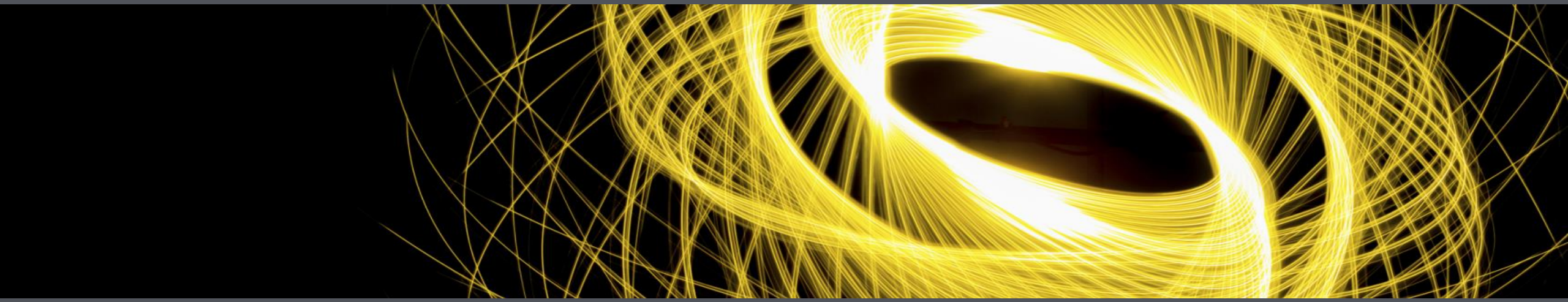
Department of Computer Science

University of Reading

CSMAD21 – Applied Data Science with Python



Python for Data Science - Introduction

Lecture Objectives

- Identify the particularities of Python as a programming language.
- Acknowledge the tools (Anaconda and Jupyter) that are going to be used during the module.
- Understand and implement Jupyter - Markdowns.

Outline

- About Python
- Anaconda and Jupyter
- Markdowns
- Questions and Answers

About Python

- First appeared in 1991 by Guido van Rossum.
- Is written in C.
- Python was designed to be readable and extensible but with a small core.
- One of the most popular interpreted programming language or script language.
- Specially popular after 2005 by its capability for web development with frameworks such as Django.
- In the last 10 year it has developed a large scientific data analysis community.
- Often compared with R, MATLAB and SAS.
- Ease to integrate with C, C++, and FORTRAN.

Why Python?

- Python has some nice features as a programming language:
  - Concise – no bracket vomit!
  - Readable
  - Portable
  - Dynamic typing
  - Supports many programming paradigms
  - Garbage collected (automatic memory management)
  - Large ecosystem
  - Easy to develop prototypes

Why Python?

- Widely used:
  - Second most popular on Github
  - Embedded as a scripting language in games
  - Sources:
    - https://octoverse.github.com/
    - https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages

Why Python?

- Many applications:
  - Data science
  - Web apps
  - Scientific computing
  - Neural networks
- Used for:
  - Shell scripting
  - Frameworks – glue that controls other programs
  - Full applications

Why Python?

- Many applications:
  - Data science
  - Web apps
  - Scientific computing
  - Neural networks
- Used for:
  - Shell scripting
  - Frameworks – glue that controls other programs
  - Full applications

- Why not Python?
  - Because is a interpreted programming language, it will run substantially slower (not the best option for transactional systems).

Indentation and Comments

- In Python indentation acts as blocks do in C or Java:

C or Java:

```
if(x<y)
{
        z = x;
}
else
{
        if(x==y)
        {
                y = y + 1;
        }
        else
        {
                z = y;
        }
 }
```

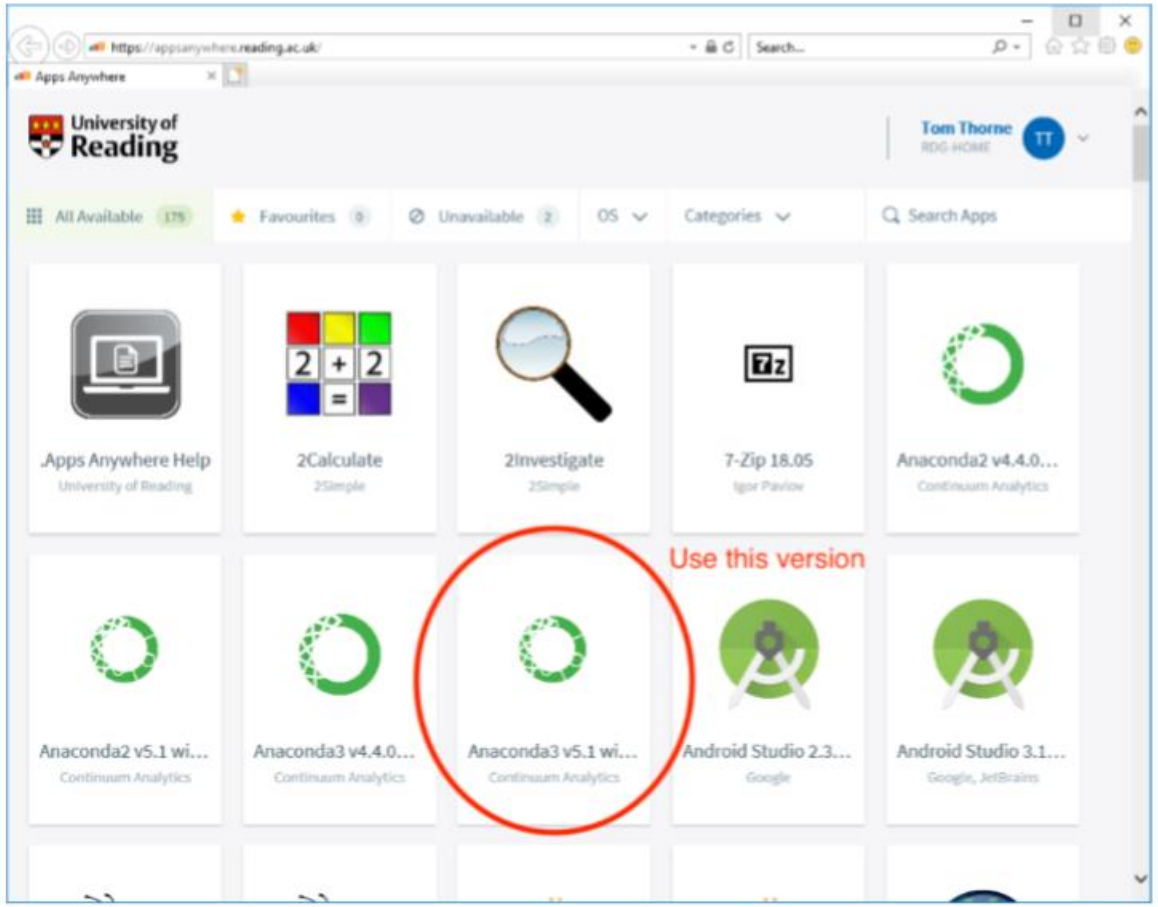
Python:

```
if x<y:
        z = x
else:
        if(x==y):
                y = y + 1
        else:
                z = y

# One line comments

""" Multi
line
Comments """
```

Anaconda Python distribution

- For this module we will be using the Anaconda Python distribution. This includes various Python packages used in scientific programming and data science.
- You can download the distribution for your own machine here:
  - https://www.anaconda.com/download/
  - You should find this comes with all the tools and libraries you need for this course.
- Please stick with version Anaconda 3 with Python 3.6 for the coursework.

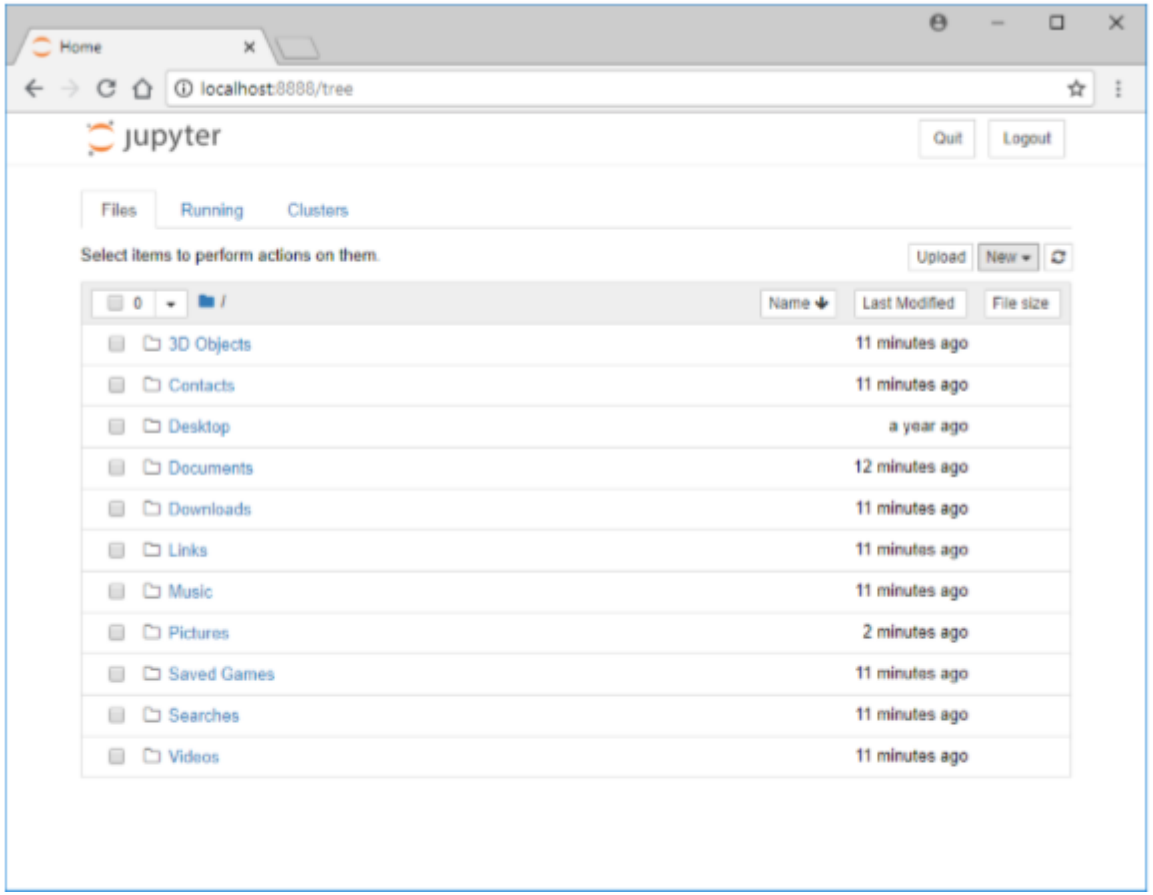
Anaconda in AppsAnywhere



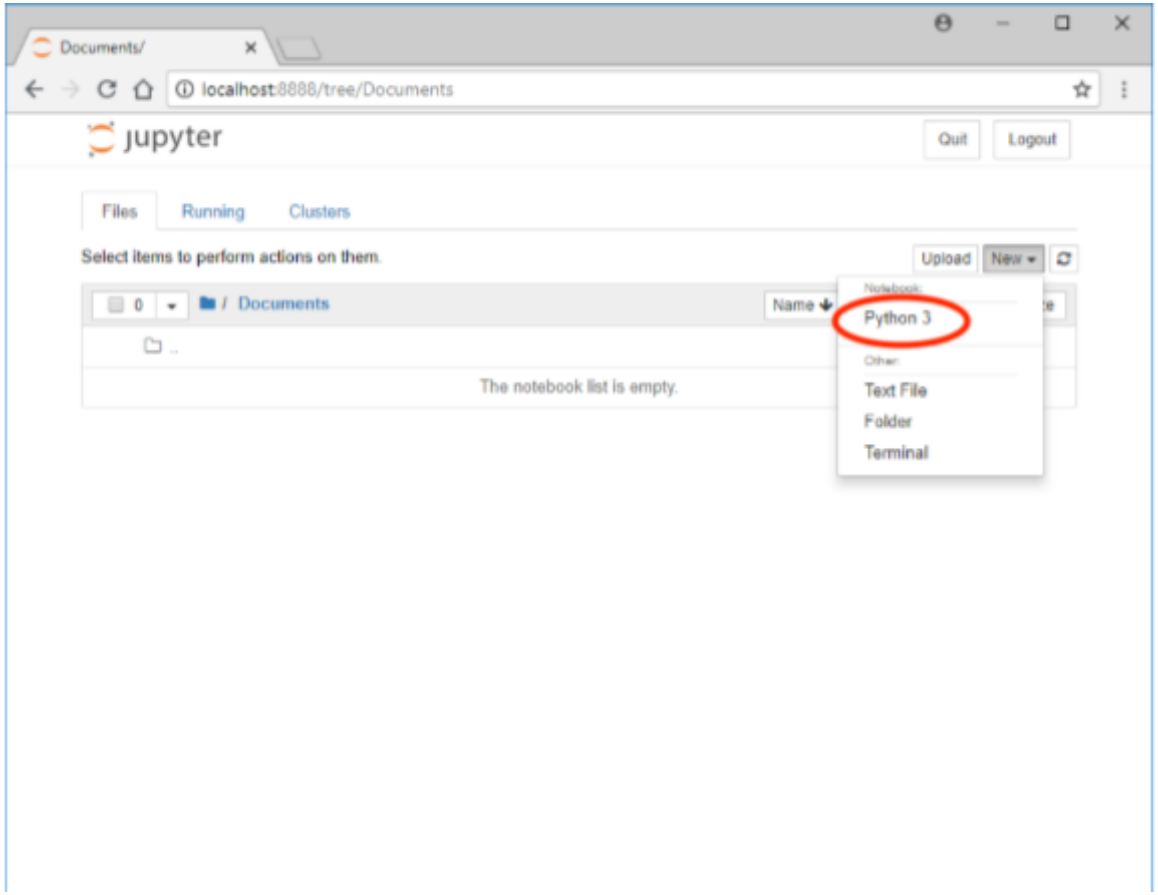
Jupyter

- We will be using a notebook system called Jupyter.
- This allows you to write blocks of code and display their output, as well as formatted text.
  - Reproducibility
  - Easy to share and publish online
- Jupyter runs as a locally hosted web-app that you interact with through a web browser. Jupyter manages running a kernel that executes your code and maintains a global state. This means that:
  - Variable and function definitions persist between blocks.
  - Blocks can be run in any order, and running a block will make a persistent change to the global state.
  - **Write your notebooks to be run from top to bottom!**

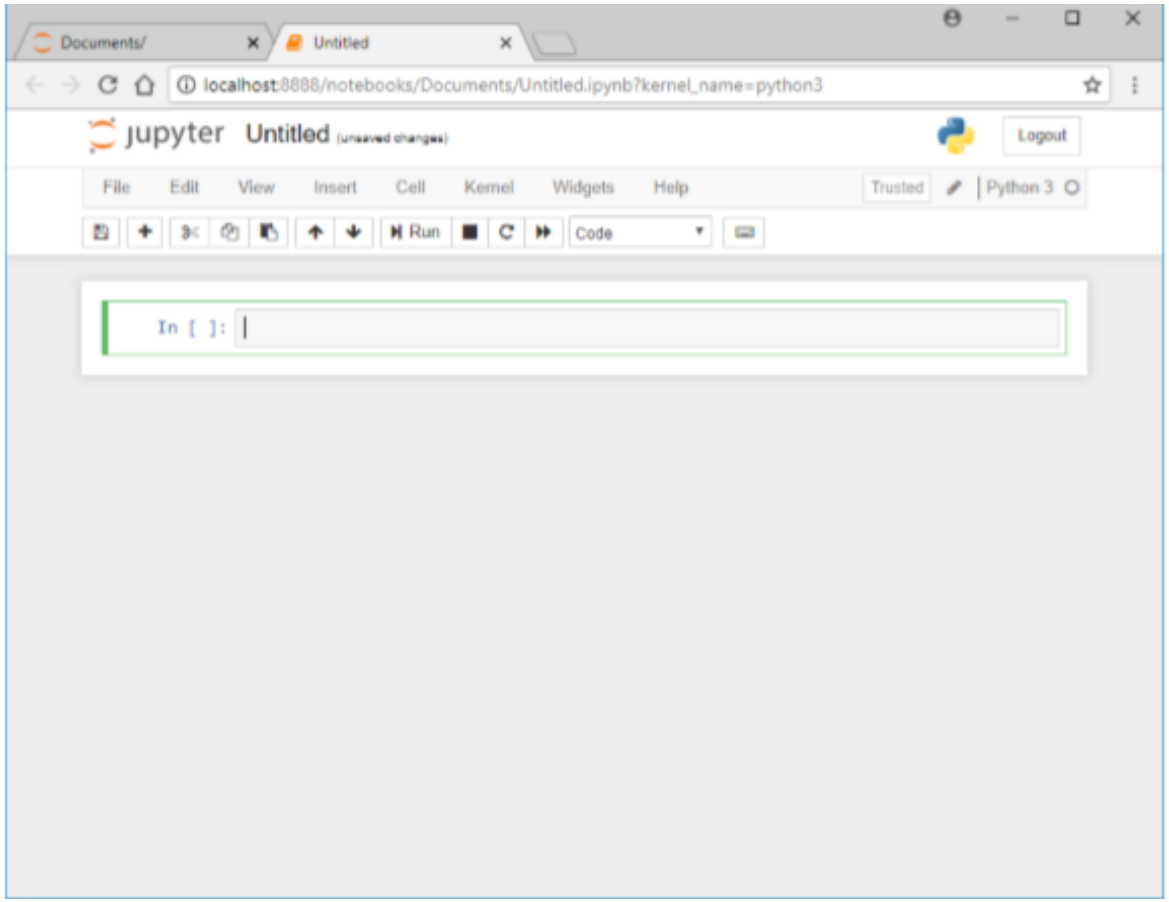
Jupyter


Home
localhost:8888/tree
jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/
Name
Last Modified
File size
3D Objects
11 minutes ago
Contacts
11 minutes ago
Desktop
a year ago
Documents
12 minutes ago
Downloads
11 minutes ago
Links
11 minutes ago
Music
11 minutes ago
Pictures
2 minutes ago
Saved Games
11 minutes ago
Searches
11 minutes ago
Videos
11 minutes ago

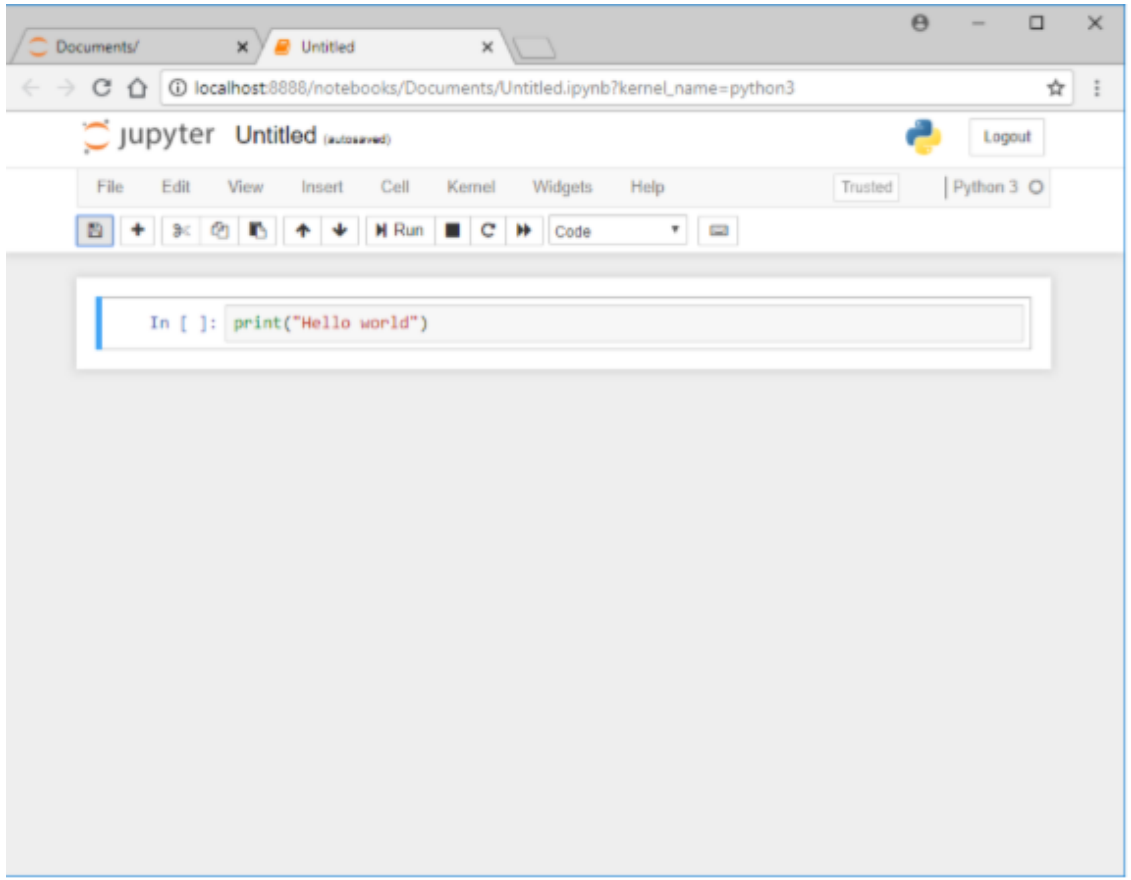
Jupyter


Documents/
localhost:8888/tree/Documents
jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/ Documents
Name
Notebook:
Python 3
Other:
Text File
Folder
Terminal
The notebook list is empty.

Jupyter

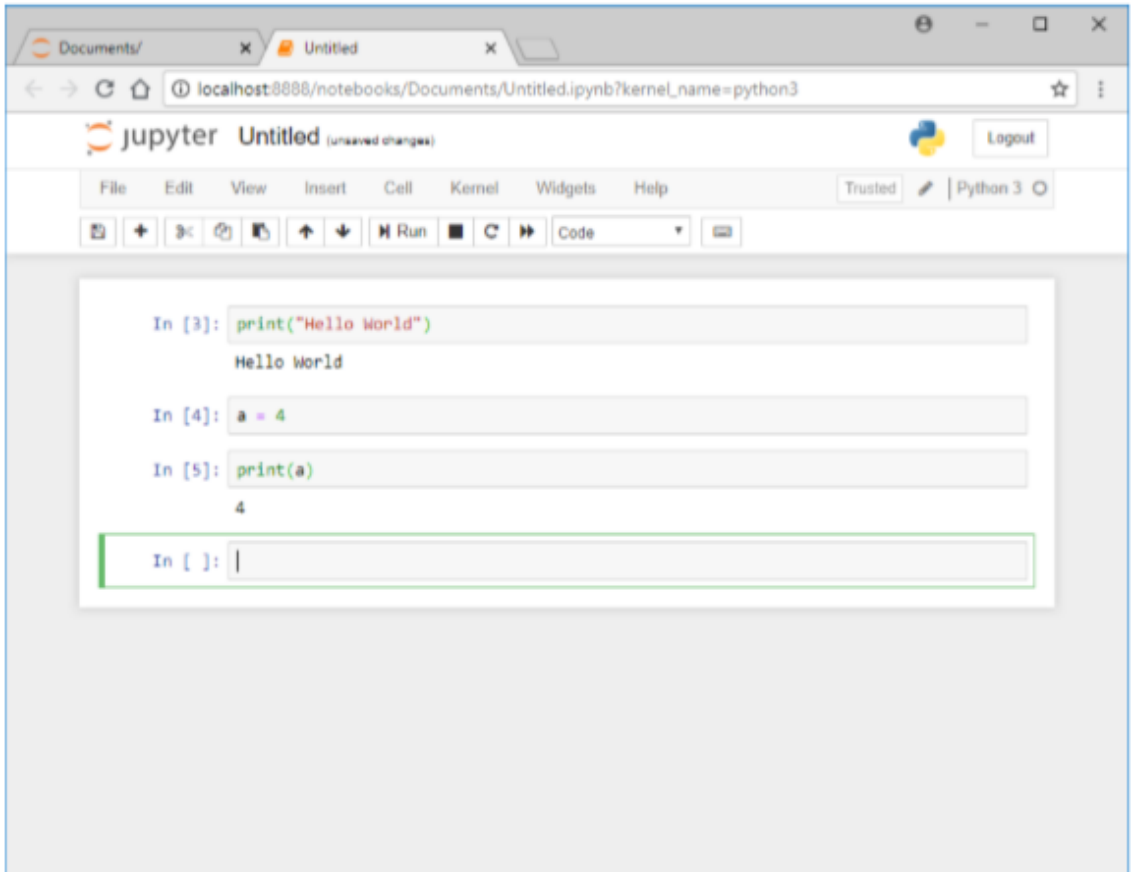

Documents/
Untitled
localhost:8888/notebooks/Documents/Untitled.ipynb?kernel_name=python3
jupyter
Untitled (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
In [ ]:

Jupyter



- To run the code in a block press Shift+Enter (or use the controls at the top of the page)

Jupyter

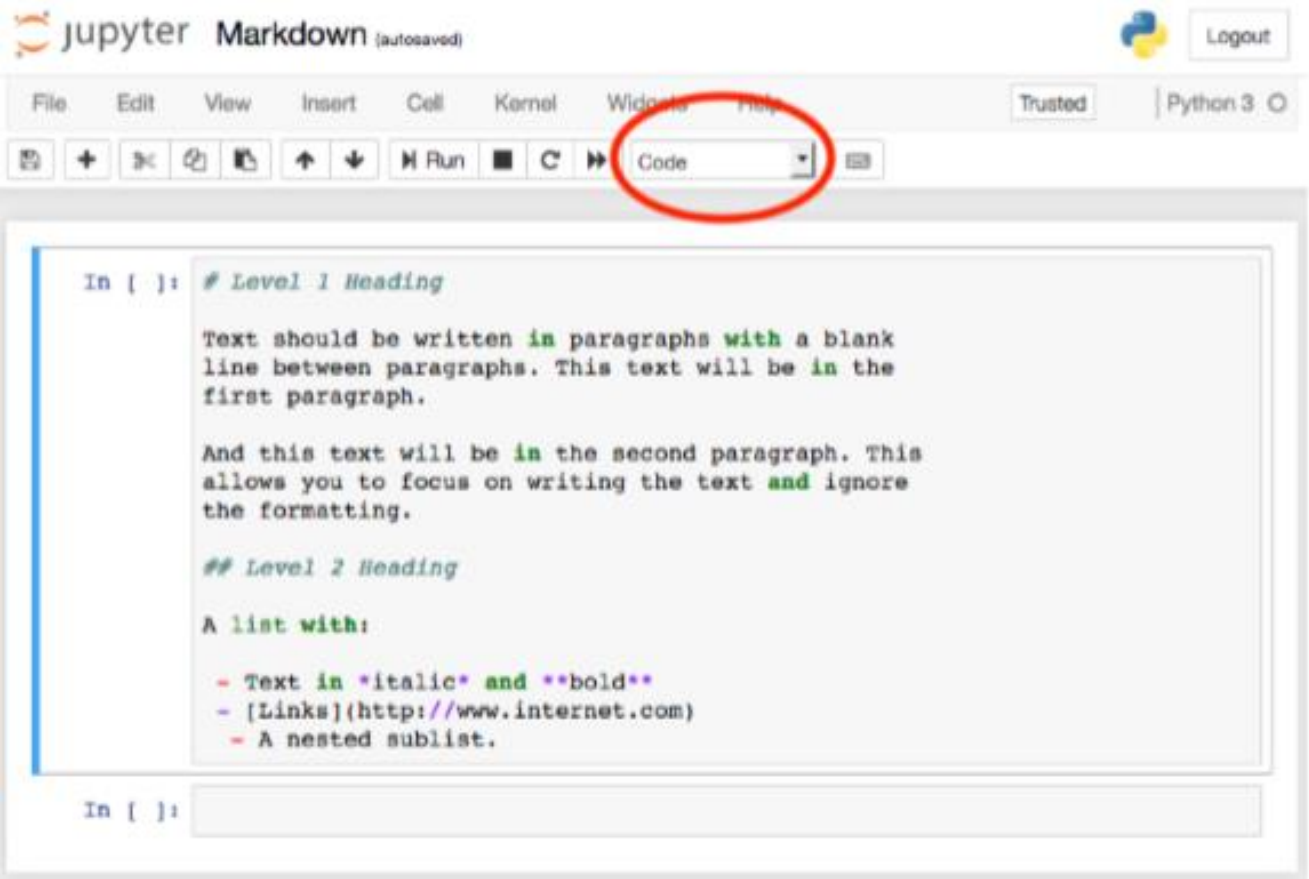

Documents/
Untitled
localhost:8888/notebooks/Documents/Untitled.ipynb?kernel_name=python3
jupyter
Untitled (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
In [3]: print("Hello World")
Hello World
In [4]: a = 4
In [5]: print(a)
4
In [ ]:

Markdowns in Notebooks

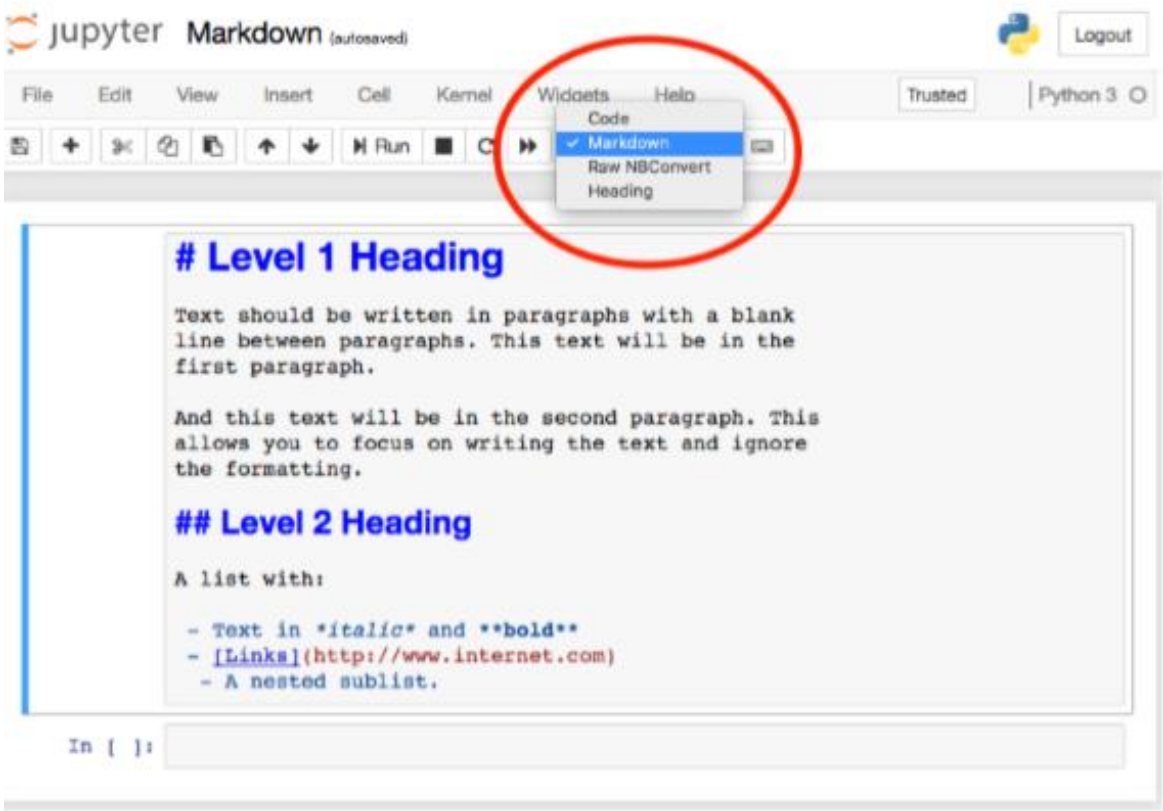
- Jupyter notebooks allow you to have different types of block. So far, we have seen Code blocks, but Jupyter also supports Markdown blocks.

- This allows you to build documents that interleave text, code and plots, which allows you to build reports that contain all of the necessary code to reproduce the results and figures shown.

Markdowns in Notebooks



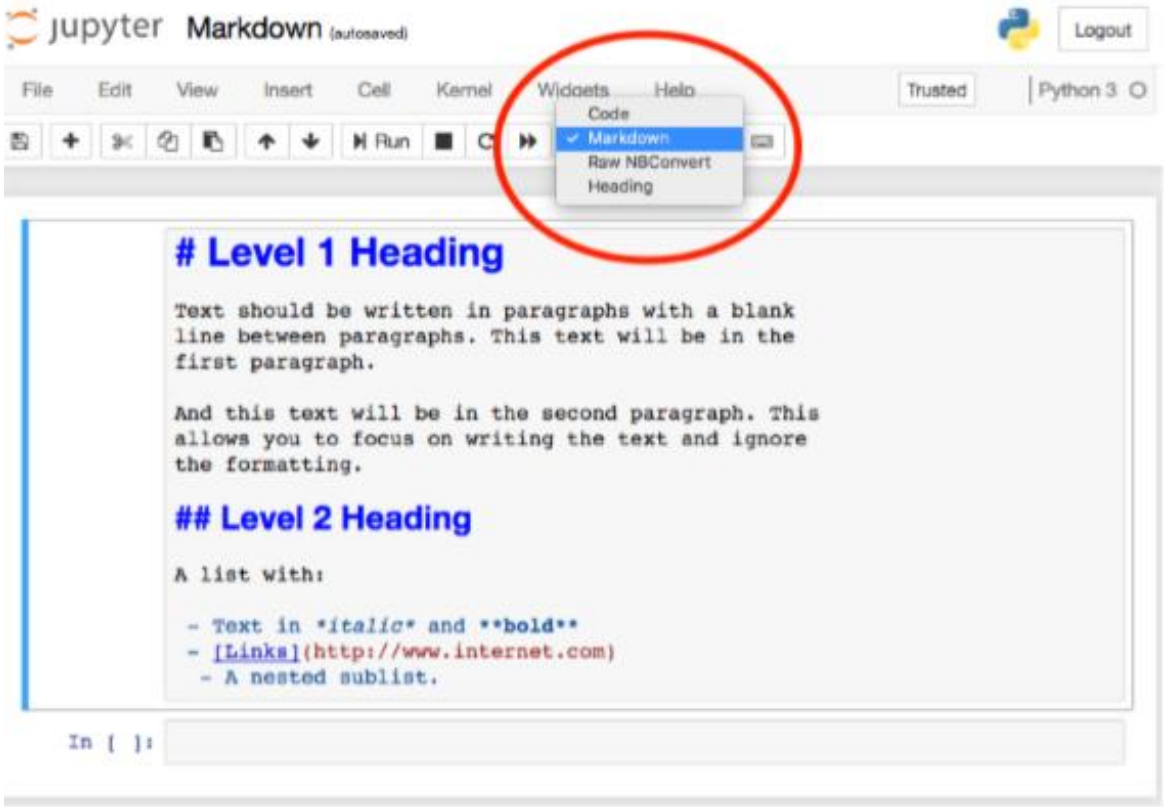
Markdowns in Notebooks



- Set the block to Markdown.

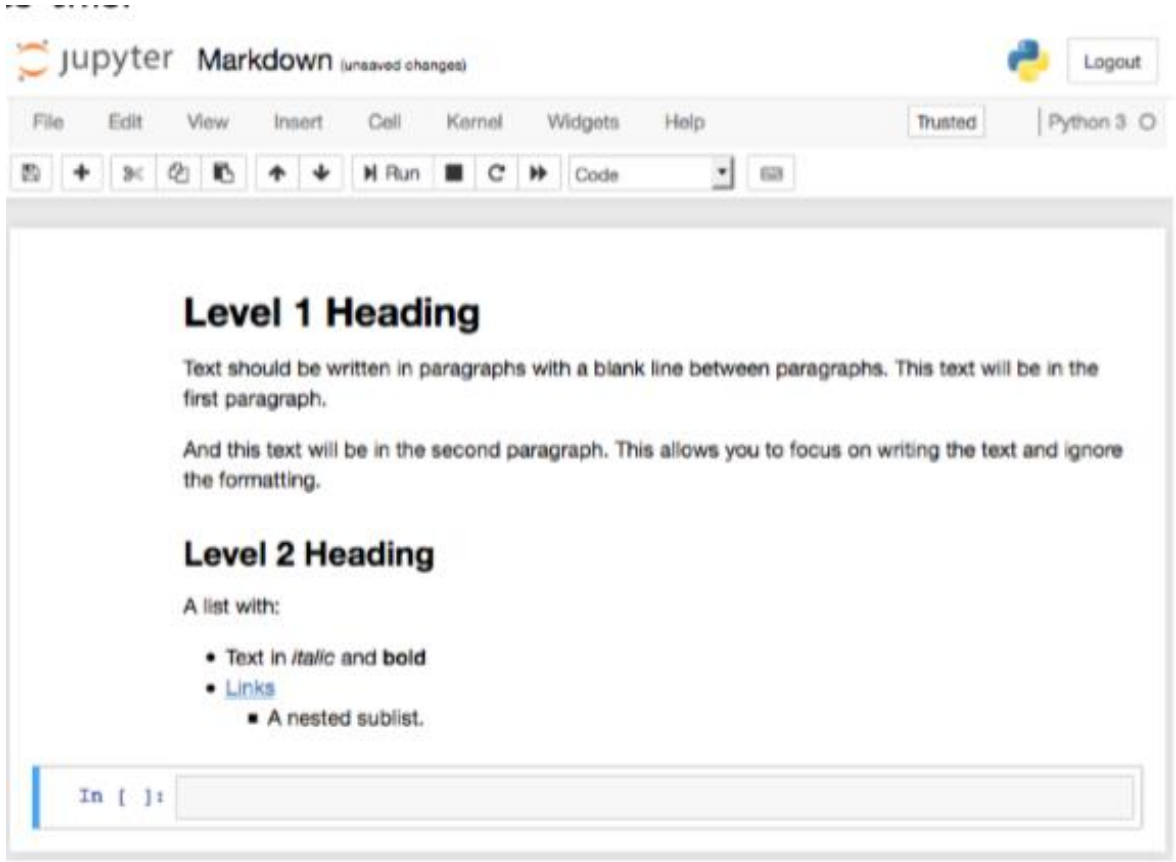
Markdowns in Notebooks

- Markdown is a plain text document markup language. It allows you to make nicely formatted text with simple annotations in standard plain text.
  - So this:



Markdowns in Notebooks

- Become this:



Markdowns Basics

- Headings can be written with #:
  - `# Level 1 Heading`
  - `## Level 2 Heading`
- *Italics* and **bold** can be written with ***** or_
  - `_italic_ or *italic*`
  - `__bold__ or **bold**`
- You can write lists just by leaving a blank line and indenting, followed by **-** :
  - `- A list of items`
  - `  - And subitems`
- More markdowns:
  - https://www.ibm.com/support/knowledgecenter/en/SSGNPV_2.0.0/dsx/markd-jupyter.html

Questions

References

- Thomas McKinney, Wes ‘Python for data analysis : data wrangling with Pandas, NumPy, and IPython’.