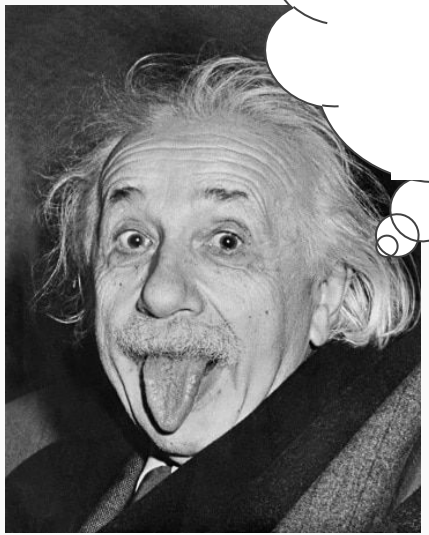
IT Tools for Physicists
2.1 - Introduzione

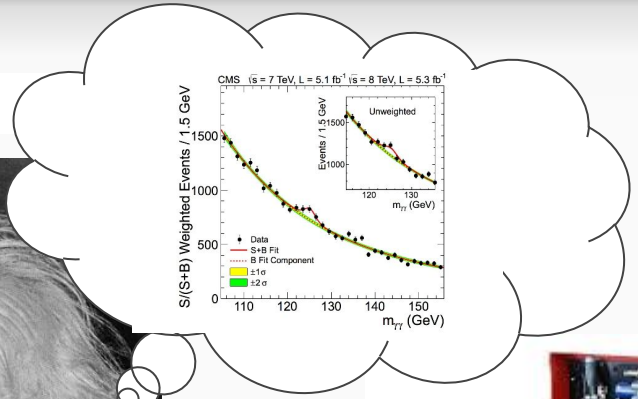
Davide Valsecchi

22/03/2019

Perché siamo qui?

Fisico

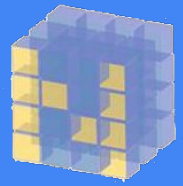
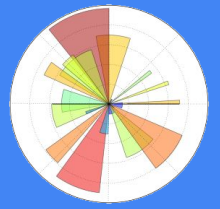




Una serie di incontri

- **IT Tools For Physicists** è una serie di incontri per introdurre gli studenti di fisica a tanti utili strumenti informatici
- Sperimentali, teorici → l'utilizzo del ***calcolatore*** è sempre fondamentale
- Il tempo è sufficiente solamente per un'introduzione → continuate a esplorare!
- In questa primo incontro ci concentriamo sul linguaggio **Python**
- Ma quali sono gli altri argomenti?

Numpy, Matplotlib, Scipy...


- **Numpy**: la libreria alla base del calcolo scientifico in python: gestione di array multidimensionali, molto veloce, infinito numero di funzioni da applicare sui dati matriciali!
- **Matplotlib**: libreria per fare grafici in python, facile da usare
- **Scipy**: framework python contenente moltissime utility per l'analisi dati
- **Pandas**: Libreria per la gestione unificata dei dataset, utilissima ad esempio per i dati di un esperimento (al posto di excel...)

PROSSIMAMENTE: Gianluca Rigoletti, PhD student @ Lion

GNU/Linux

- Odiato o amato alla follia.. prima o poi tutti ci ritroviamo ad utilizzarlo.
- La maggior parte dei contenuti di questi incontri saranno in Python e quindi eseguibili su qualsiasi sistema operativo..
- Ma saper utilizzare l’ambiente Linux è sempre più un requisito per un buon fisico!
- Introduzione a bash e alla gestione dell’ambiente linux

Cmake

- Compilare un programma in C++ può essere più o meno complicato, a seconda della complessità del programma.
- Cmake è un “nuovo” sistema di compilazione standardizzato che semplifica la vita a chi deve compilare e installare un programma
- Impareremo a utilizzare cmake, sia per compilare i software che ci servono, sia per creare il nostro!

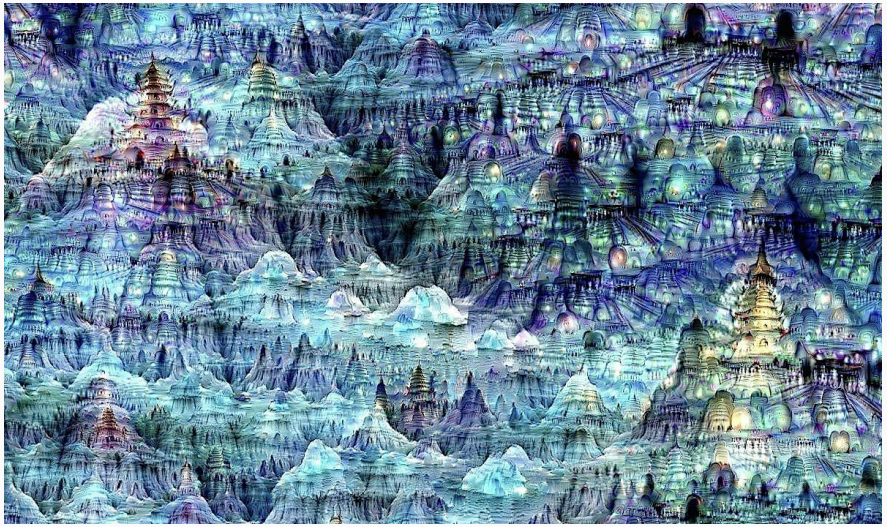
Prossimamente: **Dario Mapelli**

Git & GitHub

- Lavorare su codice scritto da altre persone o condividere il proprio lavoro con altri vi può sembrare un compito lontano… non è così
- **Git** è uno strumento che permette di salvare la storia di un insieme di file sotto forma di tante versioni successive, gestite in modo automatico e distribuito.
- Fondamentale per lavorare con pacchetti software sviluppati in gruppo (dal gruppo di Lab a quello di CMS) ma anche solo per il sorgente Latex della tesi!

Machine learning

- Introduzioni alle Deep Neural Networks
- Regressione e classificazione
- Principali librerie Python: Keras



Evento finale!

- **ITFP 2.0** quest'anno finirà con una giornata in Bicocca dedicata a problemi di ottimizzazione risolvibili tramite gli strumenti mostrati in una singola giornata
- Competizione a gruppi
- Diversi argomenti: ottimizzazione, calcolo numerico, machine learning

Ricchi premi e gadgets!

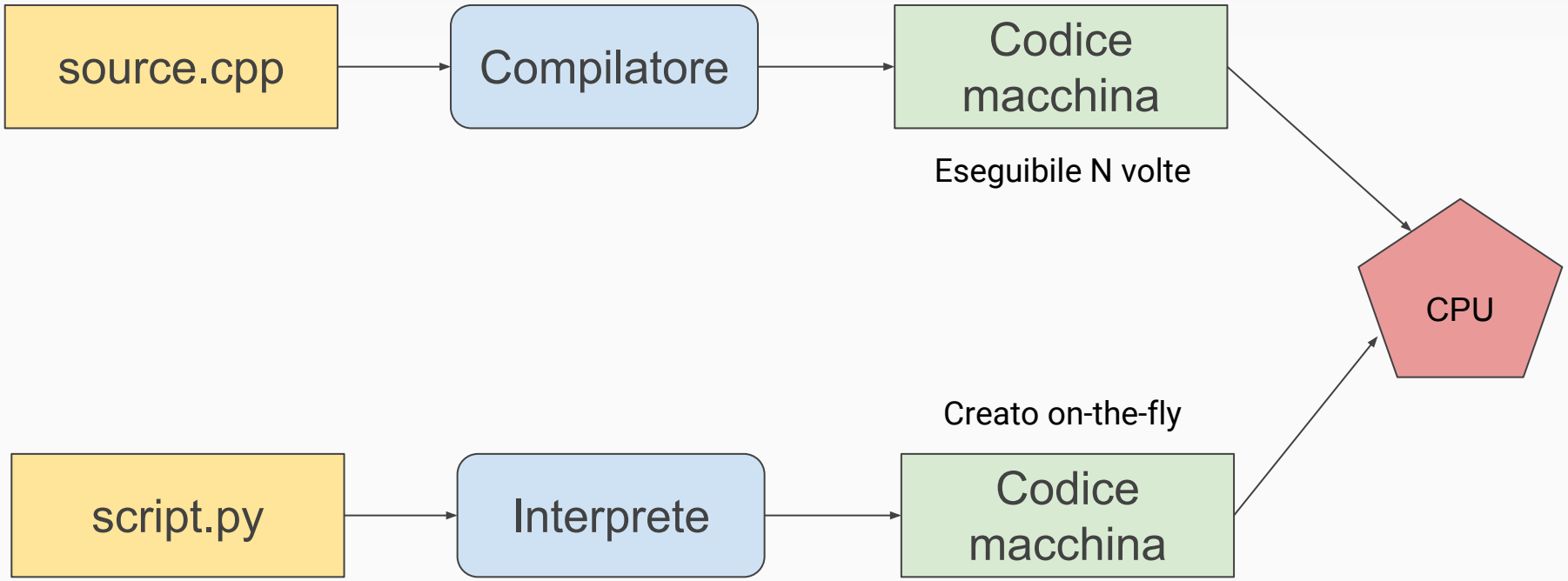
Ma perchè Python?

- E’ un linguaggio molto facile da imparare
- E’ semplice ma completo: c’è una libreria per ogni cosa! Soprattutto per l’ambiente scientifico.
- Prototipazione veloce e individuazione veloce degli errori
- Più “lento” di C++, ma il tempo risparmiato nello sviluppo è un ordine di grandezza più grande! Inoltre le librerie numeriche sono super efficienti.
- E’ ormai il linguaggio standard per interfacciarsi con molti tool scritti in C++ e Fortran, proprio per la sua facilità di utilizzo (ROOT)
- Si è affermato come linguaggio principale nel campo del machine learning

Python: caratteristiche di base

- E’ **interpretato**, non compilato
- E’ un linguaggio basato sugli **oggetti**, proprio come il C++
- Gestione **indiretta** della memoria (niente puntatori)
- **Sintassi chiara** e di facile lettura
- Può essere utilizzato per semplici script come per complessi pacchetti

Compilatore VS Interprete



L’interprete Python

- Interprete ed esegue le istruzioni python inserite riga per riga
- Utile per provare le funzionalità del linguaggio o fare test
- Uno script su file viene invece eseguito con il comando:

```
$ python script.py
```

```
[valsdav@valsdav-t450s ~]$ python
Python 3.7.2 (default, Jan 10 2019, 23:51:51)
[GCC 8.2.1 20181127] on linux
Type "help", "copyright", "credits" or "license" for more
>>>
```

Jupyter Notebook

Run some Python code!

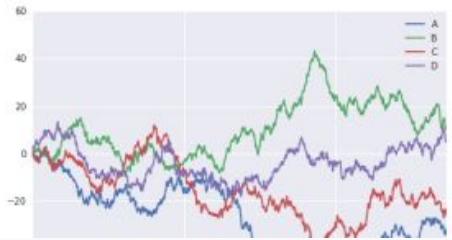
To run the code below:

1. Click on the cell to select it.
2. Press SHIFT+ENTER on your keyboard or press the play button (▶) in the toolbar above.

A full tutorial for using the notebook interface is available here.

In [1]:
```
%matplotlib inline

import pandas as pd
import numpy as np
import matplotlib

from matplotlib import pyplot as plt
import seaborn as sns

ts = pd.Series(np.random.randn(1000), index=pd.date_range('1/1/2000', periods=1000))
ts = ts.cumsum()

df = pd.DataFrame(np.random.randn(1000, 4), index=ts.index,
                  columns=['A', 'B', 'C', 'D'])
df = df.cumsum()
plt.figure(); df.plot(); plt.legend(loc='best')
```

Out[1]:
```
<matplotlib.legend.Legend at 0x7fb27b72fcc0>
<matplotlib.figure.Figure at 0x7fb283672b70>
```



- Il Jupyter notebook è un ambiente interattivo che unisce codice, testo e grafici in un'unica interfaccia.
- Utilissimo per analizzare velocemente i dati e vederne i risultati
- Facile da installare
- Si può condividere la propria analisi con altri o esportare i risultati per una relazione!

Installare Python

- Utilizzeremo la versione Python 3 (>=3.5)
- Il modo più facile di avere un ambiente Python completamente funzionante sul pc e di poter aggiungere tutti i pacchetti necessari è la distribuzione **Anaconda**
- (Potete anche usare il Python già installato sul sistema e il gestore di pacchetti **pip**)