



Corso di Coding in Python & Machine Learning

FabLab Ostiense
Stefano Capezzone

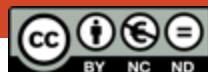


www.chirale.it/materiale-python



SSID: TalentGarden
Pass: Explorers&Innovators

This document is licensed under Creative Common Attribution 4.0 International license



Perché Coding, Python e Machine Learning?

Coding is the 'literacy' of the 21st Century 	Coding is already in the school curriculum for many countries 	CODE POWERS OUR DIGITAL WORLD	
Soon everyone will need to understand the basis of coding 	Coders are the architects of the future 	Online Shopping Amazon	Health & Fitness Tracking Devices
Builds Confidence Fosters creativity & communication	Builds Computational Thinking Builds analytical skills, problem solving and logical thinking 	Home Automation Seniors	Online Banking
Develops Computational Thinking Builds analytical skills, problem solving and logical thinking 	Future Opportunities High demand for computation skills	Robotics	Self Driving Cars
Fun & Interactive Develops Teamwork & Collaboration	Competitive Advantage Understand the digital world around you	EVERYDAY WE USE TECHNOLOGY BUILT WITH CODE 	

Il nostro metodo didattico



Prof. Mitchel Resnick



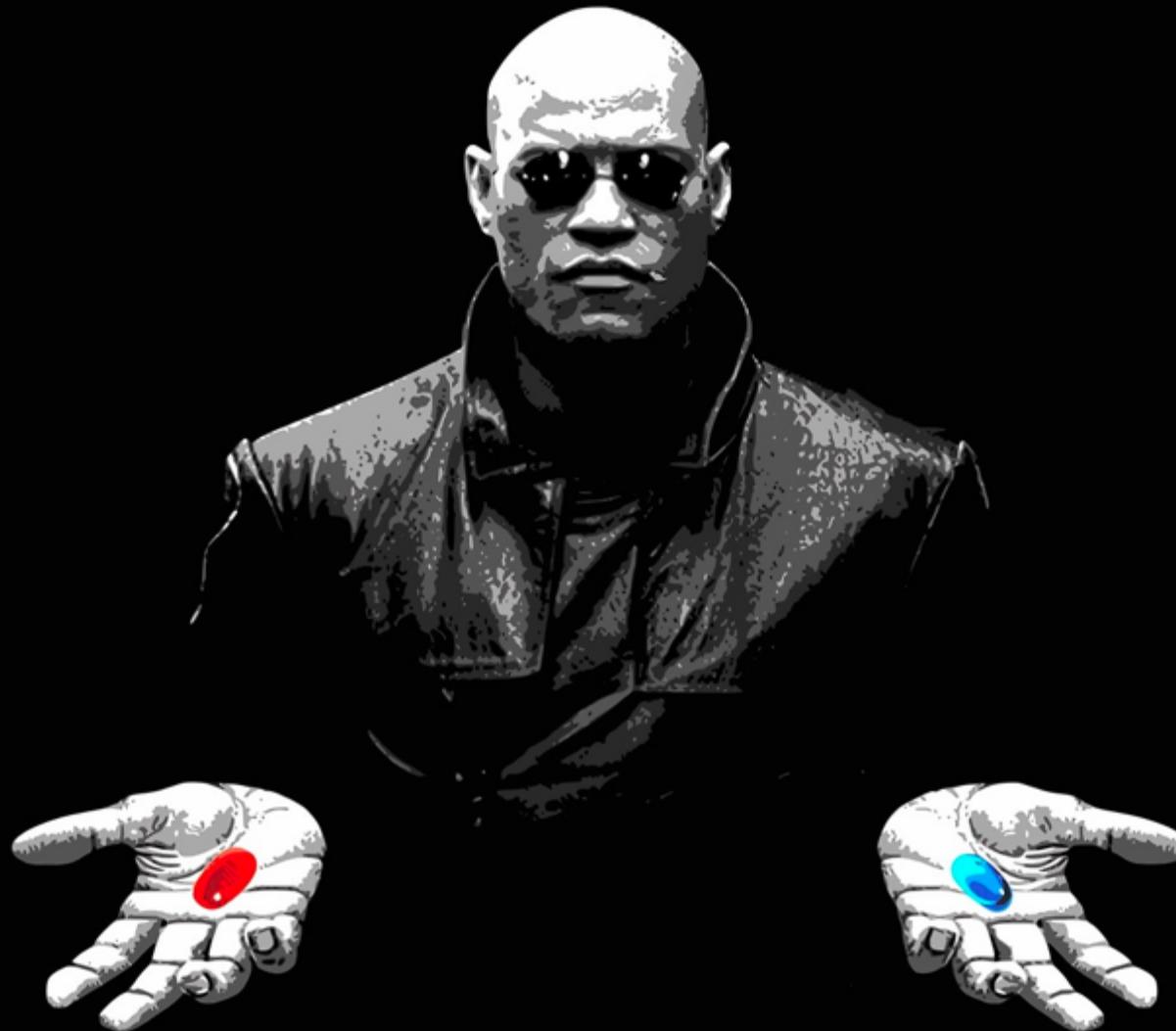
Creative Learning

4P

Projects Peers Play Passion



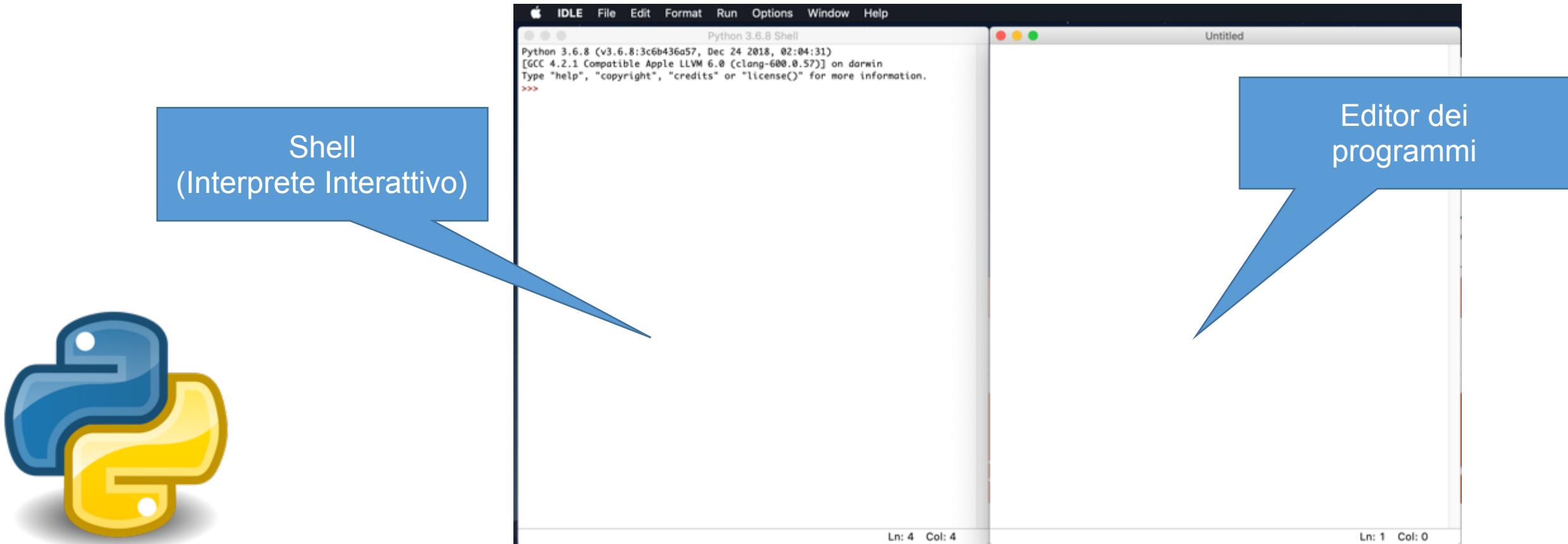
Ci sono due modi per insegnare un Linguaggio di programmazione...



Installiamo il Python



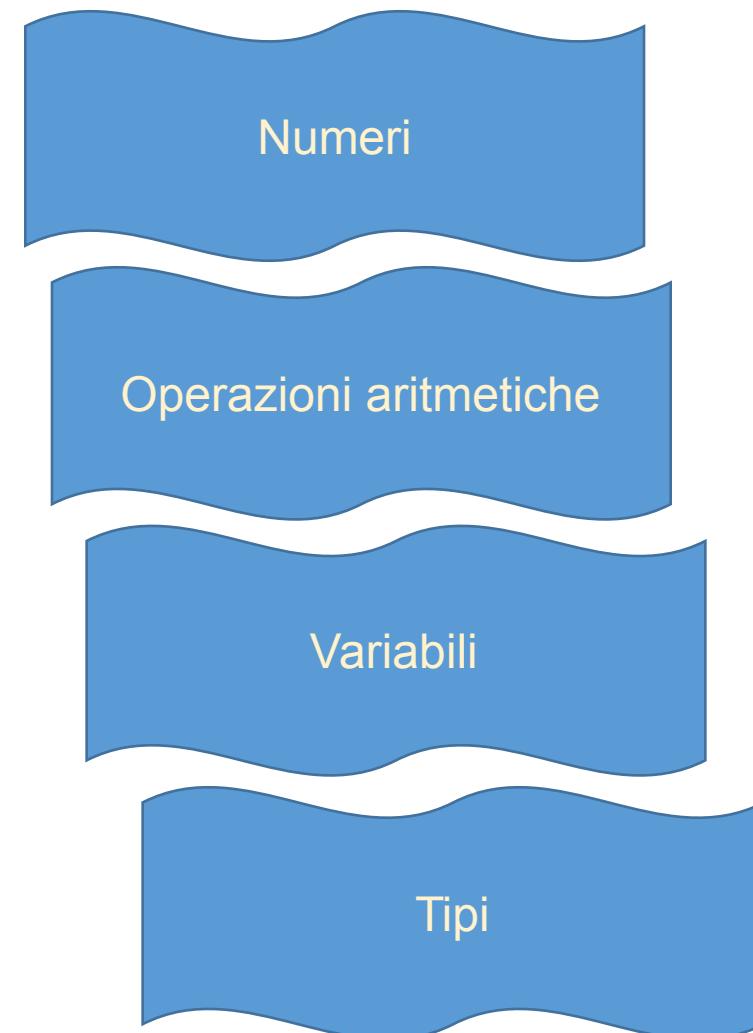
L'ambiente di sviluppo standard del Python: IDLE



Integrated Development & Learning Environment

Cominciamo ad imparare il Python...

Scopriamo da soli i concetti base del *usando il tinkering* Python ~~cazzeggiando~~ con la Shell



Scriviamo il primo programma

Hello



Se il linguaggio fosse JAVA...

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World");  
    }  
}
```



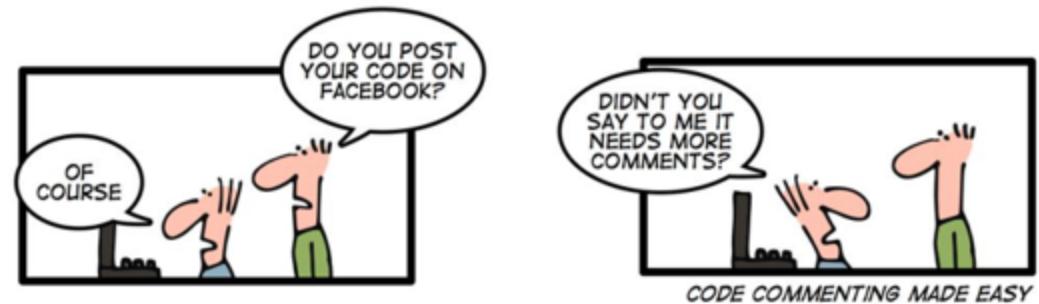
Invece (per fortuna siamo) in Python...

```
print('Hello, World')
```



Inserire commenti e note nel codice...

Commentare il codice è una prassi virtuosa aiuta gli altri membri del gruppo e se stessi a comprendere meglio la logica del programma!



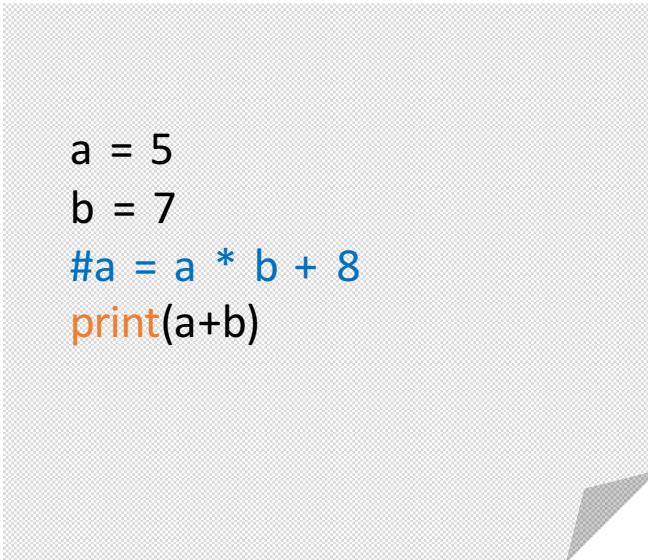
```
a = 5  
b = 7  
# questa riga sarà ignorata dall'interprete  
Python  
print(a+b)
```

```
a = 5  
b = 7  
''' questo intero blocco  
di testo sarà ignorato  
dall'interprete Python'''  
print(a+b)
```

Tramite i commenti è possibile «disattivare» parti del programma

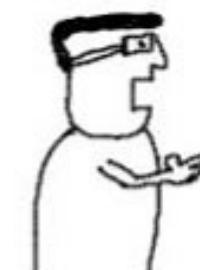
```
a = 5  
b = 7  
#a = a * b + 8  
print(a+b)
```

A volte è necessario eliminare temporaneamente istruzioni del programma per verificare situazioni o comprendere meglio le cause di un errore.



La tecnica del *comment-out* è anche utile per non dover cancellare parti di codice che potrebbero servire in futuro. Ricordatevi però di togliere le parti che non serviranno mai!

THE PROGRAM
I CODED HAS
LOTS OF BUGS
HOW DO I REMOVE
THEM?
WHY DON'T
YOU PUT ENTIRE
CODE IN
COMMENTS
!!



Un po' di nozioni per completare le basi

```
# Casting dei tipi  
a = 5  
print(type(a))  
a = float(a)  
print(type(a))  
a = str(a)  
print(type(a))
```

Operatori numerici

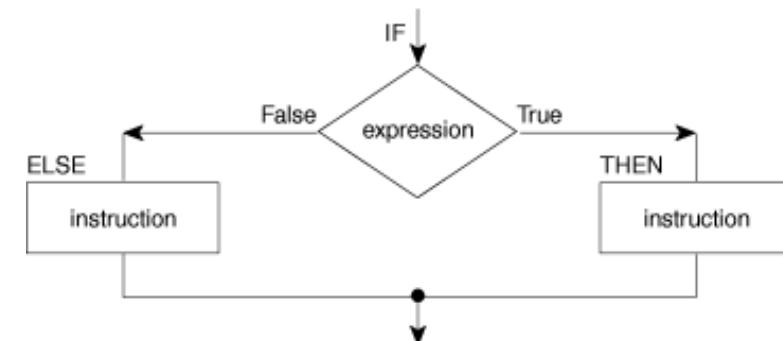
```
a = 5 + 4  
b = 5 - 4  
c = 5 * 4  
d = 10 / 3  
e = 10 // 3  
f = 10 % 3  
g = 2 ** 3
```

a += 3 # equivale ad a = a+3

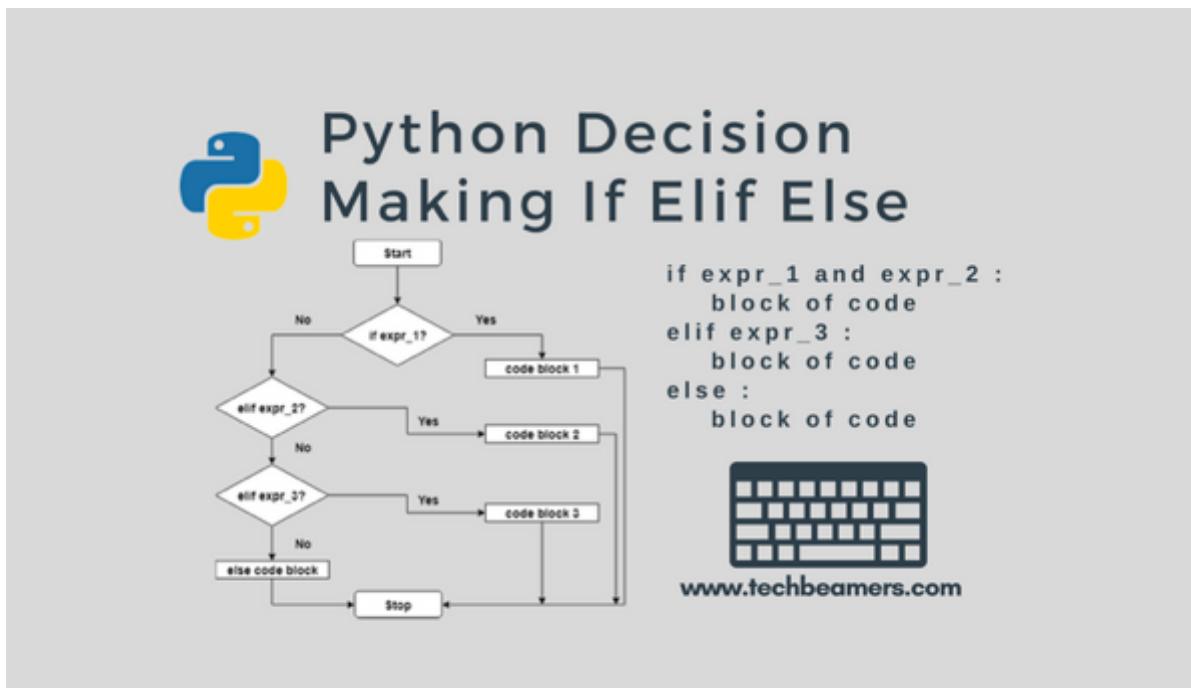


Il controllo del flusso del programma

```
# Esempio di diramazione condizionale
a = eval(input('Numero ?'))
if a > 10:
    print('Il numero è maggiore di dieci!')
else:
    print('Il numero è minore o uguale a 10')
```



L'istruzione completa IF ... ELIF ... ELSE



Esempio di if-elif-else

```
a = eval(input('Numero ?'))  
if a > 10:  
    print('Il numero è maggiore di dieci!')  
elif a == 10:  
    print('Il numero è uguale a 10')  
else:  
    print('Il numero è minore di 10')
```

Esercizio!

Scrivere un programma che:

1. Chiede il tuo nome
2. Ti saluta in modo personalizzato
3. Chiede la tua età
4. Ti risponde in base alla tua età:
 - Sei minorenne
 - Sei maggiorenne

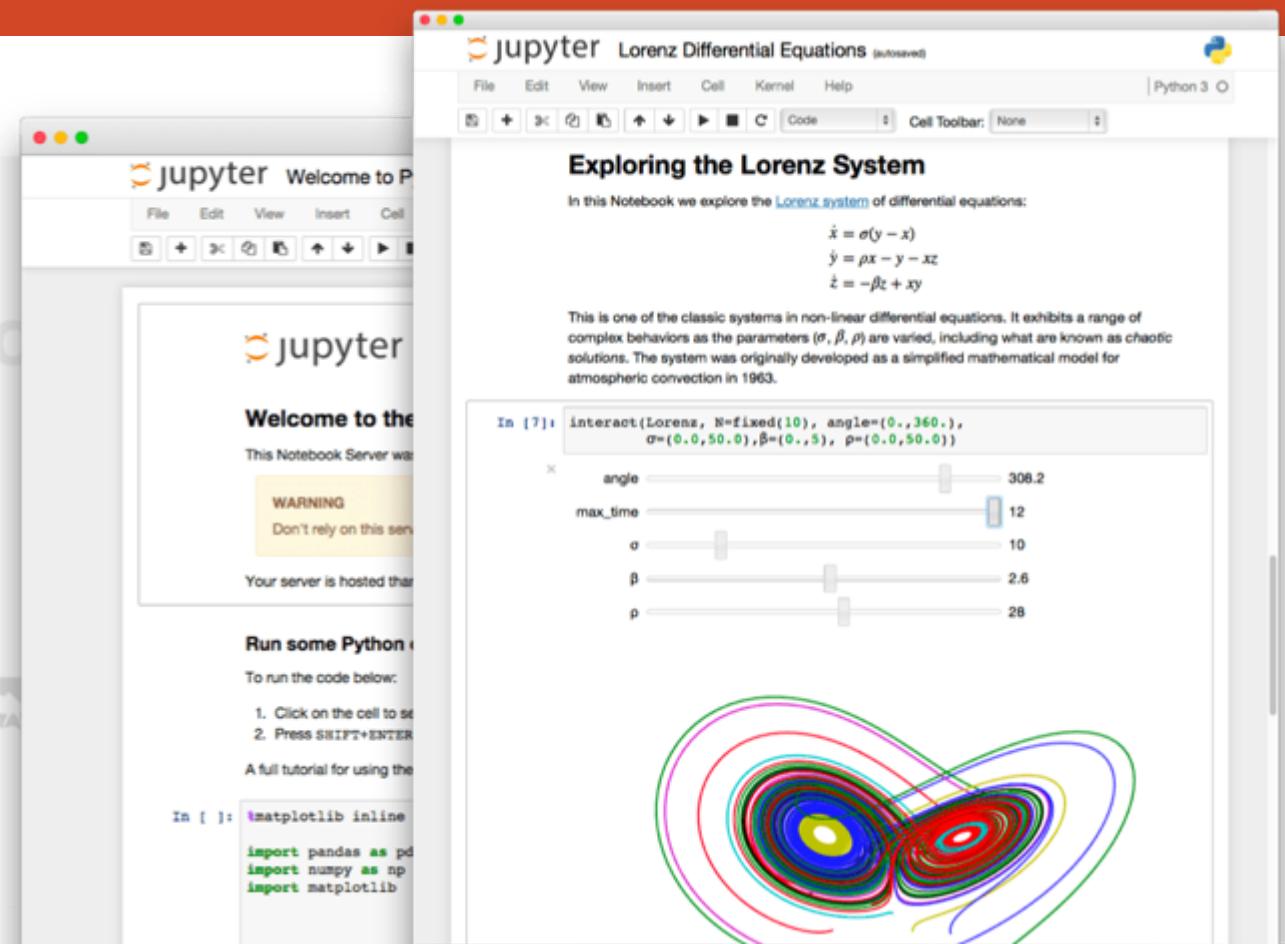
Versione basic

Scrivere un programma che:

1. Chiede il tuo nome
2. Ti saluta in modo personalizzato
3. Chiede la tua età
4. Ti risponde in base alla tua età:
 - Sei minorenne
 - Sei maggiorenne ma non puoi votare per il Senato
 - Sei maggiorenne e puoi anche votare per il Senato

Versione expert

Lo strumento Jupyter Notebook e iPython



Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

Google Colaboratory Research

Google Colab è un servizio gratuito che permette di creare e condividere *notebook Jupyter/iPython*, già configurati per utilizzare la piattaforma **TensorFlow** per lo sviluppo di applicazioni di Intelligenza Artificiale **senza che sia necessario installare e configurare nulla sul proprio PC**.

Il servizio è offerto in *cloud* a chiunque possieda un account Google-Gmail, permette di creare e salvare i notebook sul proprio spazio Google Drive e si integra con **GitHub**.



Un esempio di notebook Colab



<https://tinyurl.com/introduzione-colab>