

NOVA

IMS

Information
Management
School

Programming for Data Science

1st Session: Anaconda, Jupyter Notebooks
Introduction to python

Flávio Pinheiro

fpinheiro@novaims.unl.pt

Liah Rosenfeld

lrosenfeld@novaims.unl.pt

Maria Almeida





malmeida@novaims.unl.pt

Niclas Sturm

nsturm@novaims.unl.pt

Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Class topics

-  **Getting familiar**
-  **Installing Anaconda**
-  **Getting familiar with Jupyter notebook and creating our first notebook**
-  **Printing our first statements!**

The practical classes team 😊



Liah Rosenfeld

lrosenfeld@novaims.unl.pt



Maria Almeida





malmeida@novaims.unl.pt



Niclas Sturm

nsturm@novaims.unl.pt

Class topics

-  Getting familiar
-  Installing Anaconda
-  Getting familiar with Jupyter notebook and creating our first notebook
-  Printing our first statements!



Open Source

Anaconda Individual Edition is the world's most popular Python distribution platform with over 20 million users worldwide. You can trust in our long-term commitment to supporting the Anaconda open-source ecosystem, the platform of choice for Python data science.



Conda Packages

Search our cloud-based repository to find and install over 7,500 data science and machine learning packages. With the conda-install command, you can start using thousands of open-source Conda, R, Python and many other packages.



Manage Environments

Individual Edition is an open source, flexible solution that provides the utilities to build, distribute, install, update, and manage software in a cross-platform manner. Conda makes it easy to manage multiple data environments that can be maintained and run separately without interference from each other.

Installing Anaconda (MacOS)

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

Anaconda Installers

Windows 

Python 3.8

64-Bit Graphical Installer (466 MB)

32-Bit Graphical Installer (397 MB)

MacOS 

Python 3.8

64-Bit Graphical Installer (462 MB)

64-Bit Command Line Installer (454 MB)

Linux 

Python 3.8

64-Bit (x86) Installer (550 MB)

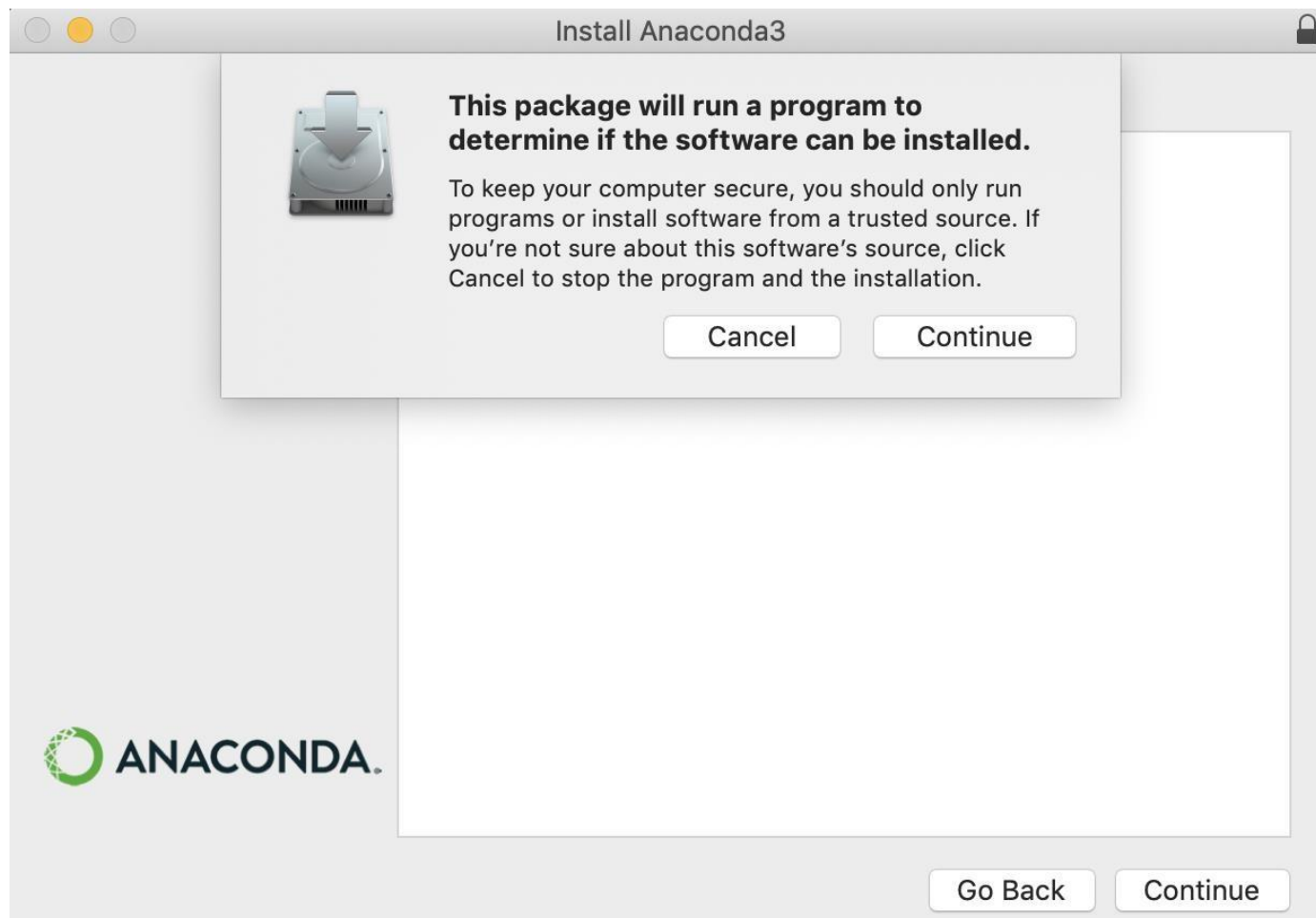
64-Bit (Power8 and Power9) Installer (290 MB)



Anaconda3-20....pkg
324/462 MB, 5 secs left







Installing Anaconda (MacOS)

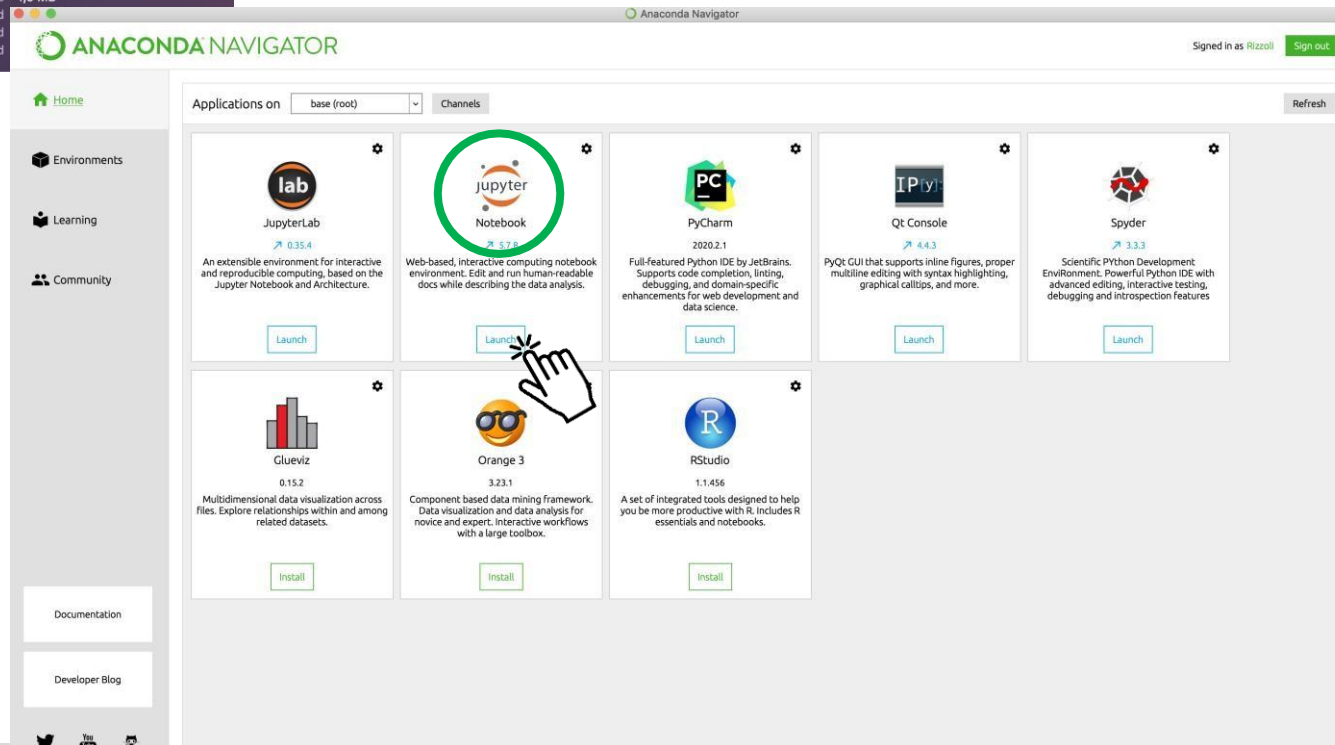


Continue...next...

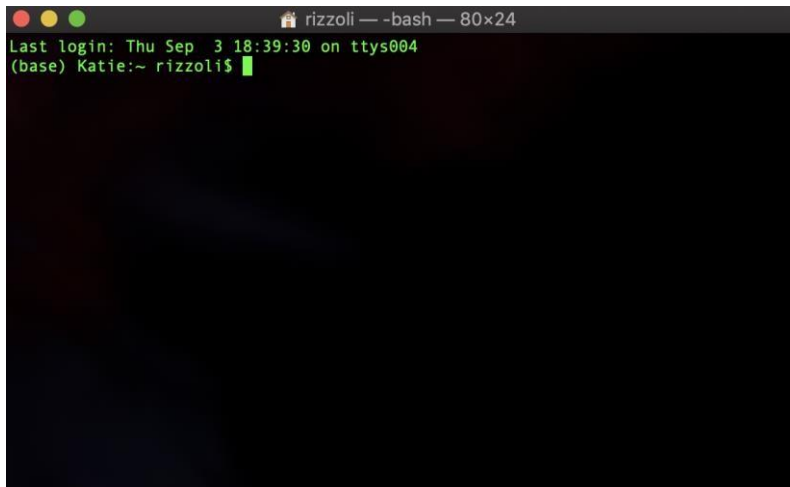
Class topics

-  Getting familiar
-  Installing Anaconda
-  Getting familiar with Jupyter notebook and creating our first notebook
-  Printing our first statements!

Anaconda Navigator and jupyter notebook

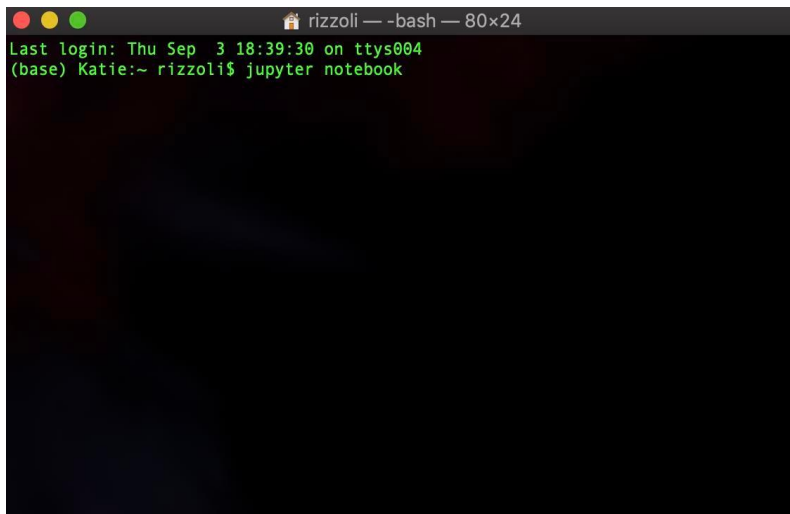


Another way to launch jupyter notebook...



```
rizzoli — -bash — 80x24
Last login: Thu Sep 3 18:39:30 on ttys004
(base) Katie:~ rizzoli$
```

Open terminal



```
rizzoli — -bash — 80x24
Last login: Thu Sep 3 18:39:30 on ttys004
(base) Katie:~ rizzoli$ jupyter notebook
```

Type jupyter notebook and click enter

What you should see...



Quit

Logout

Files **Running** Clusters

Select items to perform actions on them.

Upload

New ▾



<input type="checkbox"/> 0 ▾	/	Name ▾	Last Modified	File size
<input type="checkbox"/>	Applications		a year ago	
<input type="checkbox"/>	Desktop		15 minutes ago	
<input type="checkbox"/>	Documents		2 months ago	
<input type="checkbox"/>	Downloads		16 minutes ago	
<input type="checkbox"/>	Dropbox		4 months ago	
<input type="checkbox"/>	eclipse		a year ago	
<input type="checkbox"/>	eclipse-workspace		10 months ago	
<input type="checkbox"/>	google-cloud-sdk		a year ago	
<input type="checkbox"/>	IdeaProjects		a year ago	
<input type="checkbox"/>	Movies		a year ago	
<input type="checkbox"/>	Music		a year ago	
<input type="checkbox"/>	nltk_data		a year ago	
<input type="checkbox"/>	Pictures		a year ago	
<input type="checkbox"/>	Public		a year ago	
<input type="checkbox"/>	PycharmProjects		a year ago	
<input type="checkbox"/>	seaborn-data		9 months ago	
<input type="checkbox"/>	roulette_stats.ipynb		10 months ago	13.3 kB
<input type="checkbox"/>	testings.ipynb		9 months ago	1.82 kB
<input type="checkbox"/>	Untitled.ipynb		9 months ago	2.54 kB

You may choose a folder:



Quit

Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ↕

<input type="checkbox"/> 0 ▾	/	Name ▾	Last Modified	File size
<input type="checkbox"/>	Applications		a year ago	
<input type="checkbox"/>	Desktop		15 minutes ago	
<input type="checkbox"/>	Documents		2 months ago	
<input type="checkbox"/>	Downloads		16 minutes ago	
<input type="checkbox"/>	Dropbox		4 months ago	
<input type="checkbox"/>	eclipse		a year ago	
<input type="checkbox"/>	eclipse-workspace		10 months ago	
<input type="checkbox"/>	google-cloud-sdk		a year ago	
<input type="checkbox"/>	IdeaProjects		a year ago	
<input type="checkbox"/>	Movies		a year ago	
<input type="checkbox"/>	Music		a year ago	
<input type="checkbox"/>	nltk_data		a year ago	
<input type="checkbox"/>	Pictures		a year ago	
<input type="checkbox"/>	Public		a year ago	
<input type="checkbox"/>	PycharmProjects		a year ago	
<input type="checkbox"/>	seaborn-data		9 months ago	
<input type="checkbox"/>	roulette_stats.ipynb		10 months ago	13.3 kB
<input type="checkbox"/>	testings.ipynb		9 months ago	1.82 kB
<input type="checkbox"/>	Untitled.ipynb		9 months ago	2.54 kB

Then create a new notebook:



Quit

Logout

Files

Running

Clusters

Select items to perform actions on them.

Upload

New

0 / Documents			Name ↓	Last Modified	File size
<input type="checkbox"/>	..			seconds ago	
<input type="checkbox"/>	bol			5 months ago	
<input type="checkbox"/>	Debug			5 months ago	
<input type="checkbox"/>	EmbeddedGSGP			4 months ago	
<input type="checkbox"/>	EmbeddedGSGP_grid			3 months ago	
<input type="checkbox"/>	papers_tese			2 months ago	
<input type="checkbox"/>	random_sheit			10 months ago	
<input type="checkbox"/>	Raw_code			3 months ago	
<input type="checkbox"/>	Raw_code_no_grid			3 months ago	
<input type="checkbox"/>	practical_exam_BDA_2020.ipynb			3 months ago	427 kB
<input type="checkbox"/>	testings.ipynb			6 months ago	124 kB
<input type="checkbox"/>	BIOAVAILABILITY.txt			a year ago	248 kB
<input type="checkbox"/>	CONCRETE.txt			8 months ago	40.3 kB
<input type="checkbox"/>	Debug.zip			5 months ago	196 MB
<input type="checkbox"/>	EmbeddedGSGP_grid.zip			4 months ago	5.89 GB
<input type="checkbox"/>	ENERGY.txt			8 months ago	12.4 kB
<input type="checkbox"/>	PPB.txt			7 months ago	447 kB
<input type="checkbox"/>	Raw_code.zip			3 months ago	163 MB
<input type="checkbox"/>	Raw_code_no_grid.zip			3 months ago	1.16 MB
<input type="checkbox"/>	Screenshot 2020-03-27 at 23.45.43.png			5 months ago	1.81 MB
<input type="checkbox"/>	Screenshot 2020-03-29 at 20.37.28.png			5 months ago	1.6 MB
<input type="checkbox"/>	Screenshot 2020-03-29 at 20.37.40.png			5 months ago	43.7 kB

Our first notebook is here 😊

The screenshot shows the JupyterLab interface. At the top right, there are buttons for 'Upload', 'New', and a refresh icon. A dropdown menu is open under the 'New' button, showing options: 'Notebook:', 'Python 3*' (highlighted with a hand cursor), 'Other:', 'Text File', 'Folder', and 'Terminal'. Below the menu, there is a 'Logout' button and a Python logo. The main area shows the Jupyter logo, the text 'Untitled', and 'Last Checkpoint: a few seconds ago (unsaved changes)'. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. A toolbar contains icons for saving, creating new files, deleting, copying, pasting, undo, redo, and running cells. The main workspace shows a code cell with the prompt 'In []: |'.

Rename Notebook

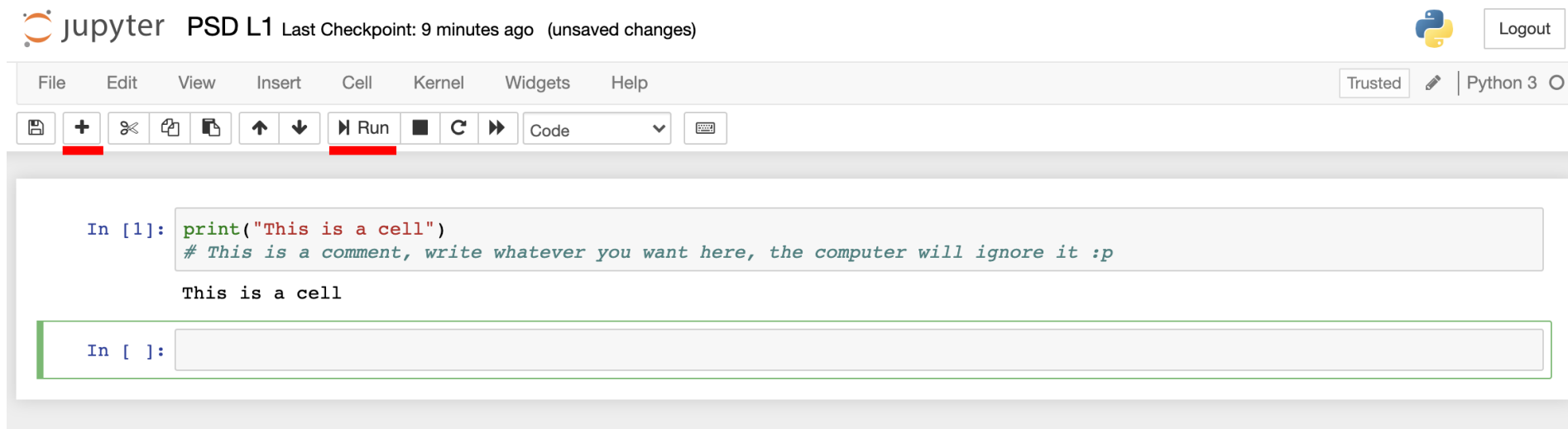
Enter a new notebook name:

PSD L1|

Cancel

Rename

Let's get familiar with the notebook...





You can run a cell (“perform”/ “activate” / run) your code by clicking the Run button












Alternatively, you can run a cell by clicking Shift + Enter. This will not only run the current cell but also create a new empty one underneath it. ➡ This is the easier way!

You can also create a new cell by clicking the “+” button

Let's get familiar with the notebook...

 **Jupyter** PSD L1 Last Checkpoint: 27 minutes ago (autosaved)  [Logout](#)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3












Code

```

In [ ]:

In [1]: print("This is a cell")
        # This is a comment, write whatever you want here, the computer will ignore it :p

        This is a cell
    
```

You can also change the order of the cells by clicking the arrows.

Feeling stuck? Let's take a look at Magic commands

- 🐡 Magic commands are intended to solve common problems in data analysis using Python.
- 🐡 Magic commands act as convenient functions where Python syntax is not the most natural one.
- 🐡 They are useful to embed invalid python syntax in their work flow.
- 🐡 You can check more about magic commands here:
<https://ipython.readthedocs.io/en/stable/interactive/magics.html>





In [3]: `int?`

```
Init signature: int(self, /, *args, **kwargs)
Docstring:
int([x]) -> integer
int(x, base=10) -> integer

Convert a number or string to an integer, or return 0 if no arguments
are given. If x is a number, return x.__int__(). For floating point
numbers, this truncates towards zero.

If x is not a number or if base is given, then x must be a string,
bytes, or bytearray instance representing an integer literal in the
given base. The literal can be preceded by '+' or '-' and be surrounded
```

Class topics

-  Getting familiar
-  Installing Anaconda
-  Getting familiar with Jupyter notebook and creating our first notebook
-  Printing our first statements!

Printing!

Printing:

```
In [6]: print("Python is supper intuitive! in order to print something we only need to type print and write what we desire betw

Python is supper intuitive! in order to print something we only need to type print and write what we desire between s
ingle or double quotes
```

Printing and concatenations:

```
In [7]: print("you can also combine different elements into one print!")

you can also combine different elements into one print!

In [9]: print("like this ",10+10," is the grade you are aiming for! :) ")

like this  20  is the grade you are aiming for! :)
```

Your first print!

```
In [10]: print( "HELLO WORLD!" )
```

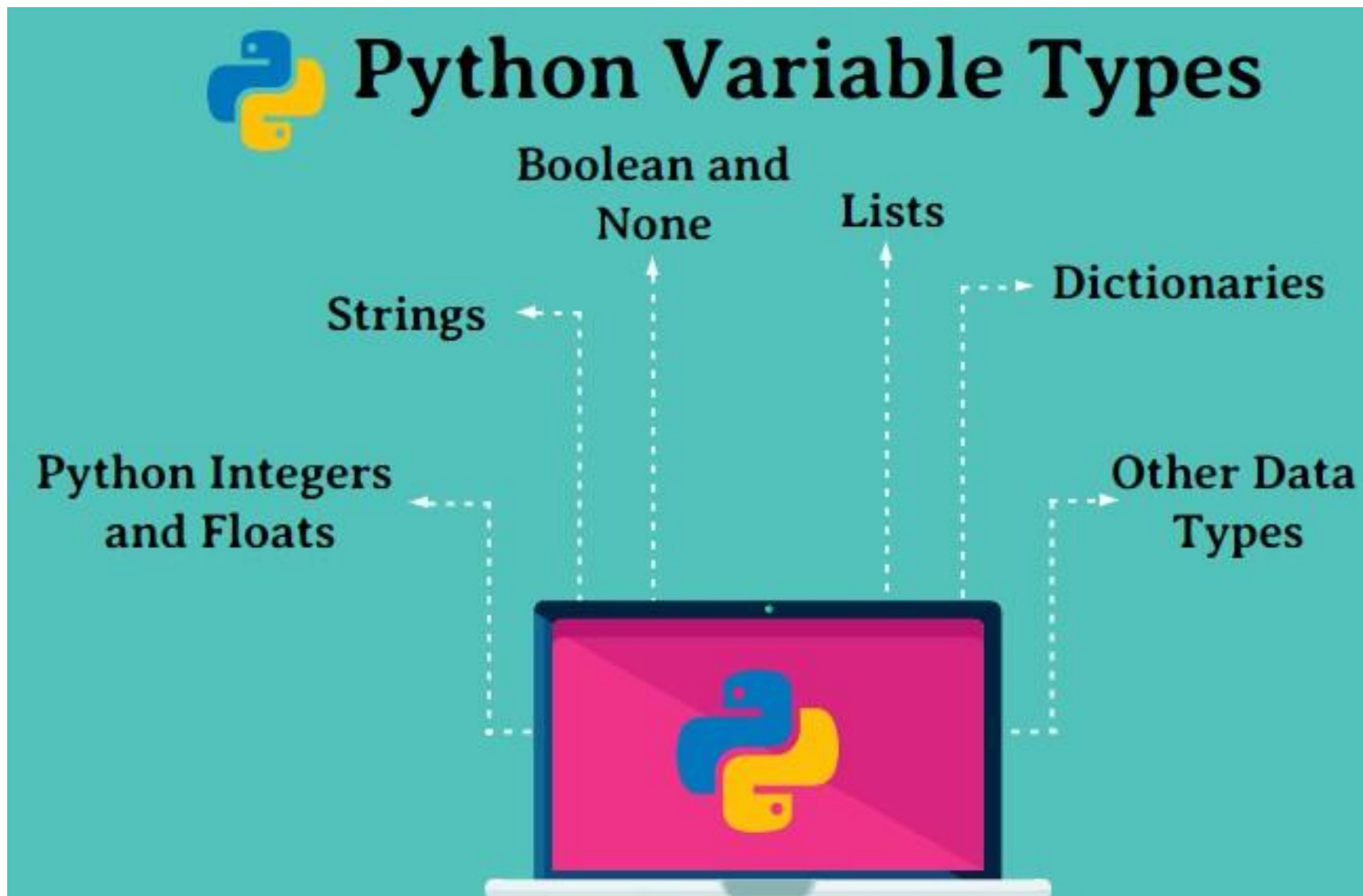
HELLO WORLD!

```
In [11]: print("HELLO WORLD!, can you figure out how to obtain the hello world written underneath? ;)")
```

HELLO WORLD!, can you figure out how to obtain the hello world written underneath? ;)

Hello WORD!

Variable Overview



Python Integers and Floats

12345
67890

12345
67890

```
In [1]: integer = 3
```

```
In [2]: integer
```

```
Out[2]: 3
```

```
In [3]: my_float = 2.4
```

```
In [4]: my_float
```

```
Out[4]: 2.4
```

Strings

```
my_string = "He stole it from us! My Preciousss..."
```

```
my_string
```

```
'He stole it from us! My Preciousss...'
```

```
my_string[0]
```

```
'H'
```

```
my_string[-1]
```

```
'.'
```

```
my_string.upper()
```

```
'HE STOLE IT FROM US! MY PRECIOUSSS...'
```

```
my_string.lower()
```

```
'he stole it from us! my preciousss...'
```

```
len(my_string)
```

```
37
```

Boolean and None

```
10 > 9
```

True

```
15%2 == 0
```

False

```
15%2 == 0 or 10 > 9
```

True

```
15%2 == 0 and 10 > 9
```

False

```
annoying_data = None
```

```
a_boolean_variable = True
```

```
a_boolean_variable
```

True

```
annoying_data + my_float
```

TypeError

Traceback (most recent call last)

```
<ipython-input-21-bfd059f75e15> in <module>
```

```
----> 1 annoying_data + my_float
```

TypeError: unsupported operand type(s) for +: 'NoneType' and 'float'

Boolean and None

```
type(annoying_data)
```

NoneType

```
annoying_data is None
```

True

Lists

List Creation

```
L = [1, 2, 3, None, 'Hello']
```

First element

```
L[0]
```

1

Last element

```
L[-1]
```

'Hello'

Adding a value

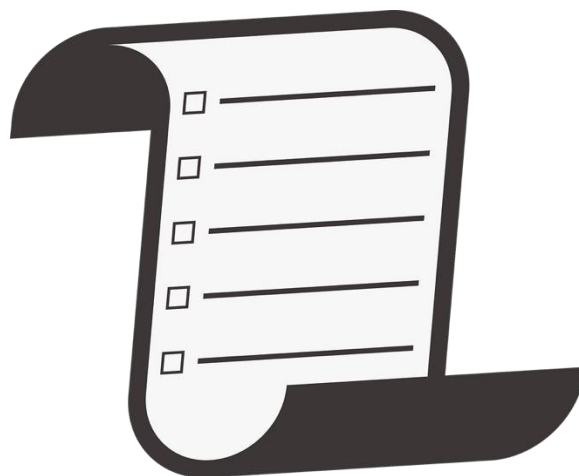
```
L.append("pickle Rick!")
```

Removing specific value

```
L.remove(None)
```

Removing by index

```
L.pop(3)
```



Etc...

Dictionaries

```
months = {  
    'jan' : 1, 'feb' : 2, 'mar' : 3, 'apr' : 4, 'may' : 5, 'jun' : 6, 'jul' : 7,  
    'aug' : 8, 'sep' : 9, 'oct' : 10, 'nov' : 11, 'dec' : 12  
}
```

Printing value of key

```
months['aug']
```

8

Printing the keys of the dictionary

```
months.keys()
```

```
dict_keys(['jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov', 'dec'])
```

Printing the values of the dictionary

```
months.values()
```

```
dict_values([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
```

Remember Mutability

Collection Data Structures

	Mutable	Immutable
Ordered	List	Tuple
Unordered	Dictionary	Sets

Can we change the value of an element in a data structure?

How are elements sorted? By an index or history of addition

Use tab to see what functions we can use with our variable

```
In [ ]: my_set = {"apple", "banana", "cherry"}
```

```
my_set.
```

- add
- clear
- copy
- difference
- difference_update
- discard
- intersection
- intersection_update
- isdisjoint
- issubset

Which will fail?

```
my_list = [1, 2, 3]
my_tuple = (1, 2, 3)
```

```
mylist[0] = 5
my_tuple[0] = 5
```

We can “cast” object to different to types

```
converted_list = list(my_tuple)
converted_list
```

```
[1, 2, 3]
```

See you next week 😊

"You can't just copy-pase pseudocode
into a program and expect it to work"



End