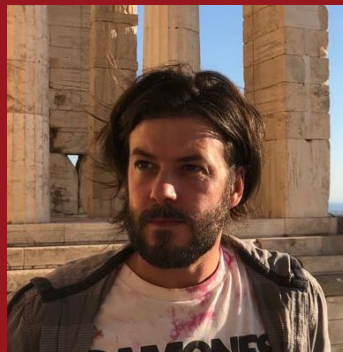


PYTHON WITH ANACONDA DISTRIBUTION

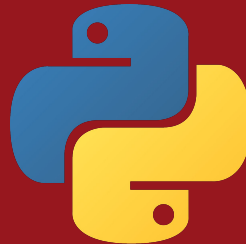
“The world’s most popular
open-source Python
distribution platform”

assembler
institute of technology



Luciano Gabbanelli

Tech Lead Data Science



ÍNDICE



NumPy



SPYDER



1

Download Anaconda / Install Python

2

What is all this stuff?

3

Built-in structures

4

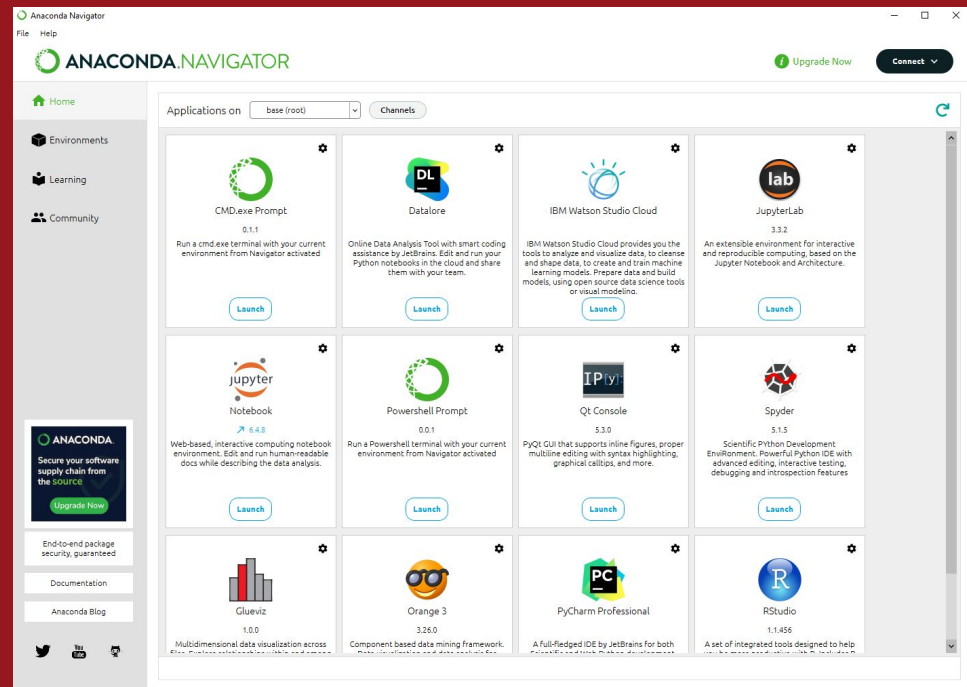
Let's code!!

Go to the Jupyter notebook

2.1-Introduction_to_Python.ipynb

What is ANACONDA.?

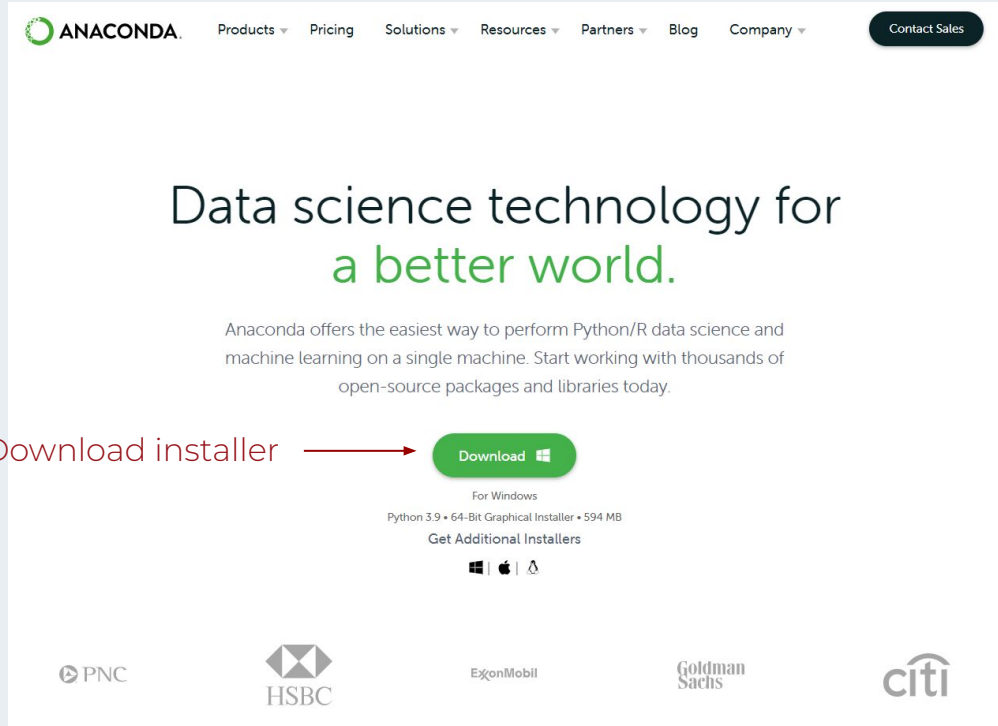
- Distribution of Python and R programming languages
- For scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.)
- aims to simplify package management and deployment.
- Initial release: 2012
- Anaconda Navigator is a desktop graphical user interface (GUI)



INSTALLATION

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

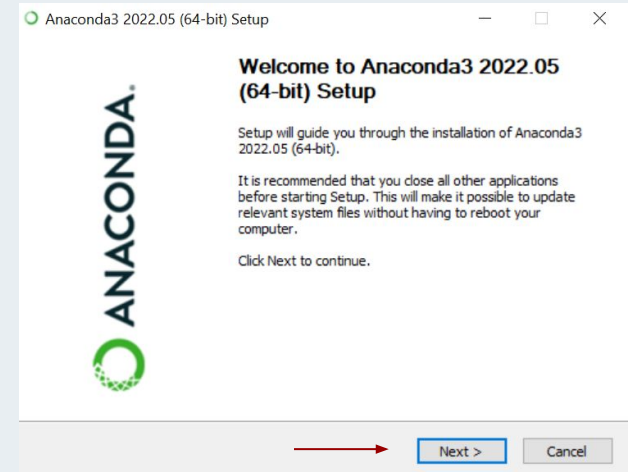
1. Download installer



2. Find the application in Downloads

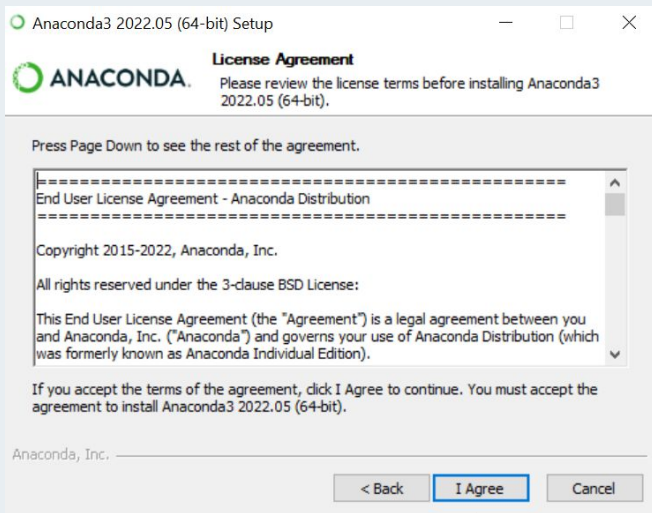
Anaconda3-2022.05-Windows-x86_64

3. Open for installation and Next

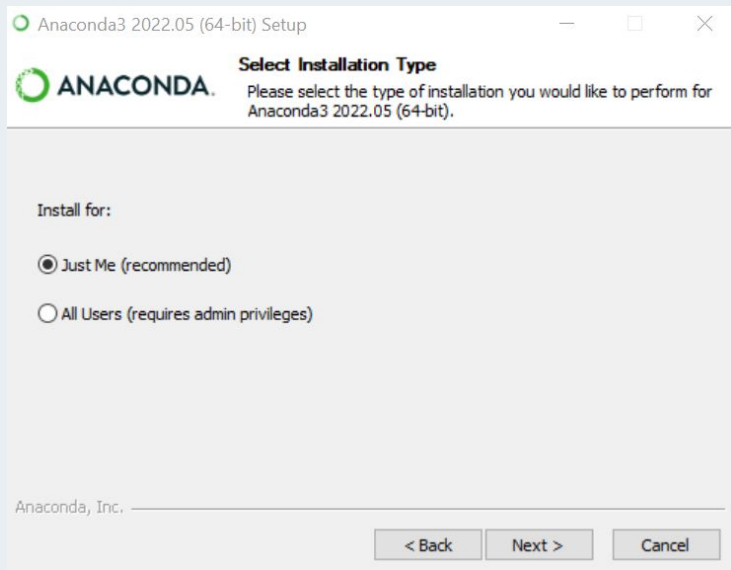


INSTALLATION

4. Agree license

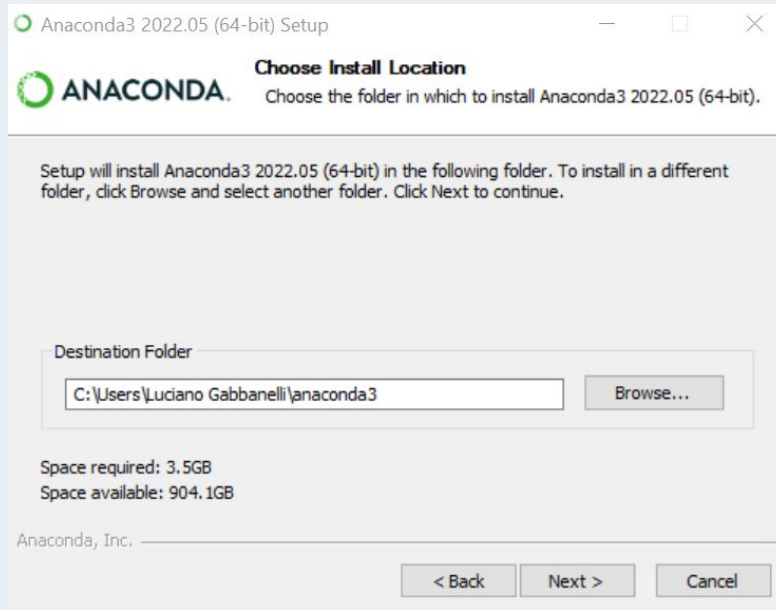


5. Installation type

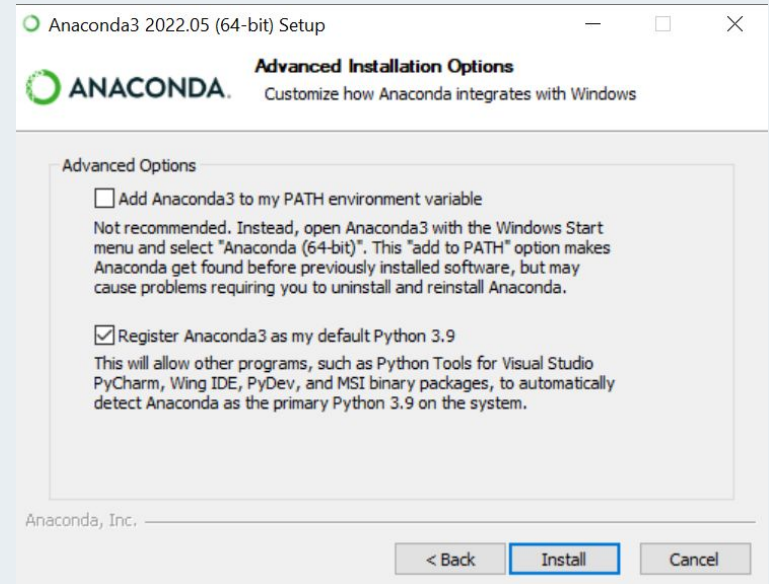


INSTALLATION

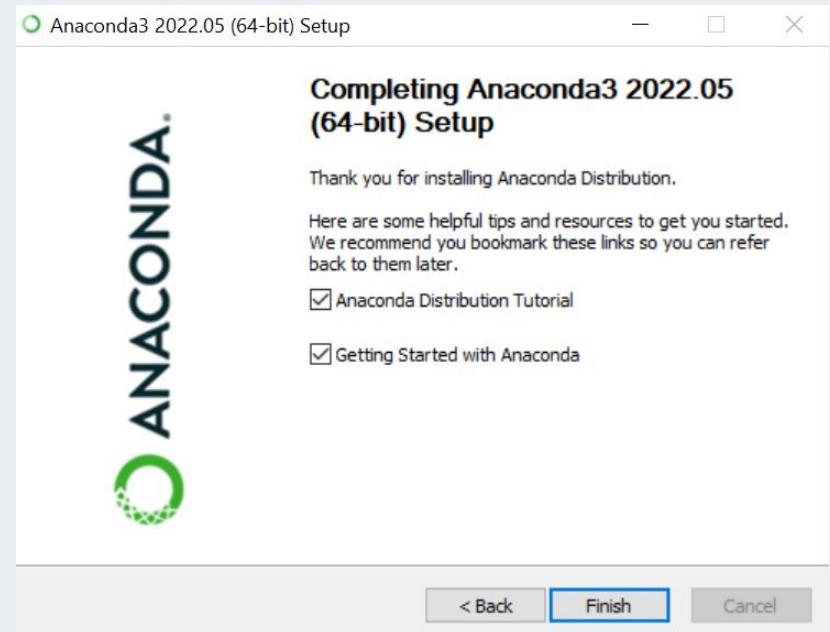
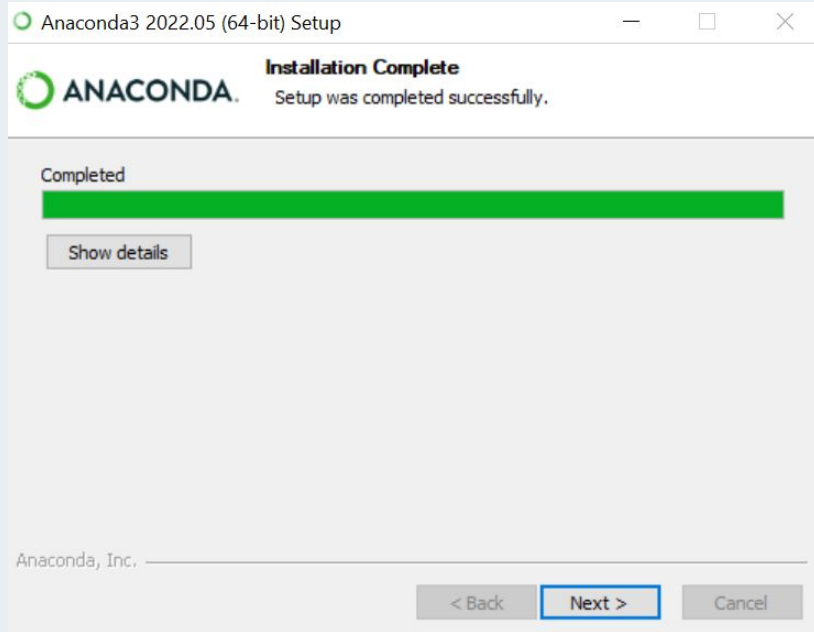
6. Destination Folder: "...\\anaconda3"
avoid spaces in names or replace them by "_"



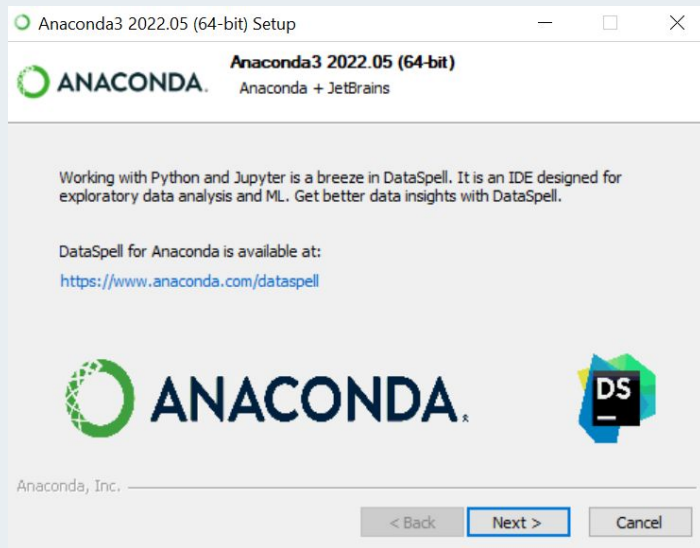
7. Install



INSTALLATION COMPLETE



INSTALLATION COMPLETE



**Let us
start!**

ÍNDICE



NumPy



SPYDER



1

Download Anaconda / Install Python

2

What is all this stuff?

3

Built-in structures

4

Let's code!!

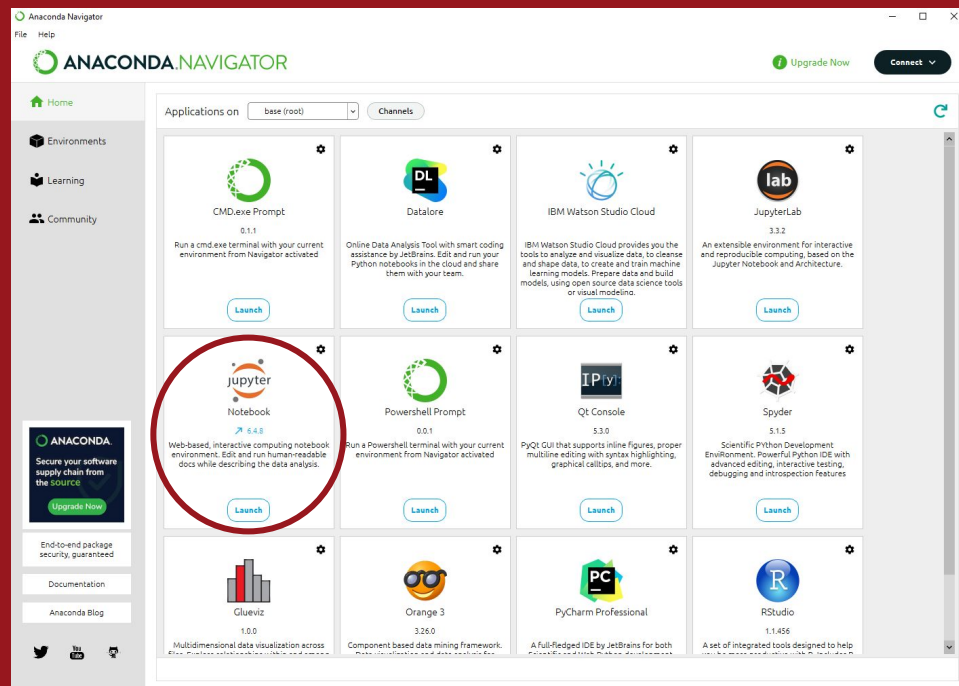
Go to the Jupyter notebook

2.1-Introduction_to_Python.ipynb

What is Python?

(Welcome to Python.org)

- High-level, interpreted, general-purpose programming language
- First release: 1991
- Python 3.0 (2008) not completely backward-compatible with earlier versions
- Main objectives: simplicity and legibility
- Object oriented
- Open source
- Large community
- Large number of standard libraries (scientific computing, text and images processing, game development, etc.)



PYTHON PROGRAMMING

Project Jupyter (Project Jupyter | Home)

is a non-profit, open-source community run project (2014). It evolved to support interactive data science and scientific computing across all programming languages.

Core supported programming languages: **Julia**, **Python**, and **R**.

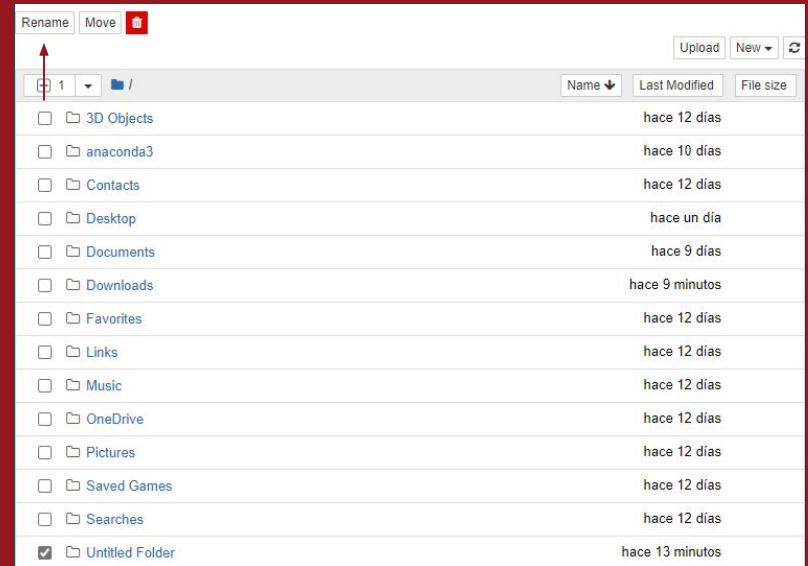
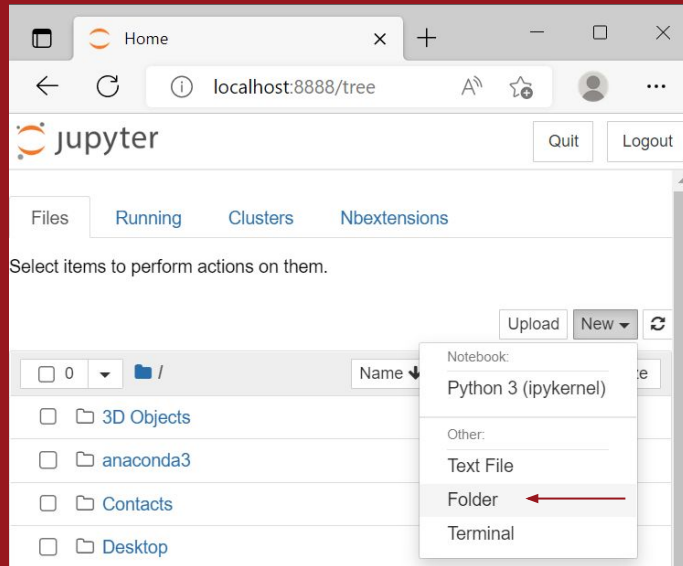
Jupyter Notebook

Jupyter Notebook is an open source web application (interactive interface) for creating and sharing computational documents.

- It works in the localhost (127.0.0.1 this means the IP of your computer), usually in 8888 port and it starts in the tree directory
- Identify the path of this directory in your computer. Usually it is in **C : \ Users \ your_name**
It will be useful for backups
- Jupyter Notebook files are notebooks

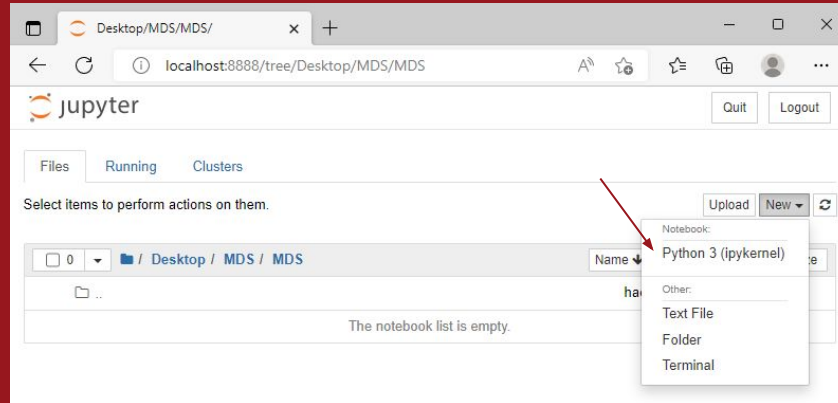
PREPARING THE WORKSPACE

- Create a folder for your project
- Change its name: Untitled Folder → MDS
- Go to the folder you created



PREPARING THE WORKSPACE

- Once you are inside the folder, create a notebook file. It will be created in the path : `/ MDS`

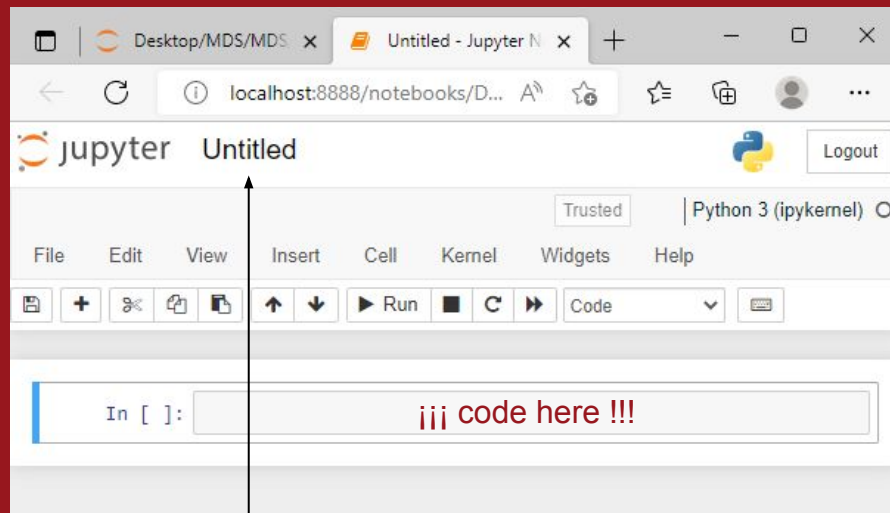
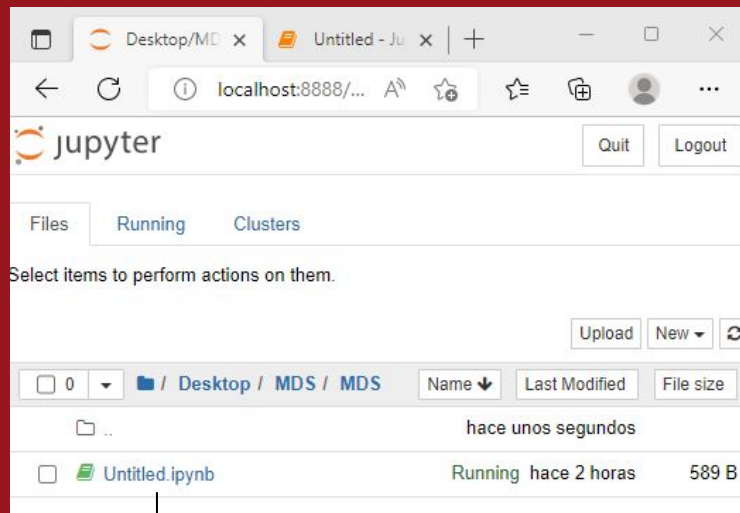


The extension of notebooks is `.ipynb`, which means **Interactive PYTHON NoteBook**

Other interesting possibilities:

- JupyterLab** (Anaconda)
- Google Colaboratory** (<https://research.google.com/colaboratory/>): similar to Google Docs / Google Sheets. Online notebooks. No need to install.
- There are also `.py` which are regular python files, usually executed in Spyder. It's plain text and contains just your code.


THE CLASSIC NOTEBOOK INTERFACE



Change the name to the .ipynb file: `Untitled.ipynb` → `1.2-First_Jupyter_Notebook.ipynb`

ZEN OF PYTHON

- **Change the name of the file:** Untitled.ipynb → 1.2-First_Jupyter_Notebook.ipynb
- **Create a title for the notebook:** # Introduction to Python
TIP: Code → Markdown (easy-to-read text format)

- Run cells using: Shift + Intro (≡  Run) or Ctrl + Intro or Alt. + Intro. **What's the difference?**
- Try: Esc. and/or Intro over cells. Change between command (blue) / edit (green) mode
- Efficient programming: Take advantage of the shortcuts!!!
- In Esc. + M → markdown Y → code
(command mode) A → insert cell above B → insert cell below
- # → Tells the Python interpreter to ignore the rest of the line. **Try it!**
Comments are sensefull only in Code mode. **Why?**
- Our first code: Type → **import** this

CORE PHILOSOPHY OF PYTHON

19 "guiding principles"
for writing
computer programs

- Check the Python version: `!python --version`

The exclamation mark operator allows to execute commands on the underlying operating system (i.e. on shell)

ONLY IN JUPYTER NOTEBOOK

- Introduce the number 4 and execute

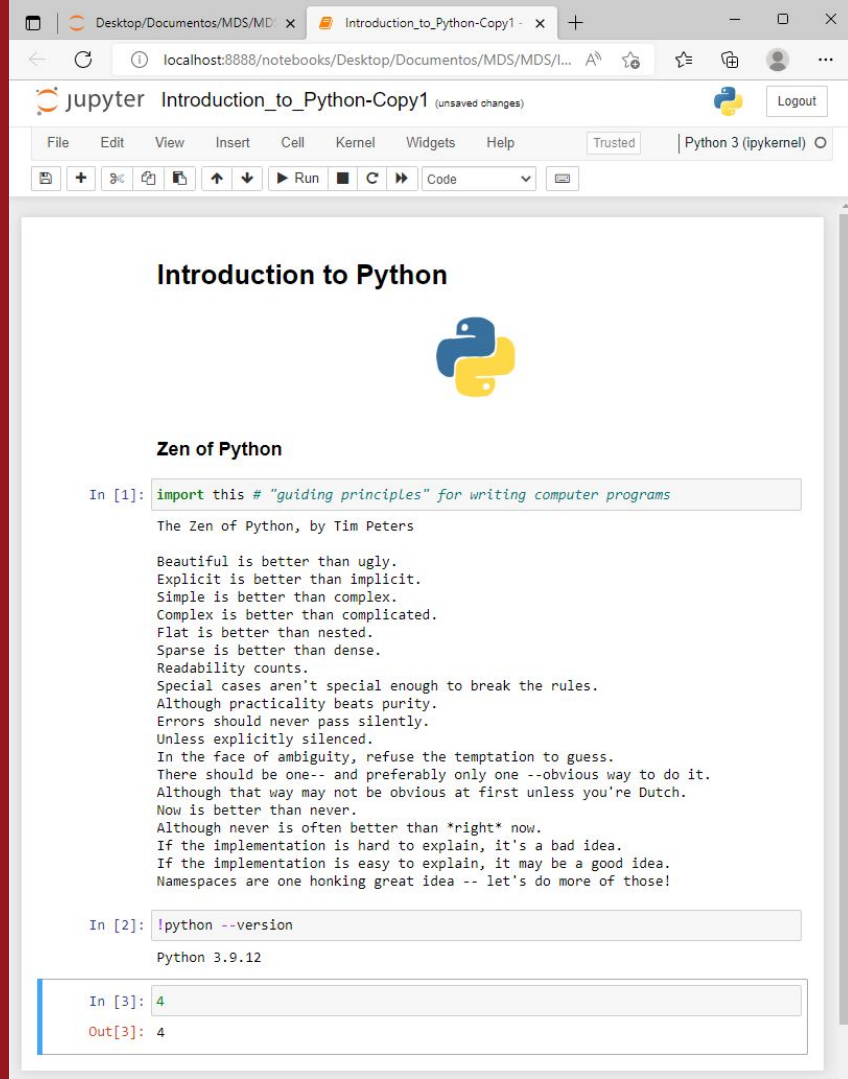
When do we have “Out” and when we do not?

- Remember again:

*If you start doing some action with the mouse,
stop and think if there is a shortcut.*

Esc. + H or Help → keyboard shortcuts

start/windows + right/left keys full screen in half screen




Desktop/Documents/MDS/MD Introduction_to_Python-Copy1 - x

localhost:8888/notebooks/Desktop/Documents/MDS/MDS/I... A ☆ ↺

jupyter Introduction_to_Python-Copy1 (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Introduction to Python



Zen of Python

In [1]: `import this # "guiding principles" for writing computer programs`

The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

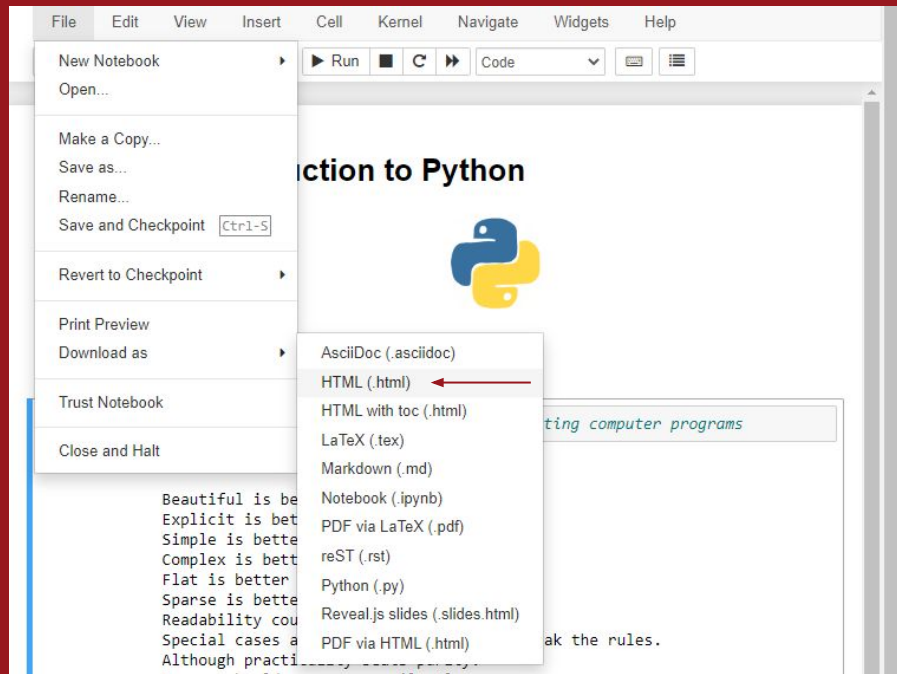
In [2]: `!python --version`

Python 3.9.12

In [3]: `4`

Out[3]: 4

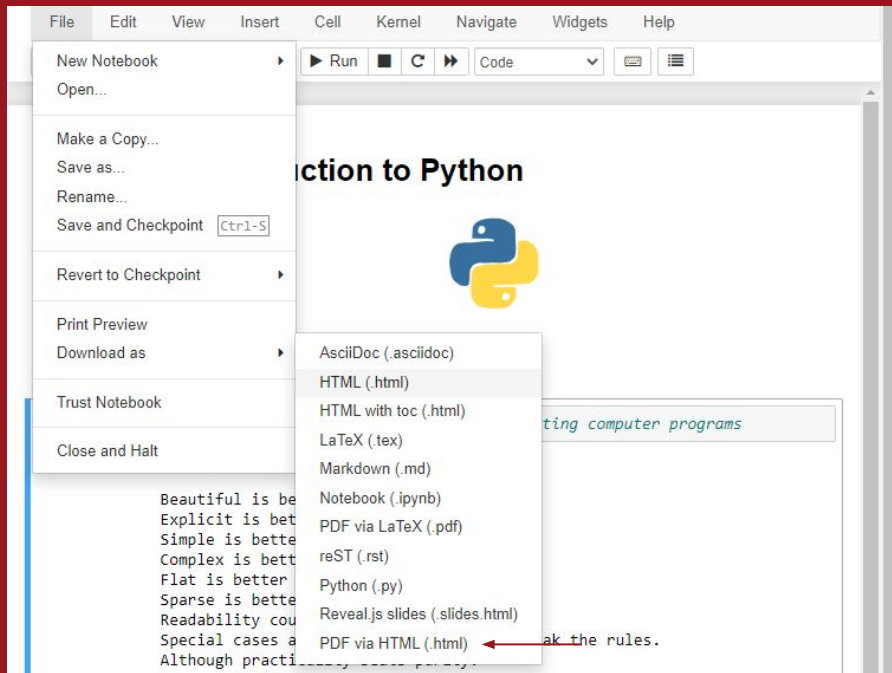
CONVERT .ipynb TO .html



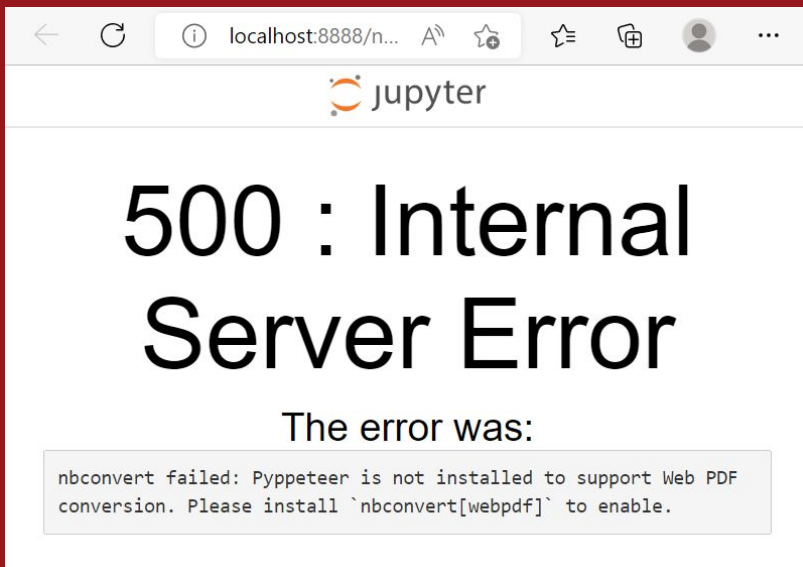
**Your file is probably
in the download
folder**

Open it!

CONVERT .ipynb TO .pdf



You will probably get this error



nbconvert[webpdf]

Google it!!

Choose the link

Nbconvert Webpdf :: Anaconda.org

Or go directly to

<https://anaconda.org/conda-forge/nbconvert-webpdf>

nbconvert-webpdf

The screenshot shows the Anaconda Forge package page for 'nbconvert-webpdf' version 6.5.0. The page is titled 'conda-forge / packages / nbconvert-webpdf 6.5.0' and includes a star icon and a '0' badge. Below the title, it says 'copied from cf-staging / nbconvert-webpdf'. There are tabs for 'Conda', 'Files', 'Labels', and 'Badges'. Under the 'Conda' tab, there is a section with the following information: 'License: Unspecified', '1416 total downloads', and 'Last upload: 3 months and 10 days ago'. Below this is a section titled 'Installers' with the command 'conda install' and a help icon. Under 'Installers', there is a button for 'noarch v6.5.0'. Below the button, there is a text box that says 'To install this package with conda run one of the following:'. Below the text box, there are two code snippets: 'conda install -c conda-forge nbconvert-webpdf' and 'conda install -c conda-forge/label/broken nbconvert-webpdf'. Two red arrows point to the right from the end of each code snippet. Below the code snippets is a section titled 'Description'.

conda-forge / packages / nbconvert-webpdf 6.5.0 ☆ 0

copied from cf-staging / nbconvert-webpdf

Conda Files Labels Badges

License: Unspecified
1416 total downloads
Last upload: 3 months and 10 days ago

Installers
conda install ?

noarch v6.5.0

To install this package with conda run one of the following:

```
conda install -c conda-forge nbconvert-webpdf
```

```
conda install -c conda-forge/label/broken nbconvert-webpdf
```

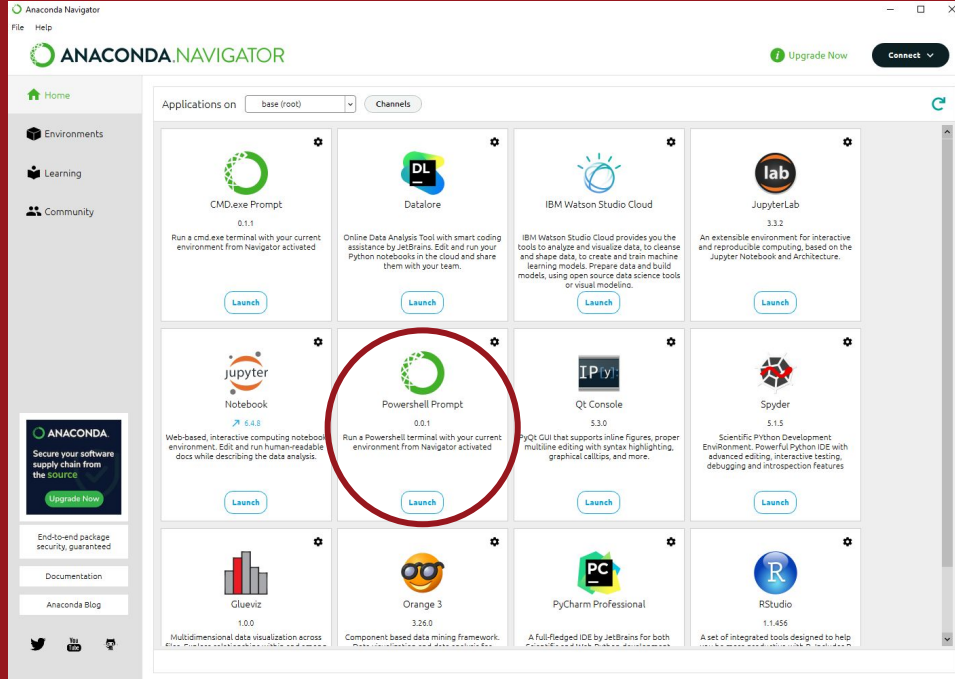
Description

Let us execute these
two commands in
the Powershell
Prompt of
Anaconda

Powershell Prompt

Lunch it!

Then run the commands

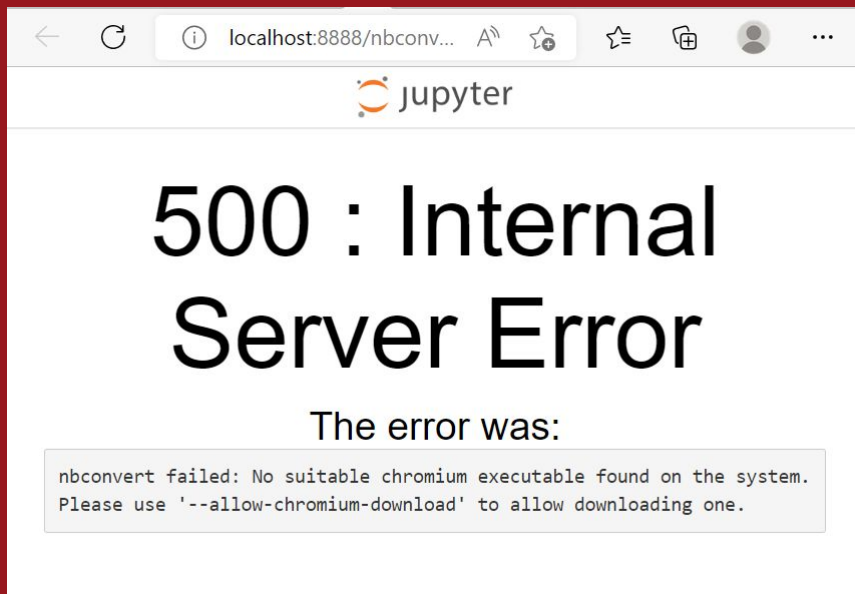


```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
(base) PS C:\Users\Luciano Gabbaneli> conda install -c conda-forge nbconvert-webpdf
collecting package metadata (current_repodata.json): /
```

**When the terminal asks to proceed,
say Yes :)**

**Try to download again: PDF via HTML
(.html)**

Oh, not again!! :(



Open the anaconda **terminal**.

Type the command

pyppeteer-install

in the **terminal**. This will download and install an appropriate version of **chromium**.

Try to download the notebook again :O

PDF via HTML (.html)

VICTORY?

ÍNDICE



NumPy



SPYDER



1

Download Anaconda / Install Python

2

What is all this stuff?

3

Built-in structures

4

Let's code!!

Go to the Jupyter notebook

2.1-Introduction_to_Python.ipynb

WHAT IS A LANGUAGE?

Noam Chomsky: a language is a set of (finite or infinite) sentences, each finite length constructed out of a limited set of elements.

In-built objects:

- **int** (integer): integer numbers.
- **float**: floating point number (represent " \mathbb{R} -numbers")
- **complex** : complex numbers
- **str** (string) : text or string of characters
- **bool**: boolean. Data with only two possible values (usually True or False). Represent the two truth values of logic and Boolean algebra.
- **list** : an ordered heterogeneous collection of objects.
- **tuple**: an immutable ordered heterogeneous collection of objects.
- **set** : an unordered heterogeneous collection of unique objects.
- **dict** (dictionary) : store data values in (key : value) pairs mapping.

`type(...)` will help from now on!



ÍNDICE



NumPy



SPYDER



1

Download Anaconda / Install Python

2

What is all this stuff?

3

Built-in structures

4

Let's code!!

Go to the Jupyter notebook

2.1-Introduction_to_Python.ipynb