



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO DI
MATEMATICA E INFORMATICA

Social Media Data Analysis 2023/2024

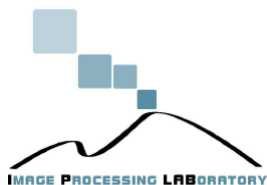
Introduzione ai Laboratori

Francesco Ragusa

francesco.ragusa@unict.it

<https://iplab.dmi.unict.it/ragusa/>

<https://iplab.dmi.unict.it/fpv/>





Python è un linguaggio di programmazione ad alto livello, orientato agli oggetti e adatto, tra i vari usi, al calcolo scientifico.

Vantaggi di Python:

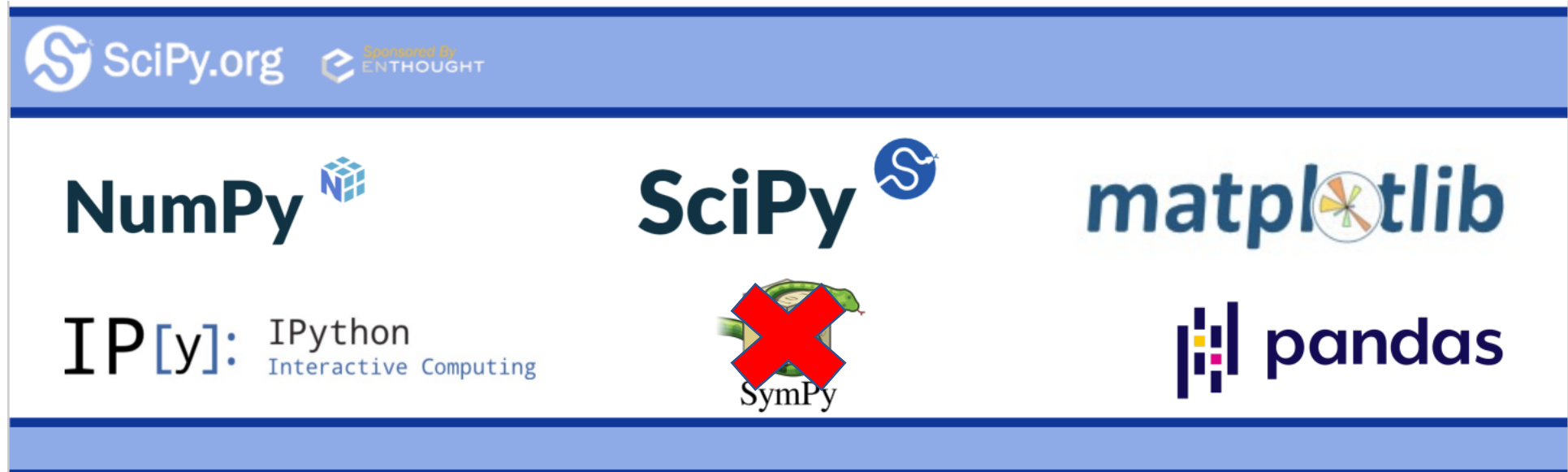
1. Diffuso;
2. Multi-paradigma;
3. Portabile;
4. Facile da usare ed elegante;
5. Dotato di modalità interattiva.

Stack SciPy



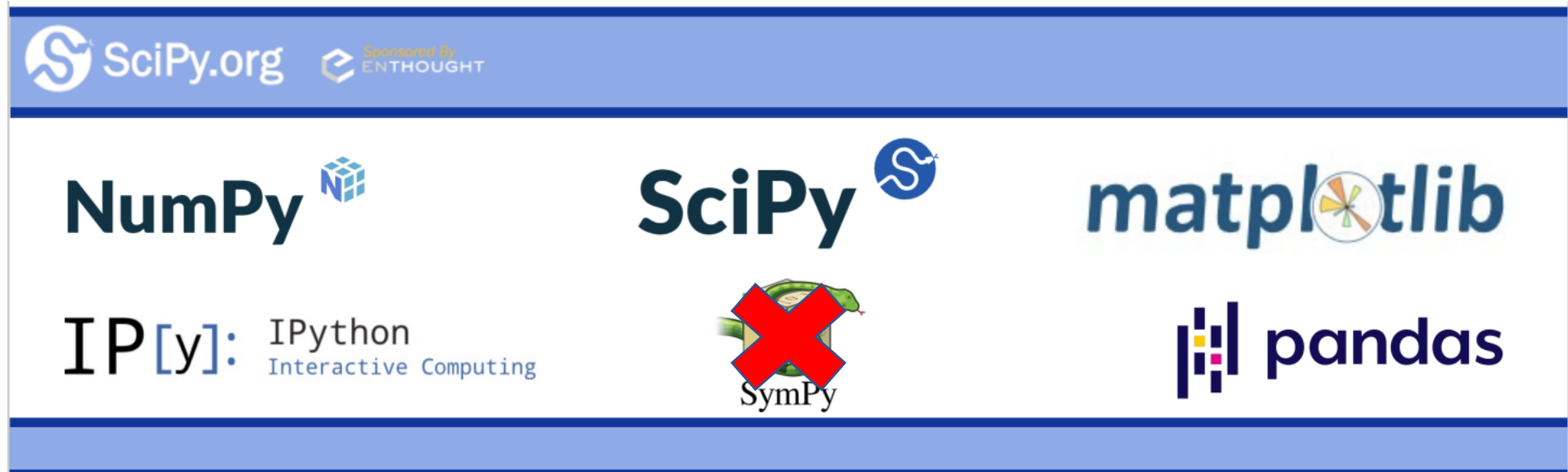
NumPy:	Supporto per vettori e matrici multidimensionali e di grandi dimensioni
SciPy:	Algoritmi fondamentali per il calcolo scientifico
Matplotlib:	Libreria per la creazione di grafici
IPython:	Ambiente interattivo per il calcolo scientifico
Sympy:	Calcolo simbolico
Pandas:	Manipolazione e analisi dei dati

Stack SciPy



NumPy:	Supporto per vettori e matrici multidimensionali e di grandi dimensioni
SciPy:	Algoritmi fondamentali per il calcolo scientifico
Matplotlib:	Libreria per la creazione di grafici
IPython:	Ambiente interattivo per il calcolo scientifico
Sympy:	Calcolo simbolico
Pandas:	Manipolazione e analisi dei dati

Stack SciPy



Statsmodel Modelli statistici
Seaborn Visualizzazioni
Scikit-learn: Machine Learning



Versioni di Python

Esistono due rami ufficiali di Python:

- Ramo 2.x: più vecchio, fino a poco tempo fa un po' più standard;
- Ramo 3.x: più moderno, sempre più diffuso.

Noi utilizzeremo Python 3.9

Distribuzioni di Python

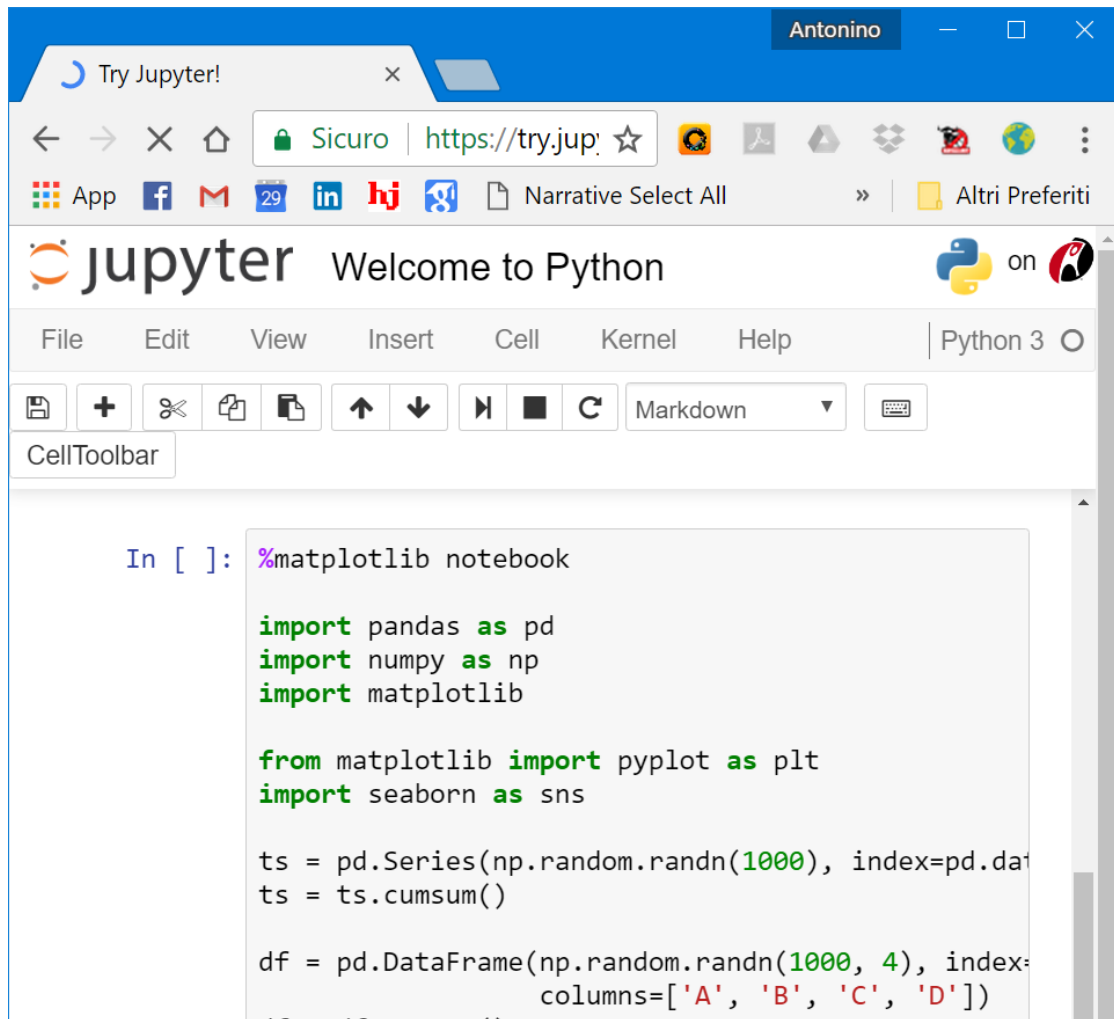
Il modo più semplice per installare tutto l'occorrente è utilizzare una **distribuzione Python**.

Esistono diverse distribuzioni, noi utilizzeremo **Anaconda**.



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

Jupyter Notebooks



Permette di creare (mediante interfaccia web) dei “notebook”, ovvero degli archivi contenenti:

- Testo formattato;
- Il codice da eseguire;
- I risultati (testo, immagini) delle computazioni.

Si tratta di uno strumento molto potente, in quanto permette di generare dei **veri e propri report** delle nostre analisi.

Jupyter Notebooks - Avvio



All'avvio Jupyter Notebooks presenta una interfaccia che permette di navigare tra le cartelle e creare un nuovo Notebook.

Ogni notebook è un file auto-contenuto conservato nel proprio file system.

Esecuzione di Codice

All'esecuzione della cella, viene stampato il risultato dell'operazione contenuta nell'ultima riga.

jupyter Notebook Last Checkpoint: 3 minuti fa (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code Toggle spell checking on a markdown cell

```
In [1]: import numpy as np
        np.random.rand(3,5)
```

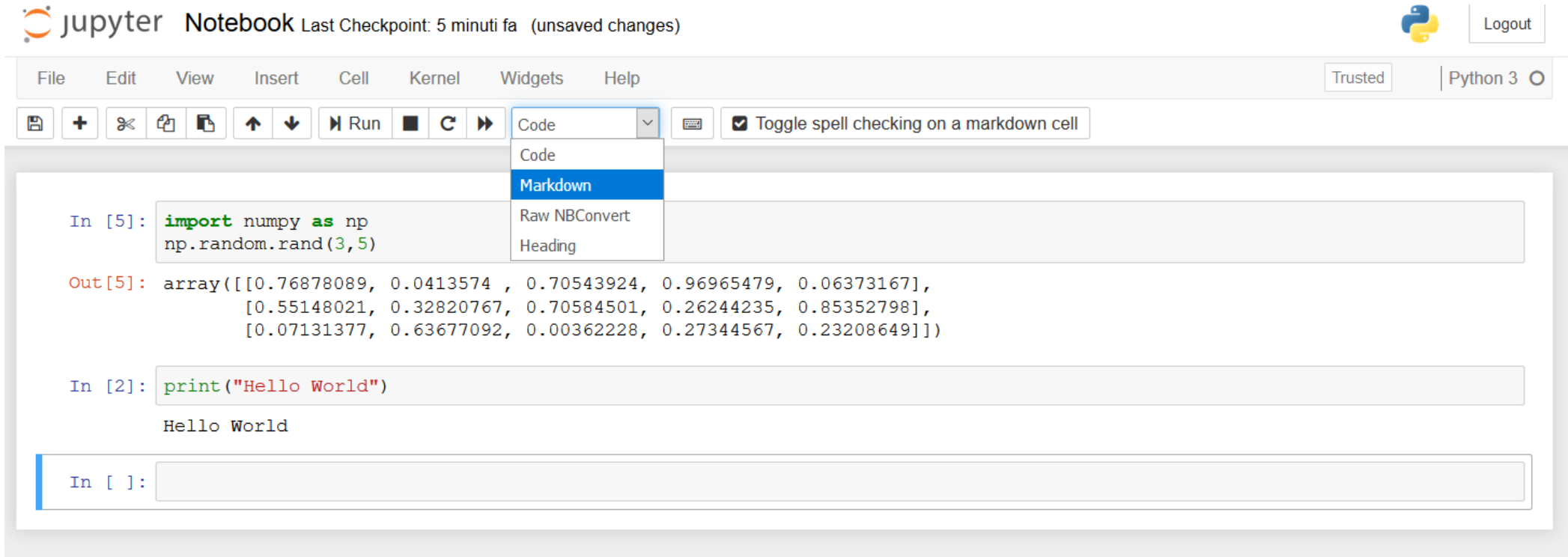
```
Out[1]: array([[0.37516405, 0.05976207, 0.1147749 , 0.4504234 , 0.06885216],
               [0.4233447 , 0.10238852, 0.18110984, 0.42348536, 0.75131921],
               [0.19912864, 0.3838271 , 0.62950433, 0.23751871, 0.32398307]])
```

```
In [2]: print("Hello World")

        Hello World
```

Ogni notebook è composto di celle. Digitando del codice in una cella e poi premendo SHIFT+INVIO è possibile eseguire il codice e passare alla cella precedente;

Tipi di Celle



The screenshot displays the Jupyter Notebook interface. At the top, the header shows the Jupyter logo, the word "Notebook", and the status "Last Checkpoint: 5 minuti fa (unsaved changes)". On the right, there is a Python logo and a "Logout" button. Below the header is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. To the right of the menu bar are buttons for "Trusted" and "Python 3". Below the menu bar is a toolbar with icons for saving, adding, deleting, and running cells, as well as a dropdown menu for cell types. The dropdown menu is currently open, showing options: Code (selected), Markdown, Raw NBConvert, and Heading. The main area of the notebook contains three cells. The first cell is a code cell with the input `In [5]: import numpy as np; np.random.rand(3,5)` and the output `Out[5]: array([[0.76878089, 0.0413574 , 0.70543924, 0.96965479, 0.06373167], [0.55148021, 0.32820767, 0.70584501, 0.26244235, 0.85352798], [0.07131377, 0.63677092, 0.00362228, 0.27344567, 0.23208649]])`. The second cell is a code cell with the input `In [2]: print("Hello World")` and the output `Hello World`. The third cell is an empty code cell with the input `In []:`.

Il tipo di una cella può essere cambiato dal menù a tendina. Esistono due tipi di celle principali: Code (tipo predefinito) e Markdown;

Celle Markdown

Questa è una cella markdown. Posso scrivere in **corsivo** o in `_corsivo_`, in ****grassetto**** o in `__grassetto__`. Posso anche inserire intestazioni:

Intestazione di primo livello

Intestazione di secondo livello

Intestazione di terzo livello

Elenchi puntati:

- * Elemento 1;
- * Elemento 2;
- * Elemento 3.

Elenchi numerati:

1. Elemento 1;
- * Elemento 2;
- * Elemento 3.

Questa è una cella markdown. Posso scrivere in *corsivo* o in `corsivo`, in **grassetto** o in `grassetto`. Posso anche inserire intestazioni:

Intestazione di primo livello

Intestazione di secondo livello

Intestazione di terzo livello

Elenchi puntati:

- Elemento 1;
- Elemento 2;
- Elemento 3.

Elenchi numerati:

1. Elemento 1;
2. Elemento 2;
3. Elemento 3.

Suggerimenti sul codice

```
In [ ]: np.zeros()
```

Docstring:

```
zeros(shape, dtype=float, order='C')
```

Return a new array of given shape and type, filled with zeros.

Per ottenere suggerimenti sul codice (es. parametri in ingresso di una funzione), è possibile premere SHIFT+TAB in prossimità dell'elemento del quale si vuole visualizzare il suggerimento.

Suggerimenti sul codice

```
In [7]: np.zeros?
```

Docstring:

```
zeros(shape, dtype=float, order='C')
```

Return a new array of given shape and type, filled with zeros.

Parameters

shape : int or sequence of ints

Shape of the new array, e.g., ``(2, 3)`` or ``2``.

dtype : data-type, optional

The desired data-type for the array, e.g., ``numpy.int8``. Default is
``numpy.float64``.

order : {'C', 'F'}, optional

Whether to store multidimensional data in C- or Fortran-contiguous
(row- or column-wise) order in memory.

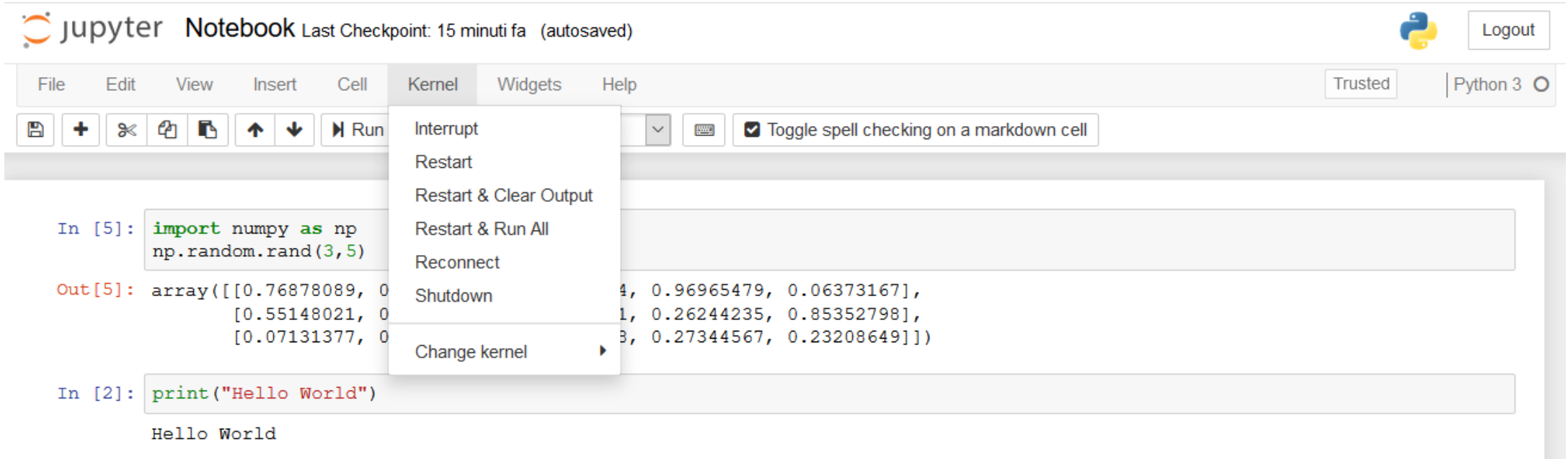
Returns

out : ndarray

Array of zeros with the given shape, dtype, and order.

In alternativa si può digitare il simbolo del quale si vuole visionare la documentazione seguito da un "?"

Validare un Notebook



Quando si conserva un notebook, è importante validarlo. Per farlo bisogna cliccare su Kernel -> Restart & Run All.

Questo cancellerà i risultati di ogni cella e rieseguirà in maniera sequenziale tutte le celle.

Questa procedura è importante per verificare che non ci sono errori nel notebook e che tutti i risultati ottenuti sono ripetibili.

Struttura dei Laboratori

- Elaborato svolto in classe insieme al docente;
- Domande sulle quali ragionare (e alle quali rispondere) in aula;
- Esercizi per casa;
- Se non completati in aula, i laboratori vanno completati a casa.

Laboratori - Domande

Inserite in punti rilevanti del testo per verificare che la comprensione è corretta. Da svolgere durante i laboratori in aula.



Domanda 1.1

Qual è il tipo della seguente variabile?

`x = (3//2*2)**2+(3*0.0)`

Laboratori - Esercizi

Posizionati alla fine di ogni elaborato. Da svolgere a casa. Per chiarimenti sulla risoluzione degli esercizi potete:

- Chiedere delucidazioni nell'incontro successivo;
- Andare a ricevimento.

Esercizi




Esercizio 1

Definire la lista `[1, 8, 2, 6, 15, 21, 76, 22, 0, 111, 23, 12, 24]`, dunque:

- Stampare il primo numero della lista;
- Stampare l'ultimo numero della lista;
- Stampare la somma dei numeri con indici dispari (e.g., 1,3,5,...) nella lista;
- Stampare la lista ordinata in senso inverso;
- Stampare la media dei numeri contenuti nella lista.

Materiale didattico laboratori

Tutti i team



Social Media Data Anal... ...

Pagina iniziale

Blocco appunti per la classe

Il lavoro in classe

Attività


Voti

Reflect

Insights

Canali

Generale

**Generale**

Post

File

+

+ Nuovo





Carica

Sincronizza

Apri in SharePoint

Avvia riunione

Documenti > General > **Materiale del corso**

	Nome	Data/ora modifica	Modificato da	+ Aggiungi colonna
	Lecture 0 - Introduzione al corso di SMDA.p...	6 giorni fa	Francesco Ragusa	
	Lecture 1 - Introduzione ai social media.pdf	6 giorni fa	Francesco Ragusa	
	 Lecture 2 - Getting The Data, Social Media ...	Pochi secondi fa	Francesco Ragusa	

Codice Teams: vo3q1m6

Installazione dell'ambiente di lavoro

Configurazione Ambiente per i laboratori di Social Media Data Analysis

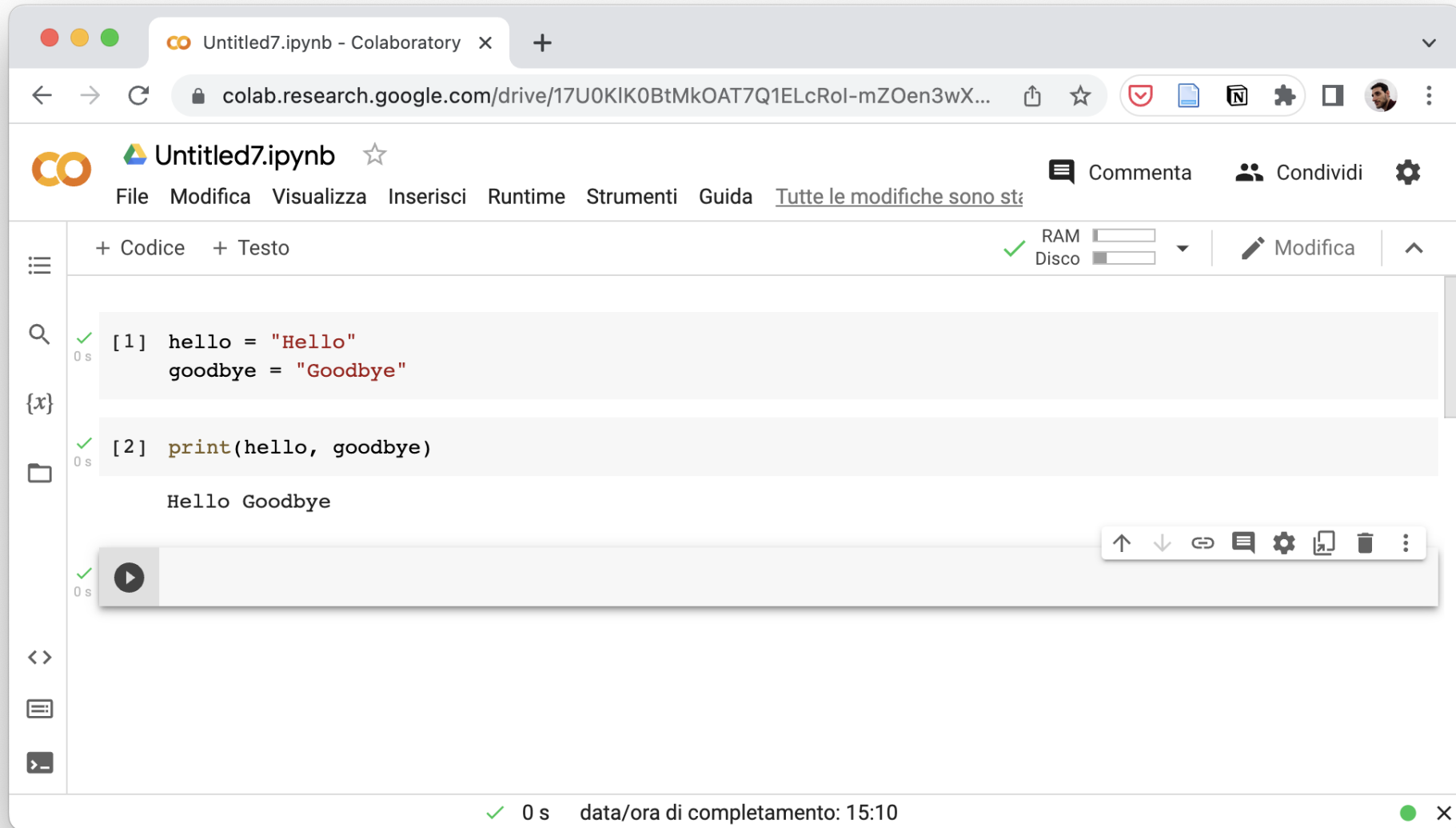
Sommario

Installazione di Python	1
Supporto per la GPU	2
Installazione di PyTorch	3
Programmare in Python	4
Interprete Python	4
Shell interattiva ipython	4
IDE	5
Notebook Jupyter	5
IDE vs Notebook	5
Google Colab	6
Risorse limitate	7
Persistenza dei dati	7

Saltare le parti relative a:

- GPU;
- PyTorch;

Google Colab



<https://colab.research.google.com/>

Primo laboratorio: introduzione a Python



Social Media Data Analysis - A.A. 2023-2024

Introduzione a Python per l'analisi dei Dati

Francesco Ragusa - <https://iplab.dmi.unict.it/ragusa/> - francesco.ragusa@unict.it

Antonino Furnari - <http://antoninofurnari.it> - antonino.furnari@unict.it

Python è un linguaggio di programmazione ad alto livello, interpretato e pensato per la programmazione "general purpose". Python supporta un sistema dei tipi dinamico e diversi paradigmi di programmazione tra cui la programmazione orientata agli oggetti, la programmazione imperativa, la programmazione funzionale e la programmazione procedurale. Il linguaggio è stato ideato nel 1991 da Guido van Rossum e il suo nome è ispirato alla serie TV satirica Monty Python's Flying Circus (https://en.wikipedia.org/wiki/Monty_Python%27s_Flying_Circus).

Benché Python non sia nato come linguaggio di programmazione per il calcolo scientifico e il machine learning, la sua estrema versatilità ha contribuito al nascere di una serie di librerie che rendono la computazione numerica in Python comoda ed efficiente. Buona parte di queste librerie fanno parte di "SciPy" (<https://www.scipy.org/>), un ecosistema di software open-source per il calcolo scientifico. Altre librerie, quali ad esempio PyTorch (<http://pytorch.org/>) mettono a disposizione una serie di strumenti per il calcolo parallelo su GPU orientato al Machine Learning. In queste dispense introdurremo il linguaggio Python 3, vedremo i fondamenti di NumPy, libreria per il calcolo scientifico e Matplotlib, libreria per il plot 2D/3D, e introdurremo gli elementi fondamentali di PyTorch, libreria per il calcolo parallelo su GPU e l'ottimizzazione di algoritmi di machine learning basati sulla discesa del gradiente.

NOTA: Questo laboratorio contiene molte nozioni di base. Il materiale più importante è denotato da un asterisco *.

Referenze importanti da consultare durante il corso, solo le seguenti documentazioni:

- Python 3.9: <https://docs.python.org/3.9/>;
- Numpy: <http://www.numpy.org/>;
- Matplotlib: <https://matplotlib.org/>;
- Pandas: <https://pandas.pydata.org/>.

Resources

- Python: <https://www.python.org/>
- Scipy: <https://www.scipy.org/>
- Project Jupyter: <http://jupyter.org/>
- Tutorial su Jupyter Notebooks:
<https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook>