

CIS 2300

PROGRAMMING & COMPUTATIONAL THINKING



w Anna O'Connell

REFERENCES.

Safari is a collection of over **40,000 ebooks in IT.**

It includes all titles published by **O'Reilly Media**, **Packt** and others.

How to access [Safari for Baruch Students](#).

Safari link: '[Learning IPython for Interactive Computing and Data Visualization - Second Edition](#)' by Cyrille Rossant.
Published by Packt Publishing, 2015

[Cyrille Rossant](#) > [Github](#) > [Chapter 1.3 on Github](#)

[Informal Introduction to Python from Python.org](#)

LECTURE 1 AGENDA

1. Anaconda.
2. Python.
3. IPython.
4. Jupyter Notebook.



DOWNLOAD AND INSTALL PYTHON ANACONDA DISTRIBUTION



<https://www.anaconda.com/distribution/>

ANACONDA DISTRIBUTION IS...

A way to perform Python/R **data science** and **machine learning**:

- Download **1,500+ packages**.
- Manage libraries, dependencies, and environments with **Conda**.
- Develop and train machine learning and deep learning models with **scikit-learn**, **TensorFlow**, and **Theano**.

Packages included in Anaconda 4+, or get with "conda install PACKAGENAME"

1. NumPy | numpy.org

N-dimensional array for numerical computation

2. SciPy | scipy.org

Collection of numerical algorithms and toolboxes, including signal processing and optimization

3. Matplotlib | matplotlib.org

Plotting library for Python

4. Pandas | pandas.pydata.org

Powerful Python data analysis toolkit

5. Seaborn | stanford.edu/~mwaskom/software/seaborn/

Statistical data visualization

6. Bokeh | bokeh.pydata.org

Interactive web visualization library

7. SciKit-Learn | scikit-learn.org/stable

Python modules for machine learning and data mining

8. NLTK | nltk.org

Natural language toolkit

9. Notebook | jupyter.org

Web-based interactive computational environment combines code execution, rich text, mathematics, plots and rich media

10. R essentials | conda.pydata.org/docs/r-with-conda.html

R with 80+ of the most used R packages for data science
"conda install -c r r-essentials"



[HTTPS://CONDA.IO/EN/LATEST/](https://conda.io/en/latest/)

- Package Management System:

- Search for packages
- Install new packages
- Update packages to a particular version
- Remove packages

- Environment Management System:

- Create new environments
- Activate and deactivate environments
- Get a list of all your environments
- List all installed packages in a specific environment

- Home
- Environments
- Learning
- Community


- Documentation
- Developer Blog
- Feedback



Applications on

Channels

Refresh



jupyter
notebook
4.2.3

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.


[Launch](#)



IP[y]
qtconsole
4.2.1

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.


[Launch](#)



spyder
3.0.0

Scientific PYTHON Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

[Launch](#)



glueviz
0.8.2

Multidimensional data visualization across files. Explore relationships within and among related datasets.

[Launch](#)

GUIDO VAN ROSSUM
IS
PYTHON BDFL

BENEVOLENT

BDFL = DICTATOR

FOR

LIFE

KEYNOTE: GUIDO VAN ROSSUM

Guido's Personal Home Page

Python BDFL, Dutch, 63 y.o.

Works for DropBox.

Lives in Belmont, California.

Created Python in December **1989**:

“I was looking for a ‘hobby’ programming project that would keep me occupied during the week around Christmas.”





A LINK TO MONTY PYTHON'S SILLY OLYMPICS

OPENING A TERMINAL



on WINDOWS, type `cmd` in the Run box.

For more recent OS, press the Windows + R keys, type `powershell` in the Run box, and press *Enter*.



on OS X, you can open the Terminal application by pressing Cmd + Space, typing `terminal`, and pressing *Enter*.



on LINUX, you can open the Terminal from your application manager.

TO INVOKE THE PYTHON INTERPRETER

From command-line, type:

python

To exit, type:

exit()

Also **quit()**, **Control-D** on Unix, **Control-Z** on Windows

COMMENTS IN PYTHON START WITH THE HASH CHARACTER

A comment may not appear within a string literal.

```
# this is the first comment  
spam = 1 # and this is the second comment  
        # ... and now a third!  
text = "# This is not a comment because it's inside quotes."
```

USING PYTHON AS A CALCULATOR

The interpreter acts as a simple calculator:

```
>>> 2 + 2
4
>>> 50 - 5*6
20
>>> (50 - 5*6) / 4
5.0
>>> 8 / 5 # division always returns a floating point number
1.6
```


FLOOR DIVISION '`//`' ROUNDS DOWN TO NEAREST INTEGER.

To calculate the remainder you can use `%`

```
>>> 17 / 3 # classic division returns a float
5.666666666666667
>>>
>>> 17 // 3 # floor division discards the fractional part
5
>>> 17 % 3 # the % operator returns the remainder of the division
2
>>> 5 * 3 + 2 # result * divisor + remainder
17
```

CALCULATING POWERS

It is possible to use the `**` operator to calculate powers

```
>>> 5 ** 2 # 5 squared
```

```
25
```

```
>>> 2 ** 7 # 2 to the power of 7
```

```
128
```

```
>>>
```

STRINGS

Strings can be enclosed in single or double quotes.

\ can be used to escape quotes.

```
>>> 'spam eggs' # single quotes
'spam eggs'
>>> 'doesn\'t' # use \' to escape the single quote...
"doesn't"
>>> "doesn't" # ...or use double quotes instead
"doesn't"
>>> '"Yes," they said.'
'"Yes," they said.'
>>> "\"Yes,\" they said."
'"Yes," they said.'
>>> 'Isn\'t," they said.'
'Isn\'t," they said.'
```

>>>

DISPLAY OUTPUT WITH THE *PRINT()* FUNCTION.

Print function displays string literals on the screen.

```
# single quotes
print ('From Python philosophy...')

# double quotes
print ("Readability counts.")

# triple quotes
print ("""Simple is better than complex.""")
```

STRING LITERALS CAN SPAN MULTIPLE LINES

Use triple quotes `"""..."""` or `'...''`

```
print("""\
Usage: thingy [OPTIONS]
    -h                Display this usage message
    -H hostname       Hostname to connect to
""")
```

THE EQUAL SIGN IS USED TO ASSIGN A VALUE TO A VARIABLE.

```
>>> width = 20
>>> height = 5 * 9
>>> width * height
900
```

```
>>>
```

A VARIABLE MUST BE DEFINED...

If a variable is not “defined” (assigned a value), trying to use it will give you an error:

```
>>> n # try to access an undefined variable
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'n' is not defined
```

```
>>>
```

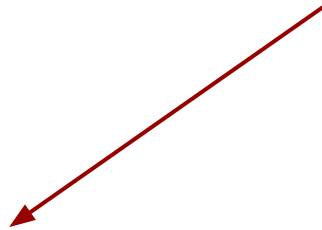
THE UNDERScore _ VARIABLE

The last printed expression is assigned to the variable `_` in interactive mode.

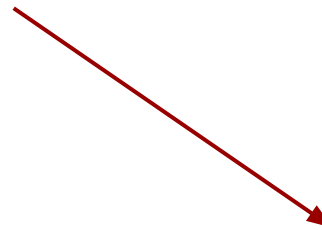
```
>>> tax = 12.5 / 100
>>> price = 100.50
>>> price * tax
12.5625
>>> price + _
113.0625
>>> round(_, 2)
113.06
```

>>>

IP[y]: IPython
Interactive Computing



Command Shell
for Interactive Computing



IPython Kernel
provides computations for
the notebook front-end.

McKinney Chapter 2.

IPYTHON.ORG

[Documentation](#)

FERNANDO PÉREZ FPEREZ.ORG IS THE CREATOR OF IPYTHON



- ★ Born in Medellin, Colombia
- ★ Now in San Fran Bay Area
- ★ Works at UC Berkeley on research for neuroimaging
- ★ PhD in Particle Physics
- ★ [Twitter](#)
- ★ [Interview](#)

TO LAUNCH IPYTHON SHELL

From command-line, type:

ipython

exit()

To run a version of IPython that runs in a PyQt Console:

ipython qtconsole

MORE SHORTCUTS

Keystroke	Action
Ctrl - l	Clear terminal screen
Ctrl - c	Interrupt current Python command
Ctrl - d	Exit iPython session



<http://jupyter.org>

REFERENCES

Main documentation on Jupyter:

<http://jupyter.readthedocs.io/en/latest/>

Jupyter Notebook interface explained:

<http://jupyter-notebook.readthedocs.io/en/latest/notebook.html>

[Online tutorial](#)

HOW TO LAUNCH JUPYTER NOTEBOOK

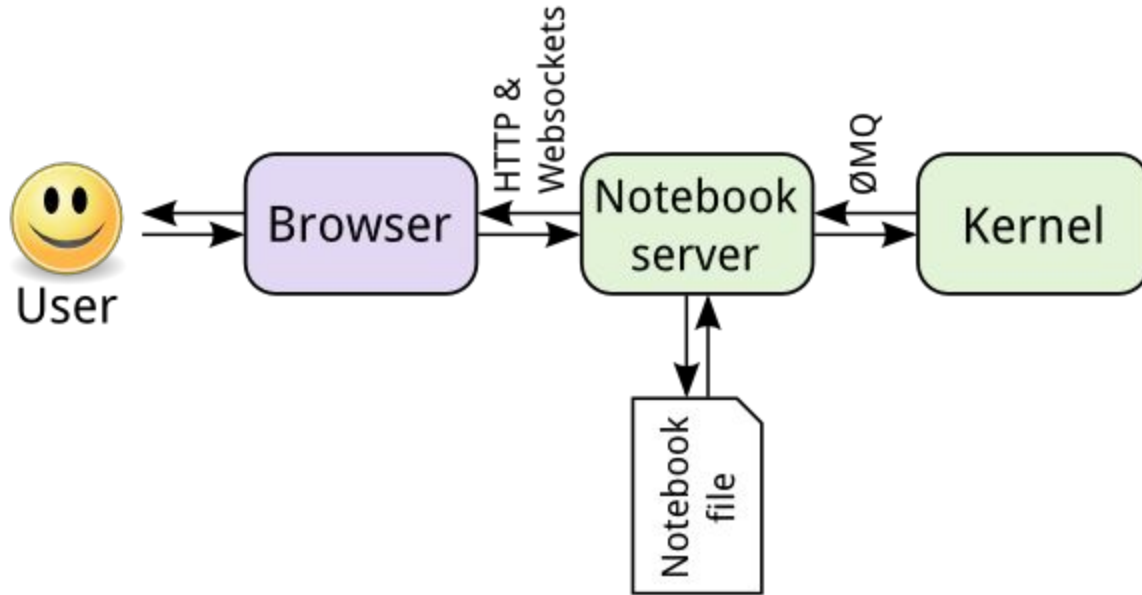
From command-line type:

jupyter notebook

To close the Notebook server from the terminal:

Ctrl + C

IPYTHON AND JUPYTER NOTEBOOK



KERNEL IS A PROCESS RUNNING AN
INTERACTIVE SESSION.

WHEN USING IPYTHON, THIS KERNEL IS A
PYTHON PROCESS.

THERE ARE KERNELS IN MANY OTHER
LANGUAGES.

JUPITER NOTEBOOK

Is an interactive **document** containing code, text, images and / or many other elements.

A notebook is saved in a file with the **.ipynb** extension.

This file is a plain text file storing a **JSON** data structure.

JSON FILE FORMAT

cell 1

```
{  
  "cells": [  
    {  
      "cell_type": "code",  
      "execution_count": 1,  
      "metadata": {  
        "collapsed": false  
      },  
      "outputs": [  
        {  
          "name": "stdout",  
          "output_type": "stream",  
          "text": [  
            "Hello world\n"  
          ]  
        }  
      ],  
      "source": [  
        "print(\"Hello world\")"  
      ]  
    }  
  ],  
}
```

cell 2

```
{  
  "cell_type": "code",  
  "execution_count": 2,  
  "metadata": {  
    "collapsed": false  
  },  
  "outputs": [  
    {  
      "data": {  
        "text/plain": [  
          "4"  
        ]  
      },  
      "execution_count": 2,  
      "metadata": {},  
      "output_type": "execute_result"  
    }  
  ],  
  "source": [  
    "2+2"  
  ]  
},
```

MODAL INTERFACE

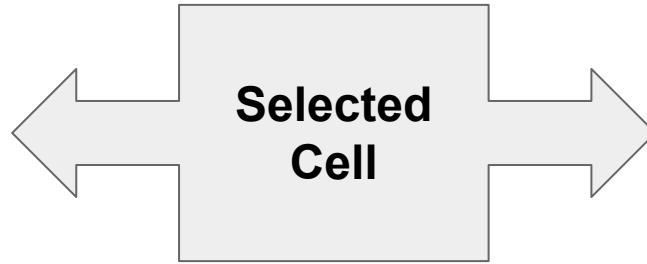
Edit Mode to Write Code

Green border around selected cell

A pen icon at the top right of the interface

Click inside cell to enable the edit mode.

Double-click with Markdown cells.



Command Mode to Operate on Cells

Grey border around selected cell

No pen icon at the top right of the interface

Click outside the cell to enable the command mode. Or press `esc` key.

PRESS 'h' (HELP) TO DISPLAY A LIST OF AVAILABLE SHORTCUTS

In command mode of a markdown cell

KEYBOARD SHORTCUTS

- *Ctrl + Enter*: run the cell
- *Shift + Enter*: run the cell and select the cell below
- *Alt + Enter*: run the cell and insert a new cell below
- *Ctrl + S*: save the notebook

New paragraph

This is *rich* **text** with [\[links\]\(http://ipython.org\)](http://ipython.org), equations:

$$\hat{f}(\xi) = \int_{-\infty}^{+\infty} f(x) \, \mathrm{e}^{-i \xi x} \, dx$$

code with syntax highlighting:

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

and images:

![This is an image](http://jupyter.org/images/jupyter-sq-text.svg)

New paragraph

This is *rich* **text** with [links](#), equations:

$$\hat{f}(\xi) = \int_{-\infty}^{+\infty} f(x) \, \mathrm{e}^{-i \xi x} \, dx$$

code with syntax highlighting:

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

and images:

IP[y]: IPython
Interactive Computing

FROM ROSSANT, CHAPTER 1