sequences = {bool} False
serialized_rollback = {bool} False
startTime = {datetime} 2015-10-09 11:38:35.521452
```

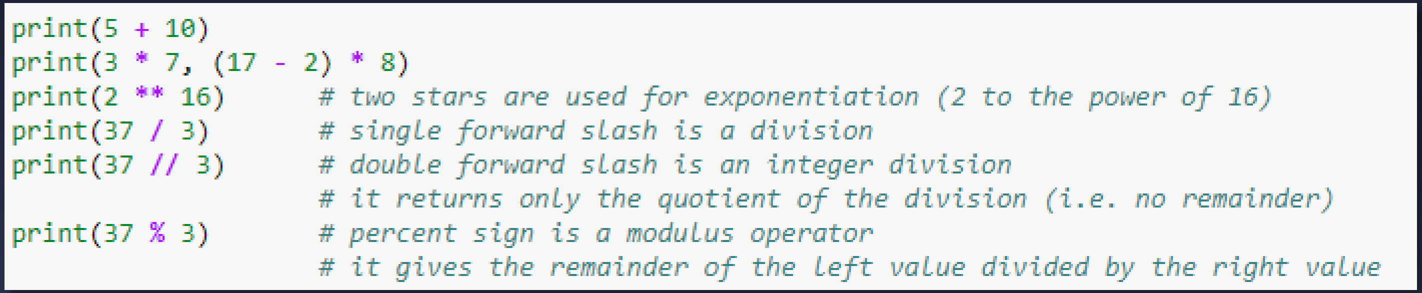
```
self.maxDiff = {int} 640
self.startTime = {datetime}... View
```

Tests Failed: 4 passed, 3 failed (4 minutes ago)

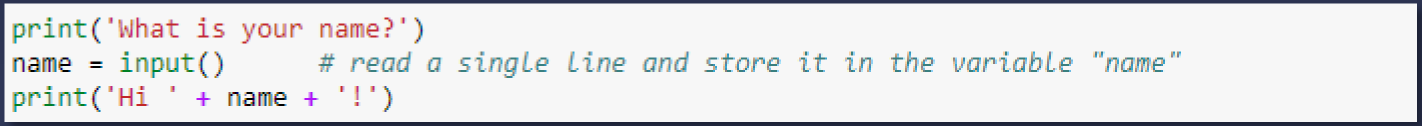
34:9 LF: UTF-8: Git: master:

How to read and write in Python?

- Function:
  - print()
  - input()



```python
print(5 + 10)
print(3 * 7, (17 - 2) * 8)
print(2 ** 16)      # two stars are used for exponentiation (2 to the power of 16)
print(37 / 3)       # single forward slash is a division
print(37 // 3)      # double forward slash is an integer division
                    # it returns only the quotient of the division (i.e. no remainder)
print(37 % 3)       # percent sign is a modulus operator
                    # it gives the remainder of the left value divided by the right value
```



```python
print('What is your name?')
name = input()      # read a single line and store it in the variable "name"
print('Hi ' + name + '!')
```

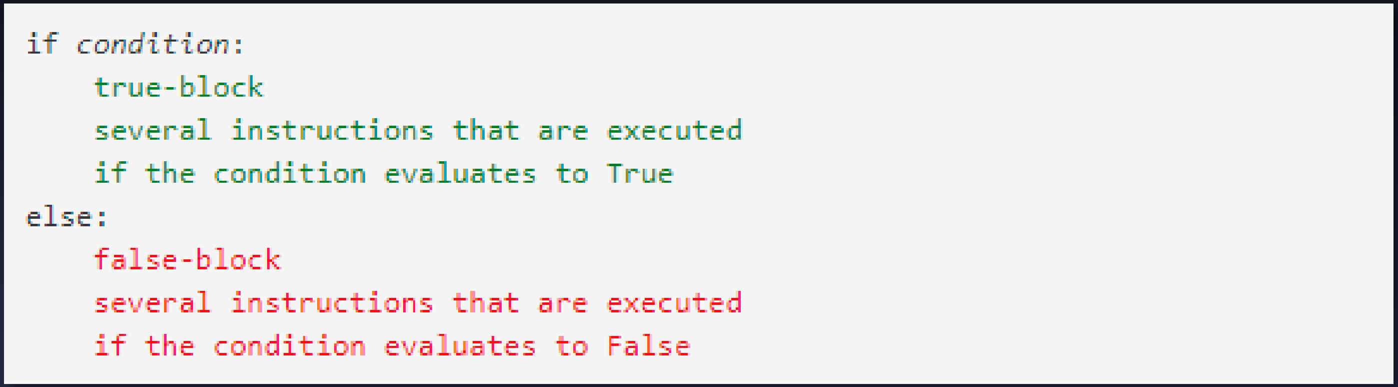
Input and Print.ipynb

Basic Data Types

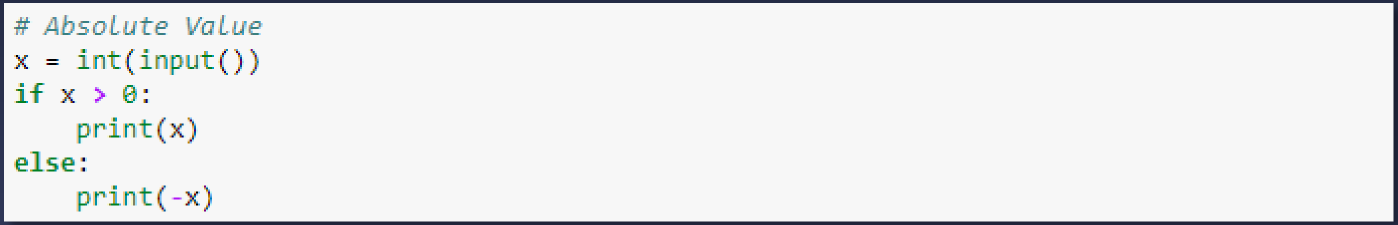
- Number
  - Integers and floats work as you would expect from other languages.
- Strings
  - String objects have a bunch of useful methods.
- Booleans
  - Python implements all of the usual operators for Boolean logic, but uses English words rather than symbols (&&, ||, etc.)
  - Conditions: if, then, else

Basic Data Types.ipynb

Conditions: if, then, else



```
if condition:
    true-block
    several instructions that are executed
    if the condition evaluates to True
else:
    false-block
    several instructions that are executed
    if the condition evaluates to False
```



```
# Absolute Value
x = int(input())
if x > 0:
    print(x)
else:
    print(-x)
```

Conditions.ipynb

Conditions: Comparison Operators

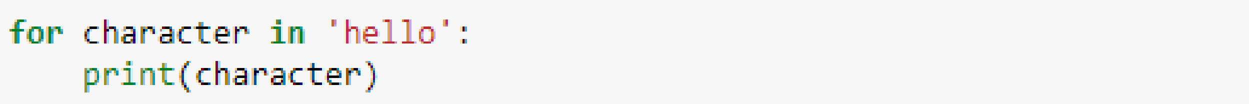
Operator	What it means
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Booleans: Python implements all of the usual operators for Boolean logic, but uses English words rather than symbols (&&, ||, etc.).

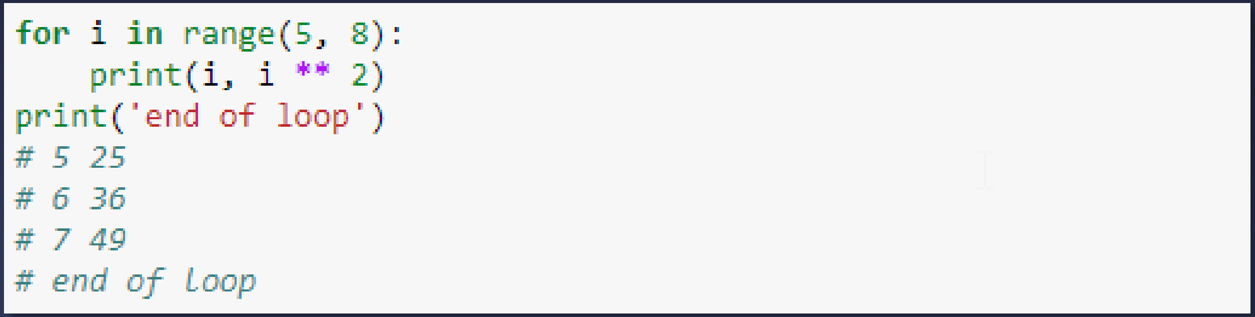
Operator	What it means	What it looks like
and	True if both are true	x and y
or	True if at least one is true	x or y
not	True only if false	not x

For loop with a range

- range(min_value, max_value)
  - generates a sequence with numbers min_value, min_value + 1, ..., max_value – 1.
  - The last number is not included.



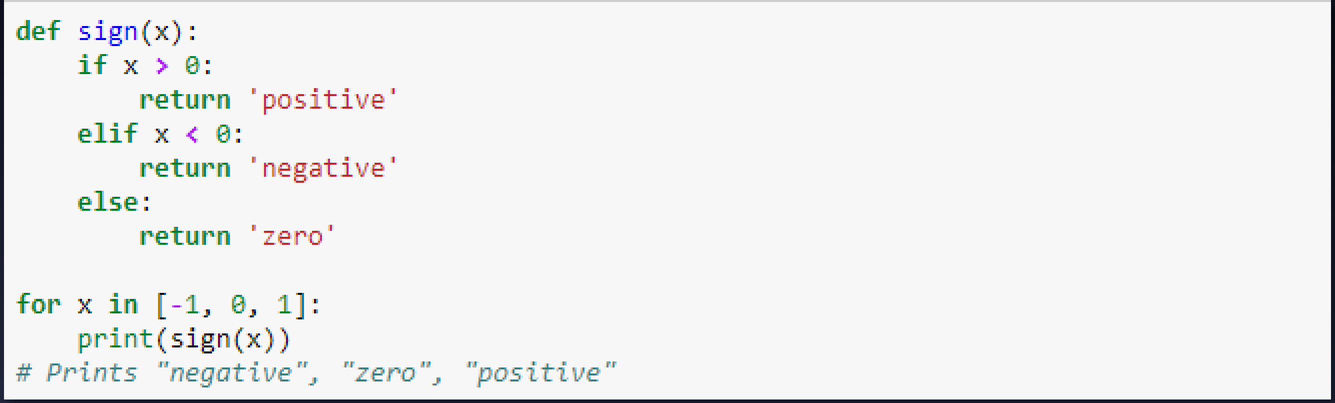
```python
for character in 'hello':
    print(character)
```



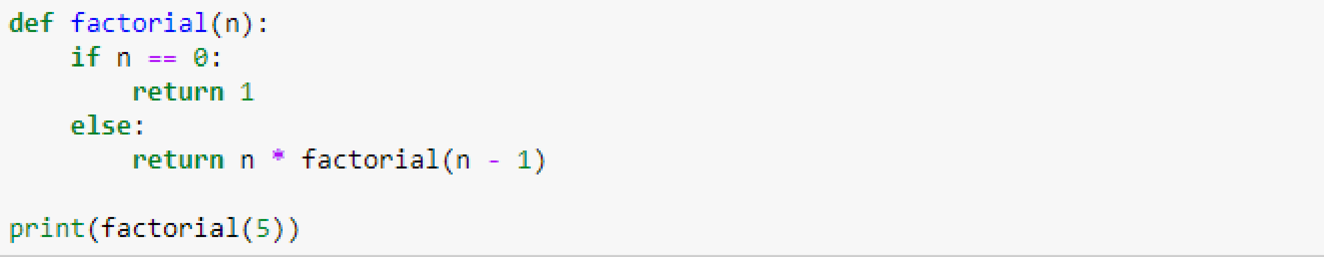
```python
for i in range(5, 8):
    print(i, i ** 2)
print('end of loop')
# 5 25
# 6 36
# 7 49
# end of loop
```

For loop with range.ipynb

Functions



```python
def sign(x):
    if x > 0:
        return 'positive'
    elif x < 0:
        return 'negative'
    else:
        return 'zero'

for x in [-1, 0, 1]:
    print(sign(x))
# Prints "negative", "zero", "positive"
```



```python
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

print(factorial(5))
```

Functions.ipynb

Exercise 1: Password Generator ☺

- Write a password generator in Python.
- Strong passwords have a mix of lowercase letters, uppercase letters, numbers, and symbols.
- Ask user about the password length
- Condition: don't accept length shorter than 6.
- At least one uppercase, lowercase and number.
- Happy Coding!

PasswordGenerator.ipynb

Containers

- Lists
  - A list is the Python equivalent of an array, but is resizable and can contain elements of different types.
  - Slicing: access to sub-lists.
  - loops
- Dictionaries
- Sets
- Tuples

Containers.ipynb

Containers

- Lists
  - A list is the Python equivalent of an array, but is resizable and can contain elements of different types.
  - Slicing: access to sub-lists.
  - Loops
  - Comprehension: An elegant way to define and create one type of data from another
- Dictionaries
  - A dictionary stores (key, value) pairs, similar to a Map in Java or an object in JavaScript.
- Sets
  - A set is an unordered collection of distinct elements. As a simple example, consider the following.
- Tuples
  - A tuple is an (immutable) ordered list of values

Containers.ipynb

List of Built-in Functions

Operation	**Result**
x + y	sum of *x* and *y*
x - y	difference of *x* and *y*
x * y	product of *x* and *y*
x / y	quotient of *x* and *y*
x // y	floored quotient of *x* and *y*
x % y	remainder of x / y
-x	*x* negated
+x	*x* unchanged
abs(x)	absolute value or magnitude of *x*
int(x)	*x* converted to integer
float(x)	*x* converted to floating point
complex(re, im)	a complex number with real part *re*, imaginary part *im*. *im* defaults to zero.
c.conjugate()	conjugate of the complex number *c*
divmod(x, y)	the pair (x // y, x % y)
pow(x, y)	*x* to the power *y*
x ** y	*x* to the power *y*

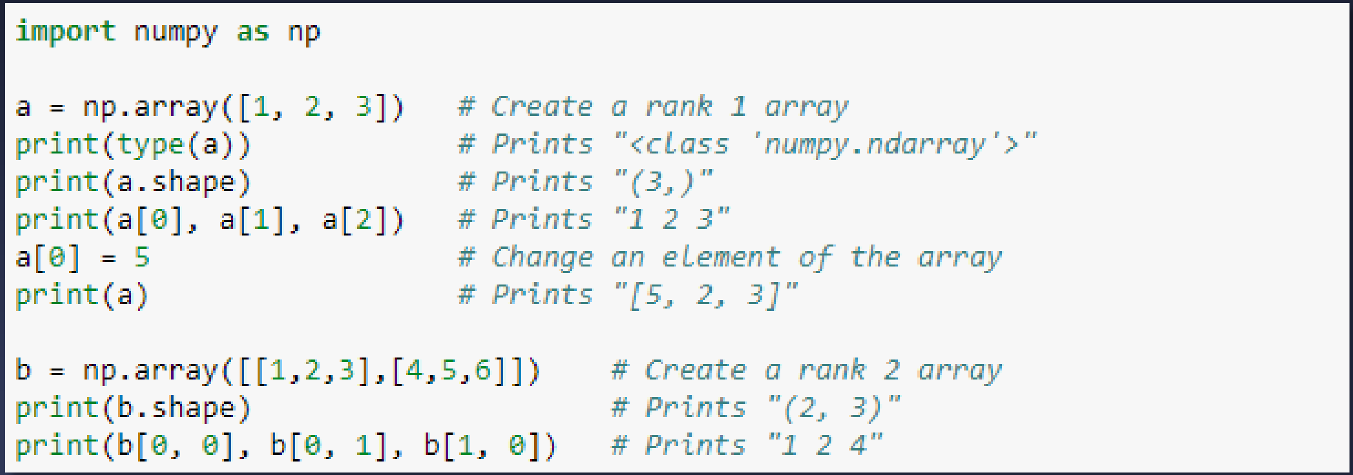
Exercise 2: Dictionary ☺

- Find the meaning of the word from two dictionary
- You should find the intersection between them
- If a word is define in both of them, show the both meaning

CombineDictionary.ipynb

Numpy

- Numpy is the core library for scientific computing in Python.
- It provides a high-performance multidimensional array object, and tools for working with these arrays.

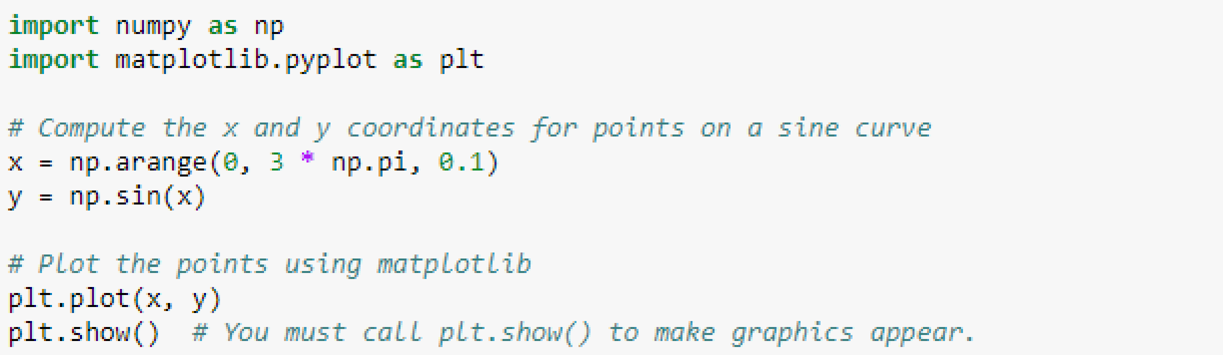


```python
import numpy as np

a = np.array([1, 2, 3])   # Create a rank 1 array
print(type(a))            # Prints "<class 'numpy.ndarray'>"
print(a.shape)            # Prints "(3,)"
print(a[0], a[1], a[2])   # Prints "1 2 3"
a[0] = 5                  # Change an element of the array
print(a)                  # Prints "[5, 2, 3]"

b = np.array([[1,2,3],[4,5,6]])    # Create a rank 2 array
print(b.shape)                     # Prints "(2, 3)"
print(b[0, 0], b[0, 1], b[1, 0])   # Prints "1 2 4"
```

Numpy.ipynb

Plotting

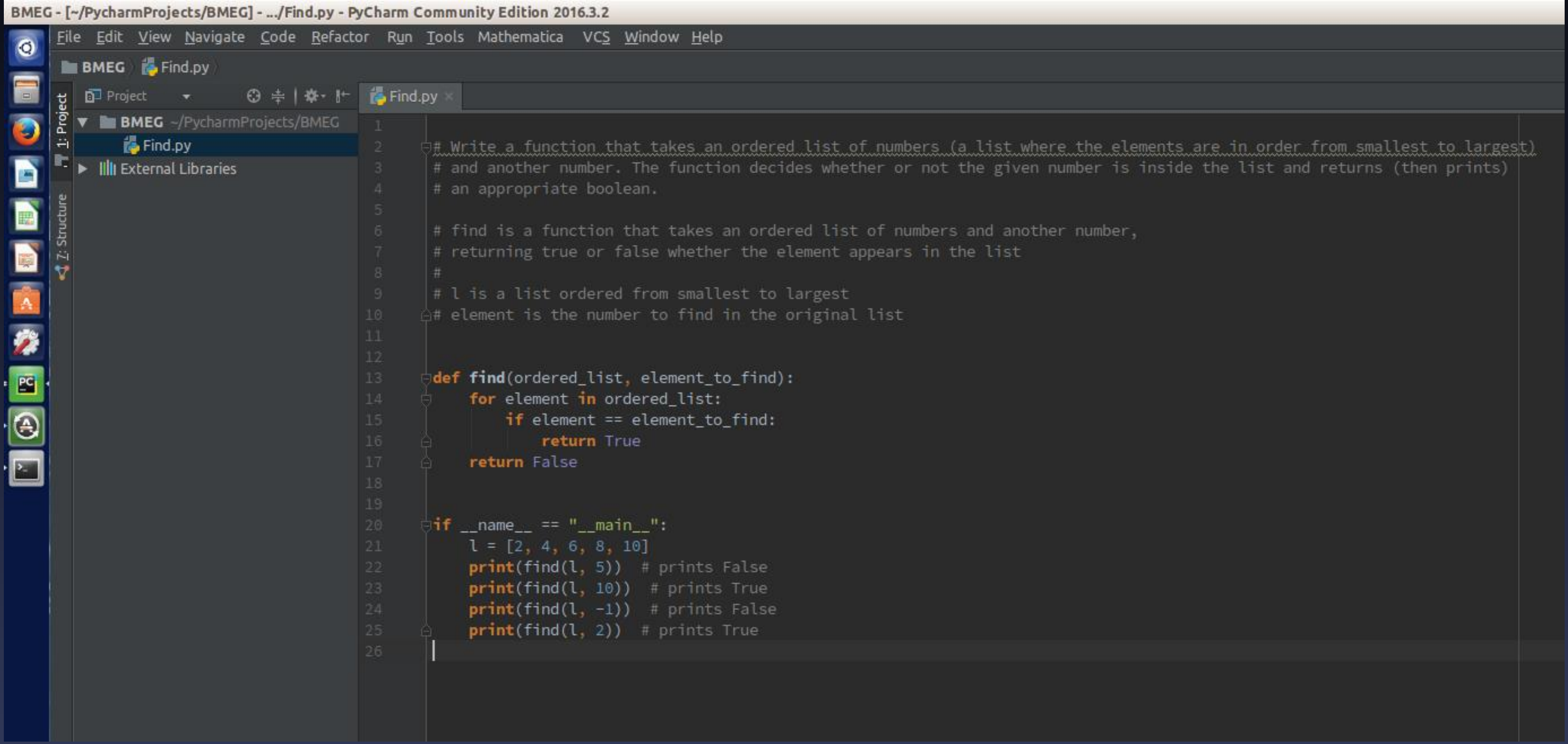
- **Matplotlib** is a plotting library.



```
import numpy as np
import matplotlib.pyplot as plt

# Compute the x and y coordinates for points on a sine curve
x = np.arange(0, 3 * np.pi, 0.1)
y = np.sin(x)

# Plot the points using matplotlib
plt.plot(x, y)
plt.show()  # You must call plt.show() to make graphics appear.
```

Plotting.ipynb

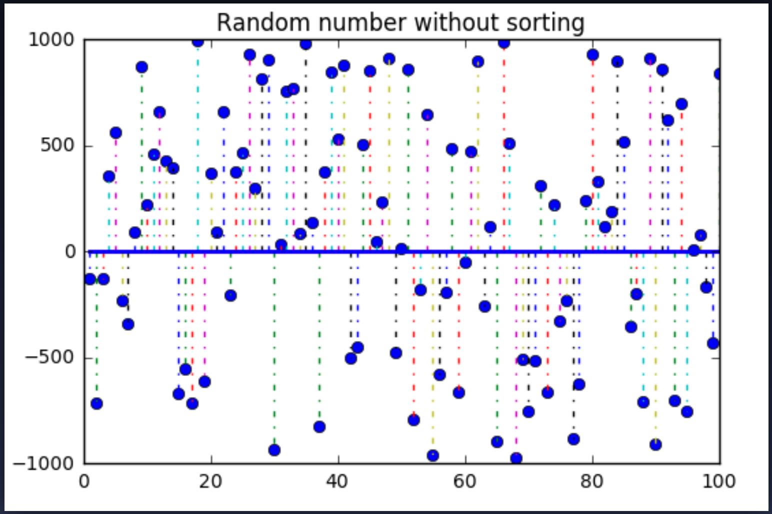
PyCharm

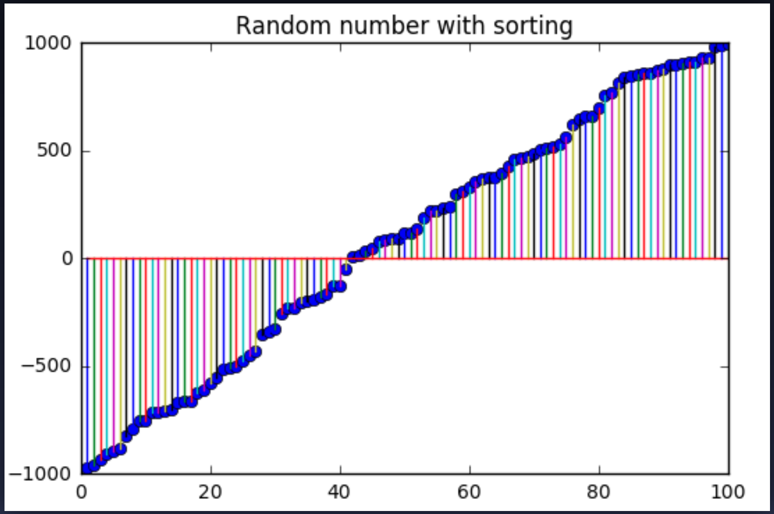

BMEG - [~/PycharmProjects/BMEG] - .../Find.py - PyCharm Community Edition 2016.3.2
File Edit View Navigate Code Refactor Run Tools Mathematica VCS Window Help
BMEG
Find.py
Project
1: Project
Z: Structure
BMEG ~/PycharmProjects/BMEG
Find.py
External Libraries
Find.py

```python

# Write a function that takes an ordered list of numbers (a list where the elements are in order from smallest to largest)
# and another number. The function decides whether or not the given number is inside the list and returns (then prints)
# an appropriate boolean.

# find is a function that takes an ordered list of numbers and another number,
# returning true or false whether the element appears in the list
#
# l is a list ordered from smallest to largest
# element is the number to find in the original list


def find(ordered_list, element_to_find):
    for element in ordered_list:
        if element == element_to_find:
            return True
    return False


if __name__ == "__main__":
    l = [2, 4, 6, 8, 10]
    print(find(l, 5))  # prints False
    print(find(l, 10))  # prints True
    print(find(l, -1))  # prints False
    print(find(l, 2))  # prints True

```

a place of mind

Assignment 😐





Assignment.ipynb

More to expect …

- Object Oriented Programming with Python:
  - By Amir Abdi
- Applied Machine Learning in Python:
  - By Amir Abdi
- Image Analysis and Computer Vision:
  - By Mehran Pesteie

Learn more:

- Python Tutorials and Courses
  - https://hackr.io/tutorials/learn-python
  - https://github.com/wzpan/Learn-Python-The-Hard-Way/tree/master/Python2
- Python 3 Programming Introduction Tutorial
  - https://pythonprogramming.net/introduction-to-python-programming/
- Python tutorials for beginners
  - http://thepythonguru.com/