



香港浸會大學
HONG KONG BAPTIST UNIVERSITY

Tools Installation

JOUR7280

Big Data Analytics for Media and Communication

Instructor: Dr. Xiaoyi Fu

Getting started: Tools

- Talking to your computer: Command line interface (CLI)
- Text Editor: vscode, sublime, notepad++, or others
- Platform for publishing and socializing: Git and GitHub, and Markdown language
- The tool: Python 3.x (Anaconda 3) and Jupyter Notebook

Command Line Interface (CLI) basics

- Open CLI: spotlight search – “terminal” / start menu search - ”cmd”
- An interesting tutorial ([macOS](#), [Windows](#))
- Basic commands of CLI:
 - – pwd
 - – date
 - – mkdir
 - – echo
 - – ls
 - – cd
 - – touch
 - – cp
 - – rm
 - – mv

Git

- Git is a “version control system”: records changes to a file or set of files over time and users can recall specific versions later.
 - Useful for collaboration project
- Download and install Git ([mac](#), [windows](#))
- Setting up your Git account
 - `git config --global user.email "your@email.com"`
 - `git config --global user.name "your name"`

GitHub

- GitHub is a web-based hosting service for software development projects that use Git version control system.” ([What is GitHub](#))
- Push and pull
- Public/Private repositories
- The social aspects of GitHub (share, fork, star)
 - A profile that shows your portfolio

GitHub

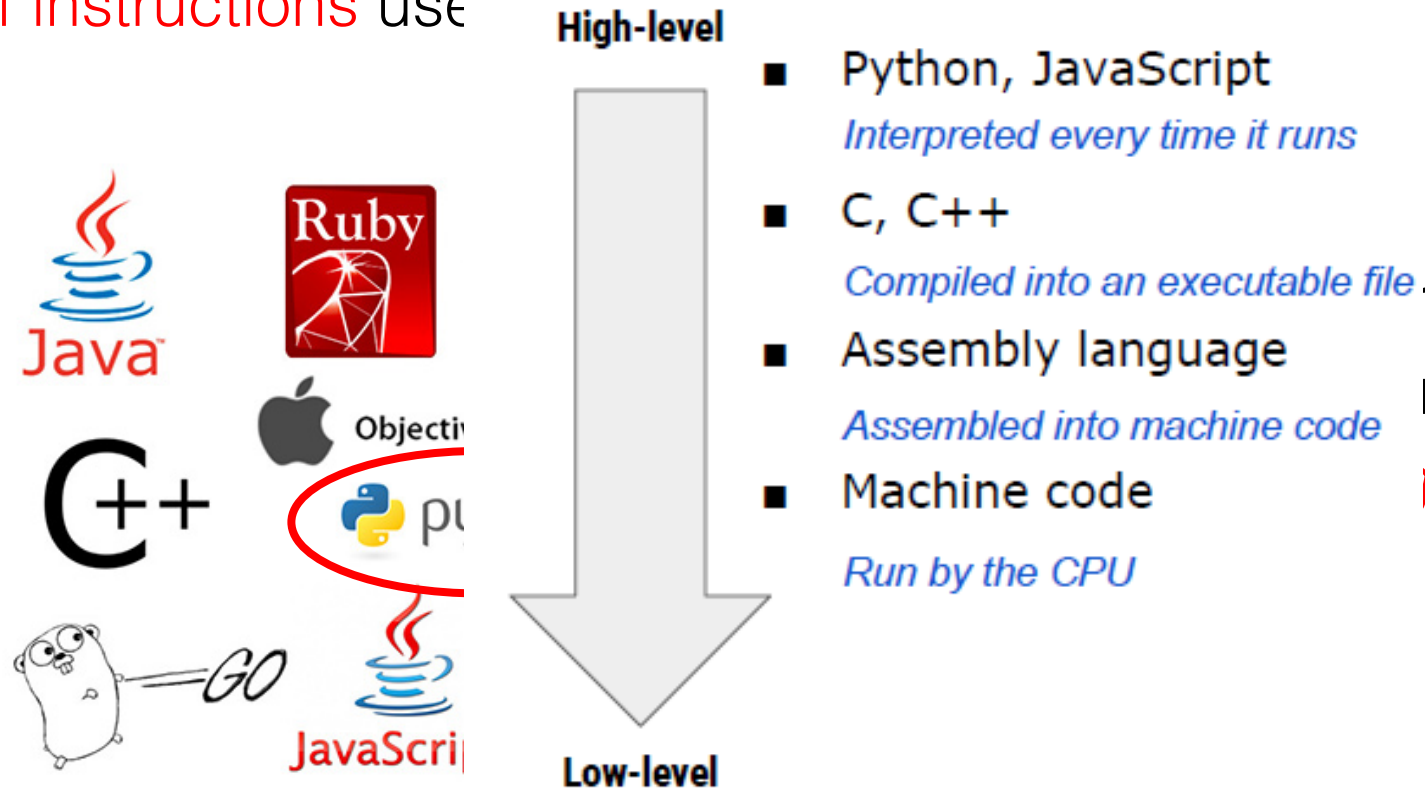
- Creating your GitHub repository (“repo”)
 - Creating your own repo (with a Readme file)
 - “Fork” another user’s repository
- Creating a local copy
- Clone the Repo
- Pull request and collaborative projects

Markdown language

- Markdown is a lightweight markup language with plain text formatting syntax
- readme.md
- Markdown language quick guide [[Link](#)]
- Work with markdown language
 - vscode Preview
 - GIT MD Syntax [[Link](#)]
 - GIT Deeper MD Syntax [[Link](#)]
 - GIT MD Emojis! [[Link](#)]

Programming Language

- A programming language is a formal language, which comprises a set of instructions used to put.



n programming language refers to **high-level** **ies**, such as C, C++, Java, Matlab...

Introduction to Python

- **Python** is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation.

Increasingly popular!

| Aug 2018 | Aug 2017 | Change | Programming Language | Ratings | Change |
|----------|----------|--------|----------------------|---------|--------|
| 1 | 1 | | Java | 16.881% | +3.92% |
| 2 | 2 | | C | 14.966% | +8.49% |
| 3 | 3 | | C++ | 7.471% | +1.92% |
| 4 | 5 | ⬆ | Python | 6.992% | +3.30% |
| 5 | 6 | ⬆ | Visual Basic .NET | 4.762% | +2.19% |
| 6 | 4 | ⬇ | C# | 3.541% | -0.65% |
| 7 | 7 | | PHP | 2.925% | +0.63% |
| 8 | 8 | | JavaScript | 2.411% | +0.31% |
| 9 | - | ⬆ | SQL | 2.316% | +2.32% |
| 10 | 14 | ⬆ | Assembly language | 1.409% | -0.40% |

| Jan 2020 | Jan 2019 | Change | Programming Language | Ratings | Change |
|----------|----------|--------|----------------------|---------|--------|
| 1 | 1 | | Java | 16.896% | -0.01% |
| 2 | 2 | | C | 15.773% | +2.44% |
| 3 | 3 | | Python | 9.704% | +1.41% |
| 4 | 4 | | C++ | 5.574% | -2.58% |
| 5 | 7 | ⬆ | C# | 5.349% | +2.07% |
| 6 | 5 | ⬇ | Visual Basic .NET | 5.287% | -1.17% |
| 7 | 6 | ⬇ | JavaScript | 2.451% | -0.85% |
| 8 | 8 | | PHP | 2.405% | -0.28% |
| 9 | 15 | ⬆ | Swift | 1.795% | +0.61% |
| 10 | 9 | ⬇ | SQL | 1.504% | -0.77% |

TIOBE Rankings

Why popular & Features

- Code readability, shorter codes, ease of writing
 - fewer lines of code in comparison to languages such as C++ or Java.
- “Simplicity is the best”
 - Closer to English language; Easy to learn
 - More emphasis on the solution to the problem rather than the syntax
- Interpreted language
 - Directly run the program from the source code.
 - No separate compilation and execution steps like C and C++.
- Rich Library Support
 - The Python Standard Library is very vast.

```
public class HelloWorld
{
    public static void main (String[] args)
    {
        System.out.println("Hello, world!");
    }
}
```

Java Code

```
print("Hello, world!") # Python version 3
```

Python Code

What Python can do



SciPy.org

Scientific Computing Tools for Python

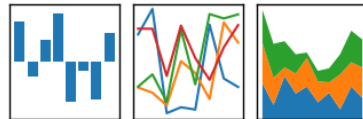
SciPy refers to several related but distinct entities:

- The *SciPy ecosystem*, a collection of open source software for scientific computing in Python.
- The *community* of people who use and develop this stack.
- Several *conferences* dedicated to scientific computing in Python - SciPy, EuroSciPy and SciPy.in.
- The *SciPy library*, one component of the SciPy stack, providing many numerical routines.

Scientific Computing

pandas

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



[home](#) // [about](#) // [get pandas](#) // [documentation](#) // [community](#) //

Python Data Analysis Library

pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the [Python](#) programming language.

pandas is a [NumFOCUS](#) sponsored project. This will help ensure the success of development of *pandas* as a world-class open-source project, and makes it possible to [donate](#) to the project.

Data analysis



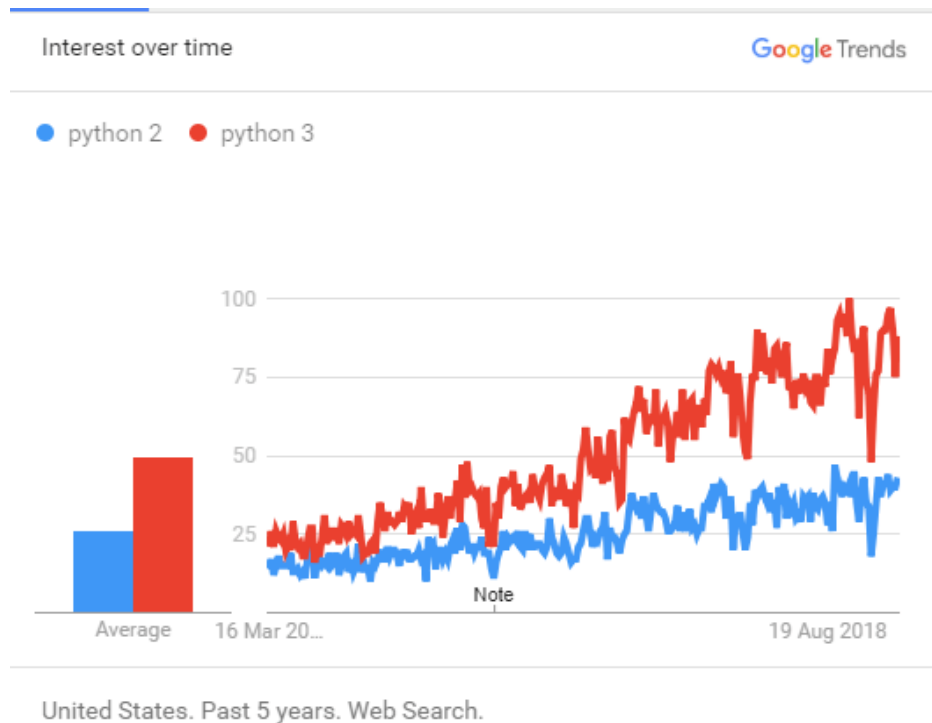
Machine Learning

matplotlib

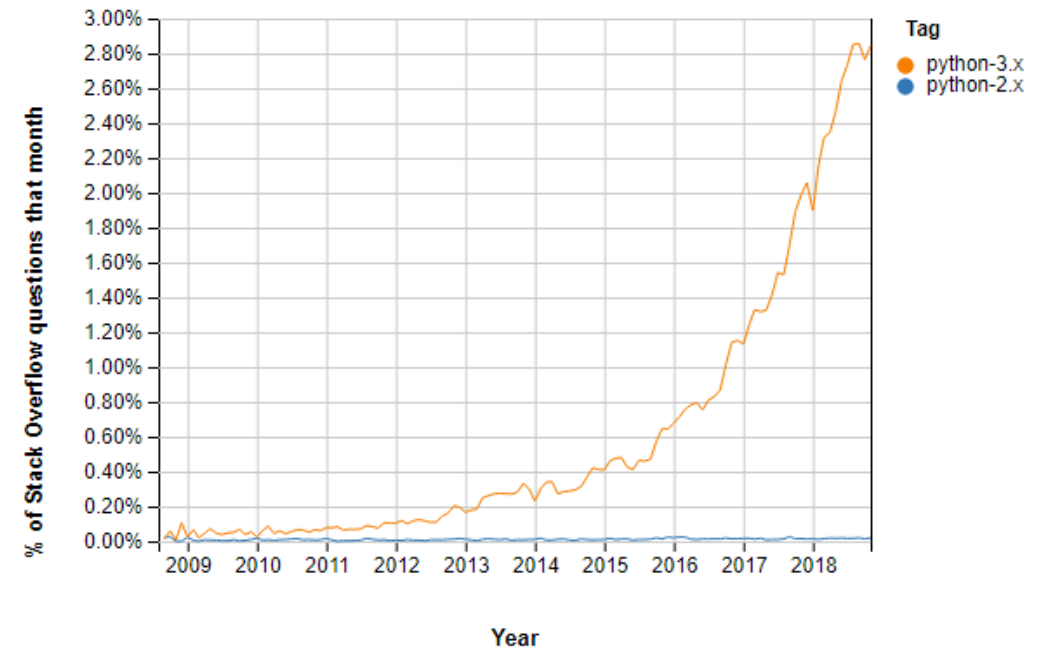
Version 3.0.2

Data Visualization

Python 2 vs 3

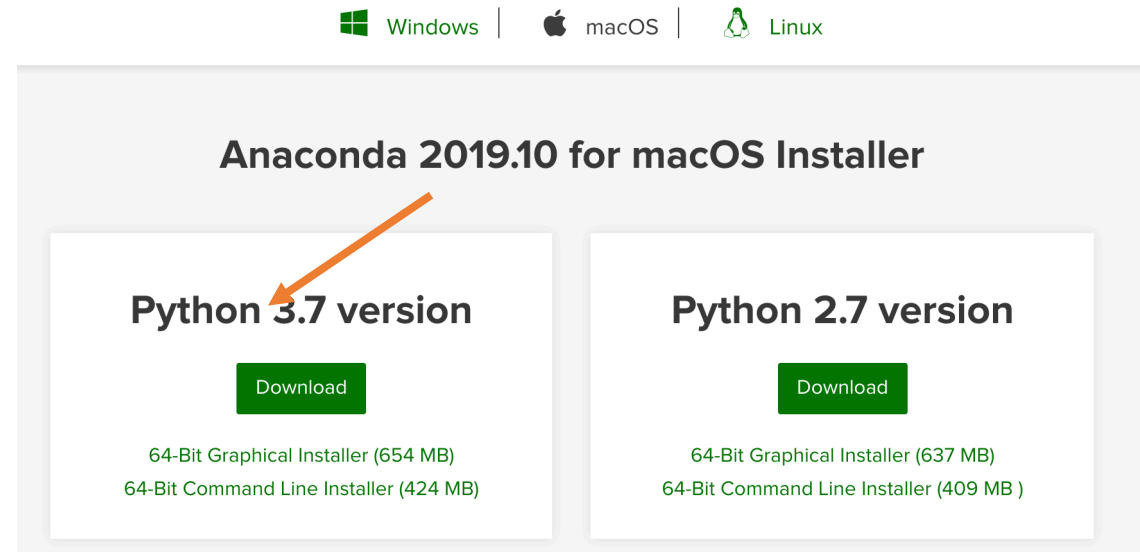


Google Trends Python 2 vs. Python 3



Stack Overflow Questions Python 2 vs. Python 3

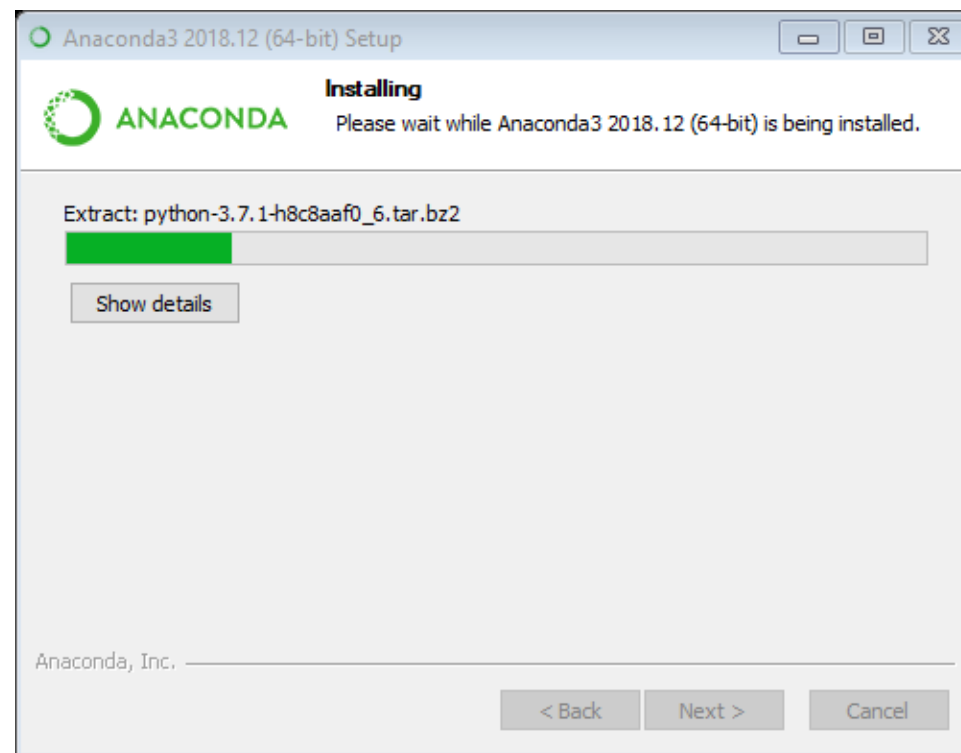
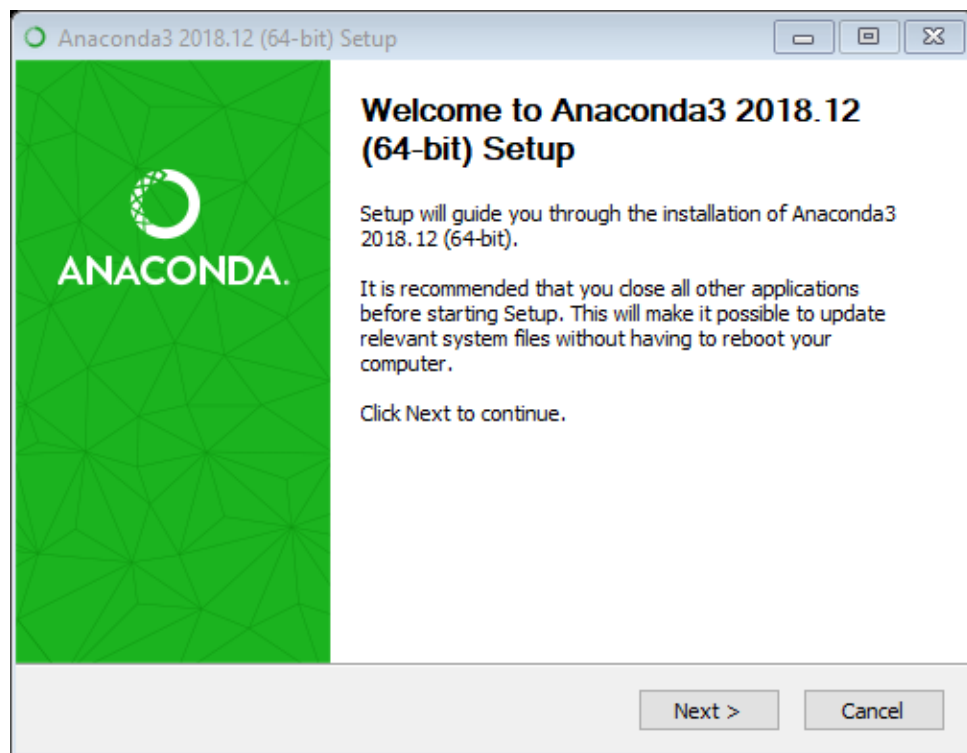
Installing Python and Jupyter Notebook



- Python 3.7
 - Anaconda [[Link](#)]
 - A free and open-source distribution of the Python and R programming languages
- Jupyter Notebook
 - Included in the Anaconda installation

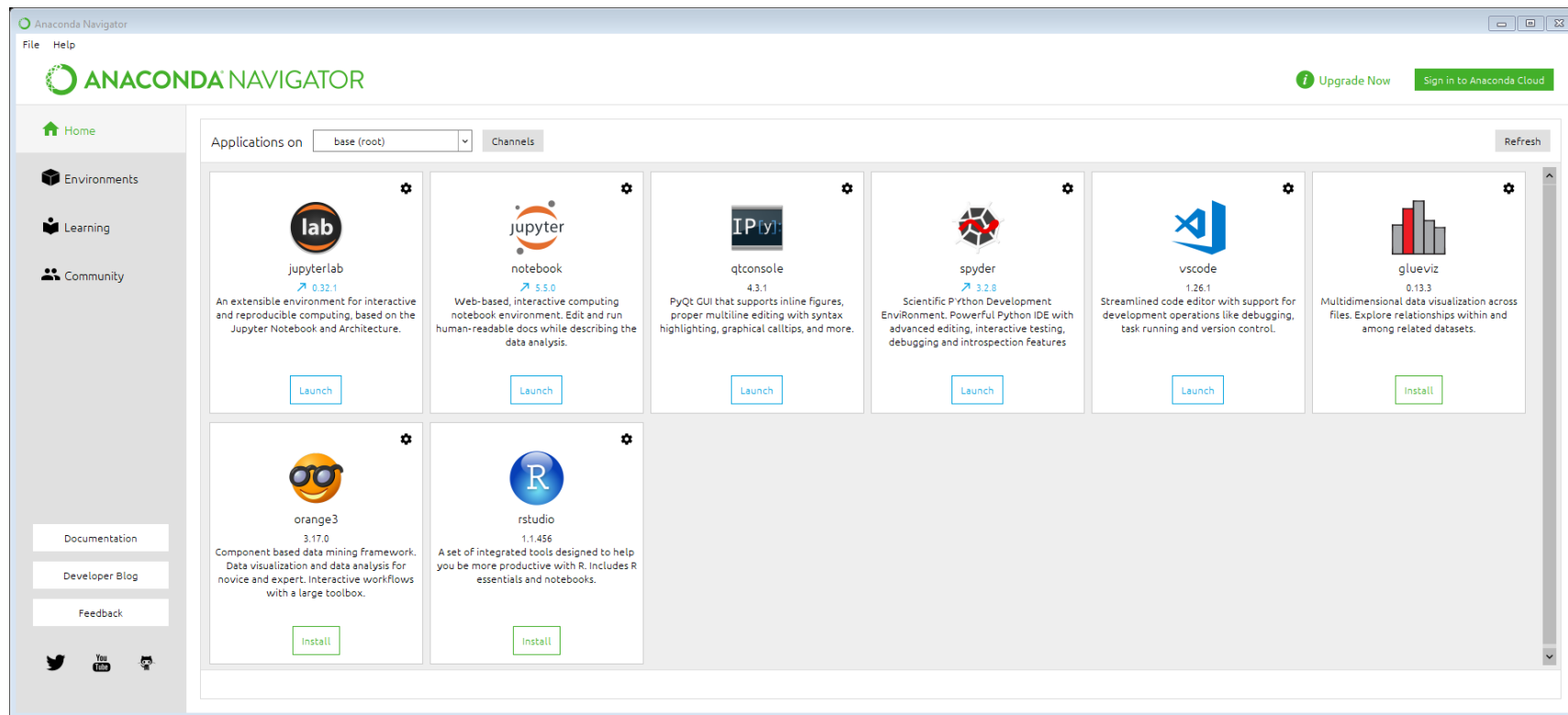
Install Anaconda

- Run the Anaconda installer.

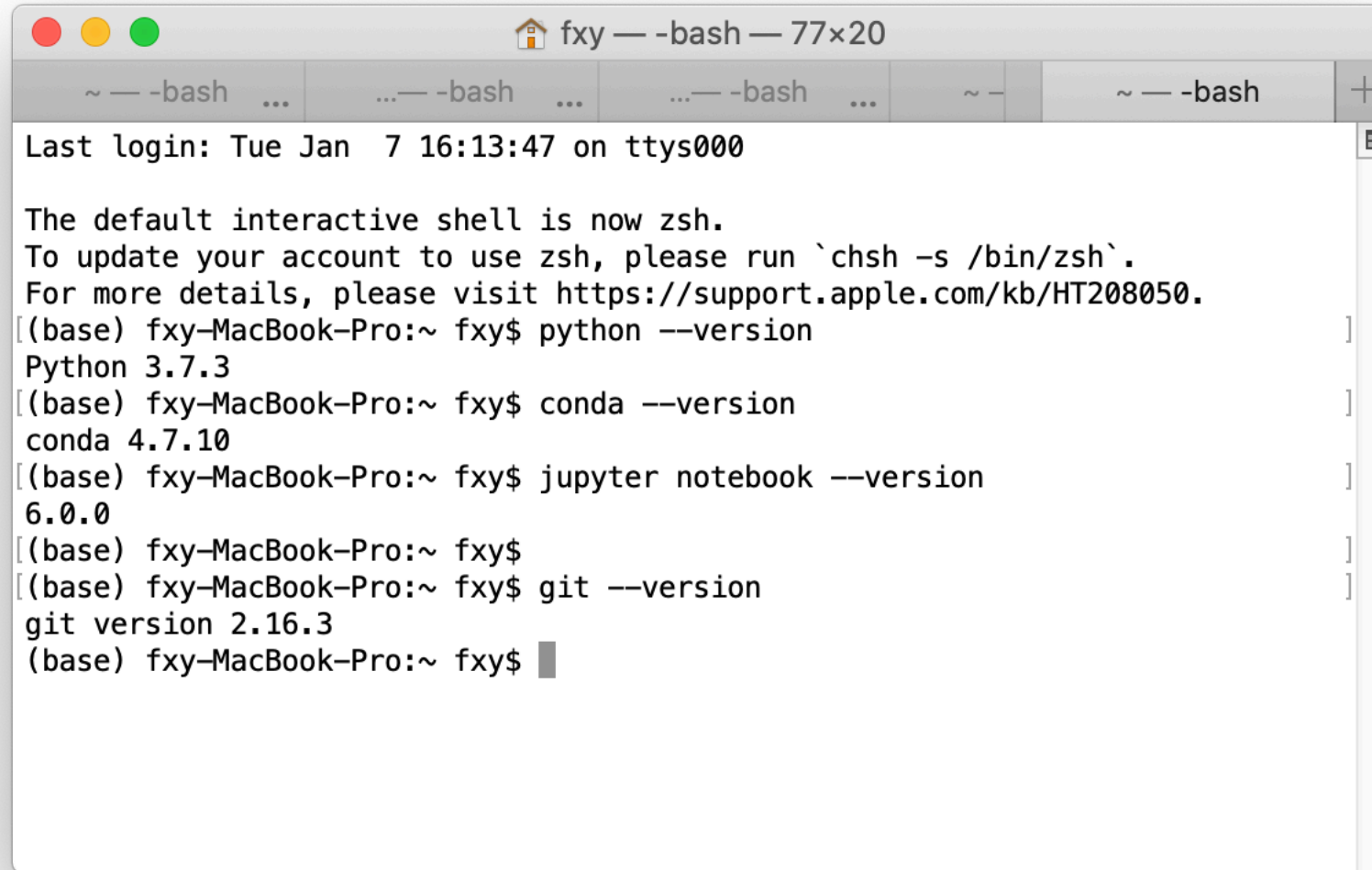


Anaconda

- Open the *Anaconda Navigator* after installation. You can install and launch different environment for later development.



Check your existing versions

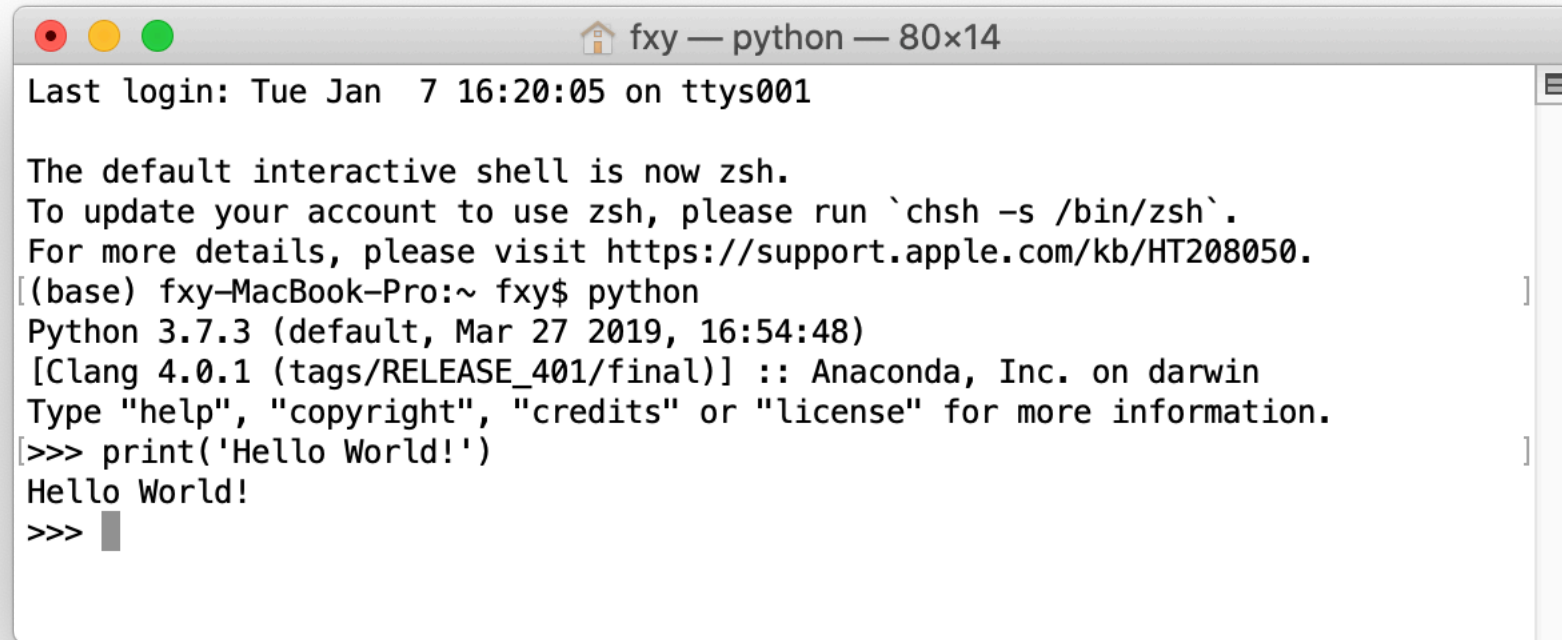


```
fxy — -bash — 77x20
~ — -bash ... .. — -bash ... .. — -bash ... ~ — ~ — -bash +
Last login: Tue Jan  7 16:13:47 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) fxy-MacBook-Pro:~ fxy$ python --version
Python 3.7.3
(base) fxy-MacBook-Pro:~ fxy$ conda --version
conda 4.7.10
(base) fxy-MacBook-Pro:~ fxy$ jupyter notebook --version
6.0.0
(base) fxy-MacBook-Pro:~ fxy$
(base) fxy-MacBook-Pro:~ fxy$ git --version
git version 2.16.3
(base) fxy-MacBook-Pro:~ fxy$
```


Your first Python program

- Two ways of printing “Hello World!”
 - via CLI

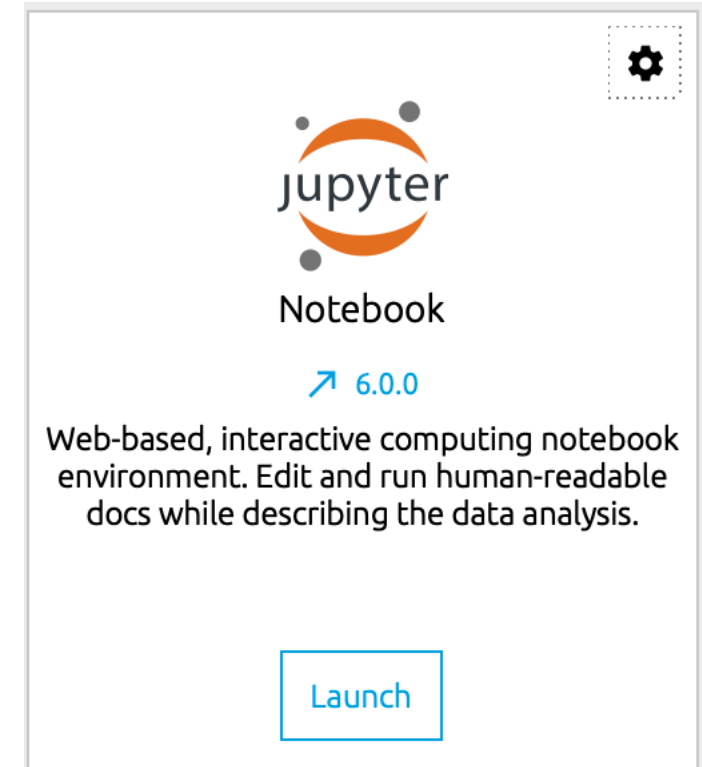
A screenshot of a macOS terminal window titled 'fxy — python — 80x14'. The window shows the output of a 'python' command. It starts with a login message, then informs the user that the default shell is zsh. The user enters 'python', which starts the Python 3.7.3 interpreter. The interpreter displays its version and the Anaconda environment. The user enters '>>> print('Hello World!')', and the interpreter outputs 'Hello World!' followed by a new prompt '>>>'.

```
fxy — python — 80x14
Last login: Tue Jan  7 16:20:05 on ttys001

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) fxy-MacBook-Pro:~ fxy$ python
Python 3.7.3 (default, Mar 27 2019, 16:54:48)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> print('Hello World!')]
Hello World!
>>>
```

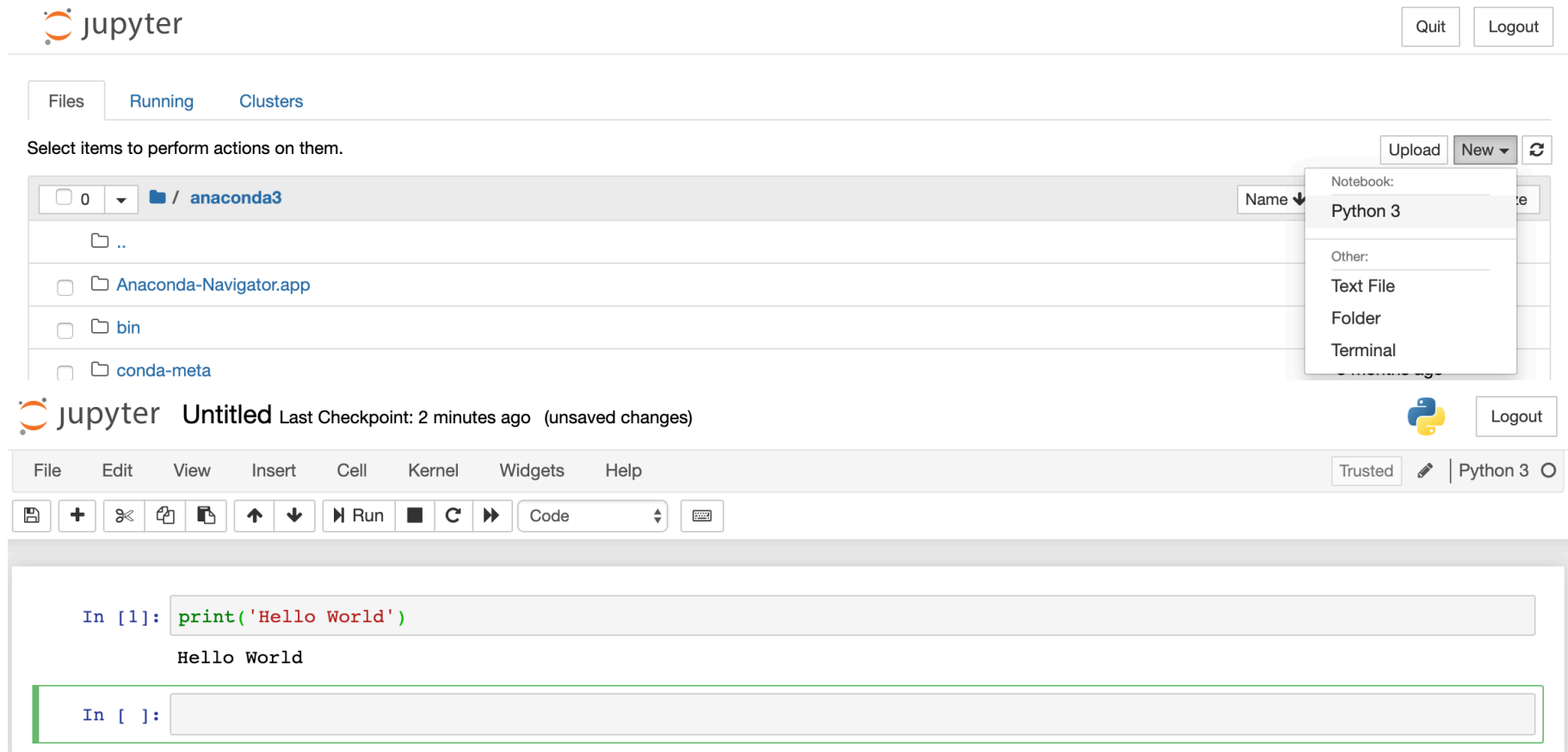
Your first Python program

- Via Jupyter Notebook
 - Contain both code and rich text elements, such as figures, links, equations, ...
 - The ideal place to bring together an analysis description, and its results
 - We will use Jupyter Notebook in this course



Your first Python program

- Two ways of printing “Hello World!”
 - via Jupyter Notebook



The screenshot displays the Jupyter Notebook web interface. At the top, the Jupyter logo and 'jupyter' text are on the left, and 'Quit' and 'Logout' buttons are on the right. Below this is a navigation bar with 'Files', 'Running', and 'Clusters' tabs. A message 'Select items to perform actions on them.' is followed by an 'Upload' button, a 'New' dropdown menu, and a refresh icon. The 'New' menu is open, showing options: 'Notebook: Python 3' and 'Other: Text File, Folder, Terminal'. The file browser shows a directory structure under '/ anaconda3' with files like '..', 'Anaconda-Navigator.app', 'bin', and 'conda-meta'. Below the file browser, the Jupyter logo and 'Untitled' title are shown, along with 'Last Checkpoint: 2 minutes ago (unsaved changes)' and a 'Logout' button. A menu bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. A toolbar contains icons for saving, adding, deleting, and running cells, along with a 'Code' dropdown. The main area shows a code cell with the input 'In [1]: print('Hello World')' and the output 'Hello World'. Below it is an empty code cell with 'In []:'.

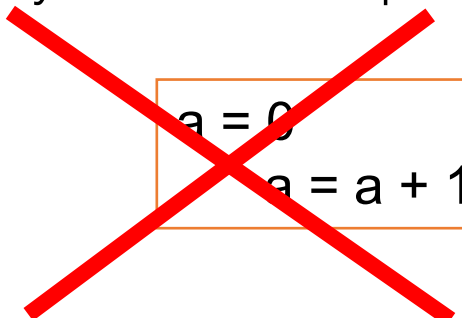
Before we start

- The most controversial feature of Python's syntax: Whitespace is meaningful!

```
a = 0  
a = a + 1
```

is different from

```
a = 0  
a = a + 1
```

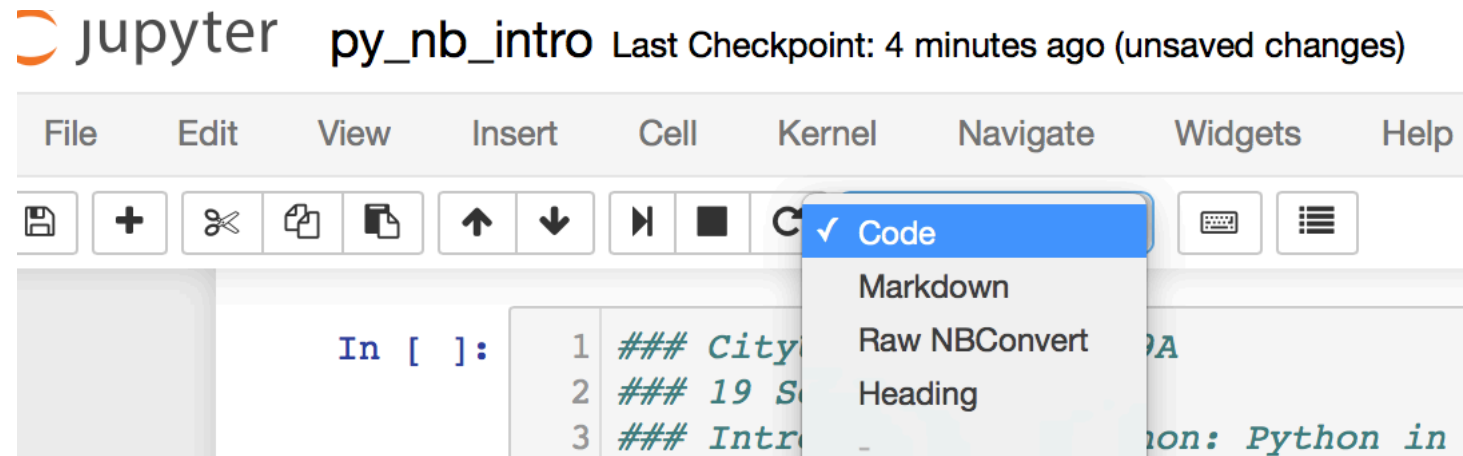


- **Indentation:** an empty space at the beginning of a line
 - It is used to identify blocks of code
 - If a block of code is “inside” the other, it means it should be executed separately from the previous (or when something happens)
- It is very important when
 - Using conditional statements (we’ll see later)
 - Using iterative statements
 - Defining functions

THIS IS A SYNTAX ERROR

Jupyter Notebook

- Jupyter Notebook can contain two main "cell types"
 - Markdown
 - Code
- To change cell type:



Jupyter Notebook

- Some short-cut keys
 - “alt” + “enter”: adding one more line
- When selecting a cell (NOT inside a cell)
 - “a” (adding one more line above the current line)
 - “b” (adding one more line below the current line)
 - “dd” (removing the current line - seems to be undoable...so please be careful here)
 - “m” turns the current line into markdown

Code cells

- This cell is ready to be executed

```
In [ ]: # print - comments are written in this way  
print( "Hello World!" )
```

- No number inside brackets ([])
- No output below
- This cell has been already executed

```
In [1]: # print - comments are written in this way  
print( "Hello World!" )
```

Hello World!

- Number in brackets ([1]) represents the order of execution
- Output below the cell

Kernel

- The kernel's state persists over time and between cells
- It pertains to the document as a whole and not individual cells.

Exercise

- Install Anaconda & Jupyter notebook
- Try “Hello World!” program
- Optional
 - Register a GitHub account
 - Create a new repository
 - Write a readme.md

References

- Anaconda installation FAQ [[Link](#)]
- Add Anaconda3 to path (mac) [[link](#)]
- Git tutorial [[link](#)]
- Change Jupyter notebook working directory [[link](#)]
- Learning website
 - <https://www.learnpython.org/>
 - <https://stackoverflow.com/>

Thank You



References

Books

- Magnus Lie Hetland. (2005). *Beginning Python: From Novice to Professional*. Apress. Retrieved from <https://www.apress.com/gp/book/9781590599822>
- Eric Matthes. (2015). *Python Crash Course. A Hands-On, Project-Based Introduction to Programming*. No Starch Press. Retrieved from <https://nostarch.com/pythoncrashcourse>
- Peter Harrington. (2012). *Machine Learning in Action*. Manning Publications Co., Greenwich, CT, USA. Retrieved from <https://www.manning.com/books/machine-learning-in-action>

Learning Websites

- <https://www.learnpython.org/>
- <https://scikit-learn.org/stable/>
- <https://stackoverflow.com/>