

Introduzione alla Robotica e Python

VARS
CONTROL

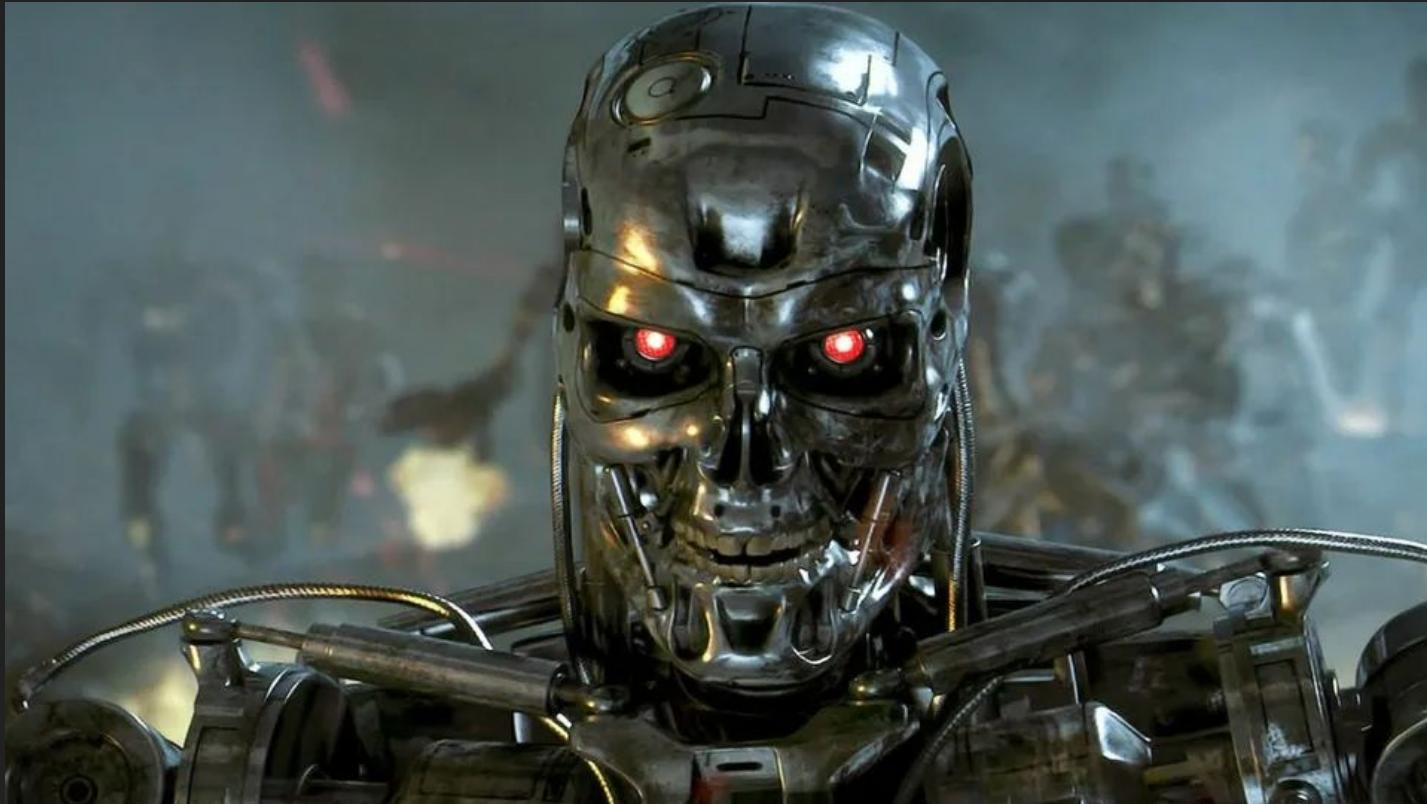


UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

Pensate ad un **Robot**. Cosa immaginate?



Pensate ad un **Robot**. Cosa immaginate?



Definizione di robot

Un robot è una qualsiasi macchina (più o meno *antropomorfa*) in grado di svolgere, più o meno indipendentemente, un lavoro al posto dell'uomo. Il nome proviene dalla parola ceca *robo*ta che significa *lavoro pesante*.

Definizione di robot

Un robot è una qualsiasi macchina (più o meno antropomorfa) in grado di svolgere, più o meno indipendentemente, un lavoro al posto dell'uomo. Il nome proviene dalla parola ceca *robo*ta che significa *lavoro pesante*.

Nel linguaggio comune, un **robot** è un'apparecchiatura artificiale che compie determinate azioni **in base ai comandi che gli vengono dati** e alle sue funzioni, sia in base ad una supervisione diretta dell'uomo, sia **autonomamente** basandosi su linee guida generali, magari usando processi di intelligenza artificiale.

Macchina automatica vs Robot

Macchina Automatica:

Robot:

Macchina automatica vs Robot

Macchina Automatica: Fa sempre la stessa cosa, nello stesso modo, senza "pensare". È come un programma con un solo comando. Pensa a un tostapane: mette il pane, schiacci il pulsante, esce il pane tostato (più o meno). Non si adatta a niente.

Robot:

Macchina automatica vs Robot

Macchina Automatica: Fa sempre la stessa cosa, nello stesso modo, senza "pensare". È come un programma con un solo comando. Pensa a un tostapane: mette il pane, schiacci il pulsante, esce il pane tostato (più o meno). Non si adatta a niente.

Robot: È una macchina programmabile che può fare diverse cose, e in alcuni casi può anche "reagire" a quello che succede intorno. È come un computer con tanti programmi diversi. Pensa a un robot aspirapolvere: pulisce la stanza, ma se trova un ostacolo lo evita, e se finisce la batteria torna alla base a ricaricarsi.

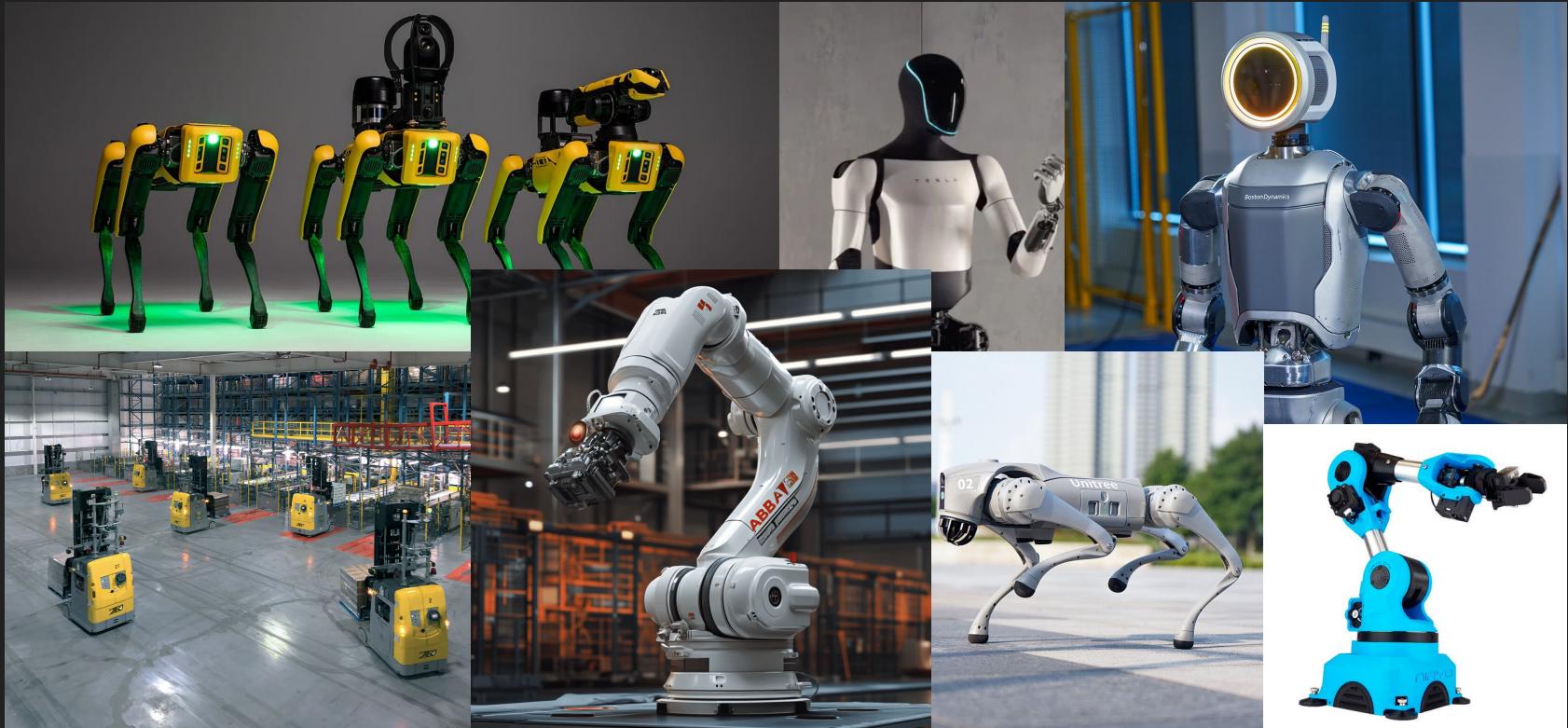
Macchina automatica vs Robot



Pensate ad un Robot. Cosa immaginate?



Pensate ad un Robot. Cosa immaginate?



Pensate ad un **Robot**. Cosa immaginate?

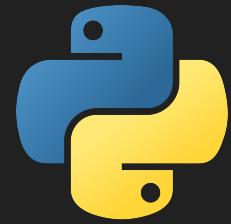




Divisione in team per la challenge

[Random Team Generator](#)





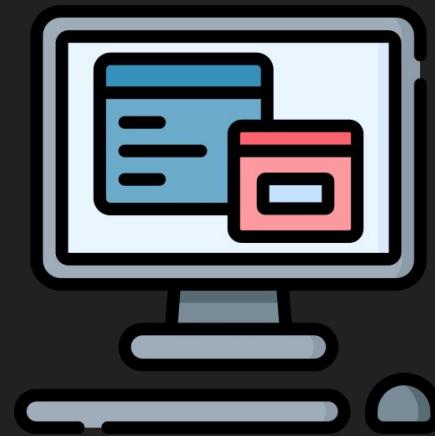
Introduzione a Python

PCTO - Lezione 1

<https://tinyurl.com/pctopython>

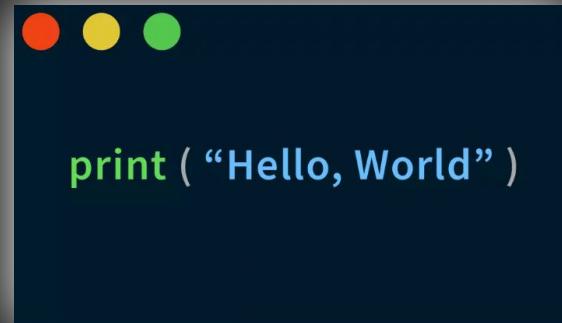
Linguaggio di programmazione interpretato, di alto livello e general-purpose.

```
33     self.logdupes = True
34     self.debug = debug
35     self.logger = logging.getLogger(__name__)
36     if path:
37         self.file = open(os.path.join(path, 'fingerprint.log'), 'w')
38         self.file.seek(0)
39         self.fingerprints.update(self._read_file())
40
41     @classmethod
42     def from_settings(cls, settings):
43         debug = settings.getbool('superuser_debug')
44         return cls(job_dir(settings), debug)
45
46     def request_seen(self, request):
47         fp = self.request_fingerprint(request)
48         if fp in self.fingerprints:
49             return True
50         self.fingerprints.add(fp)
51         if self.file:
52             self.file.write(fp + os.linesep)
53
54     def request_fingerprint(self, request):
55         return self._fingerprint(request)
```



Perchè Python?

- Linguaggio semplice
- Moltissime librerie a disposizione 💪
- Principalmente utilizzato in ambito scientifico, data science, ... e **robotica!**



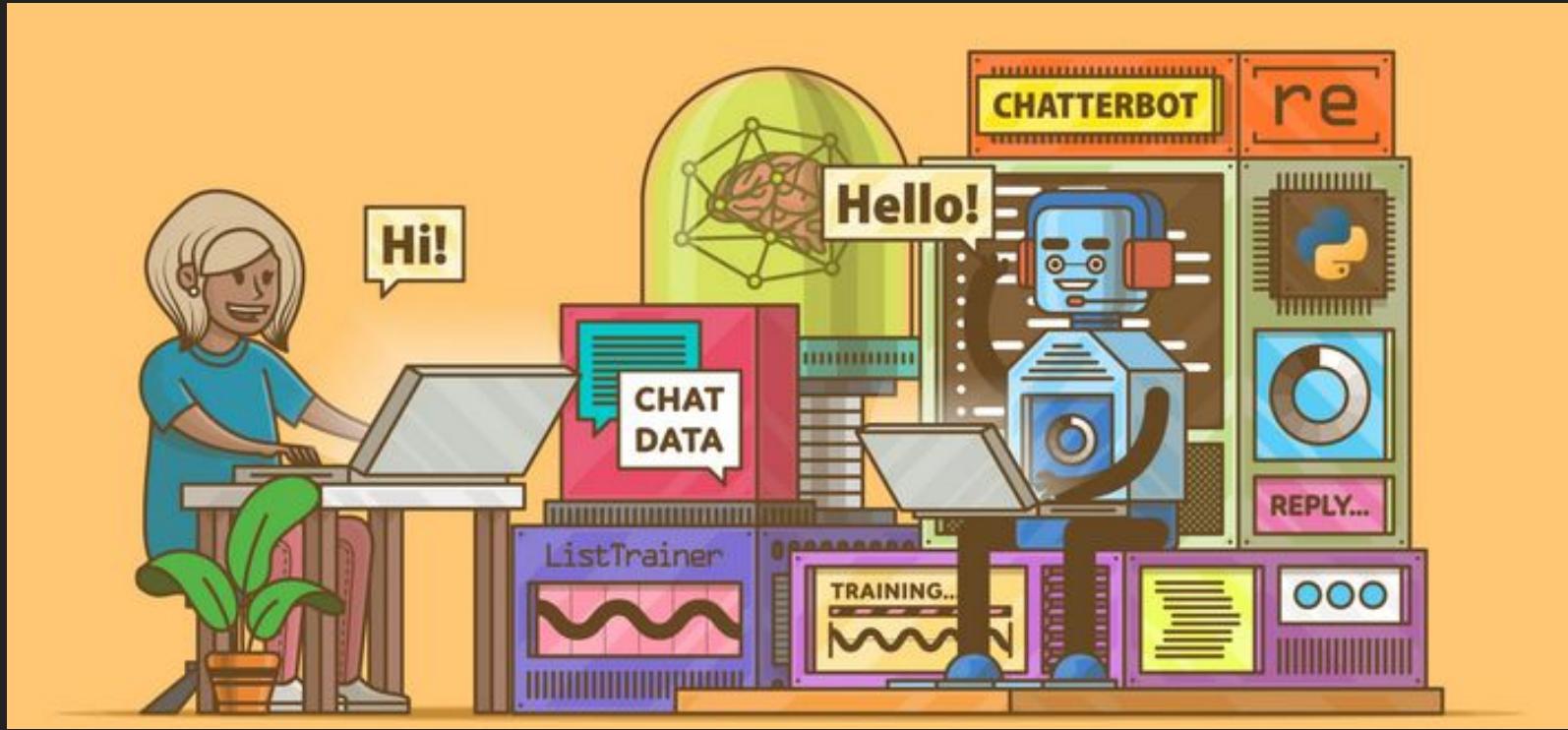
NETFLIX



Utilizzato nella nostra
vita quotidiana



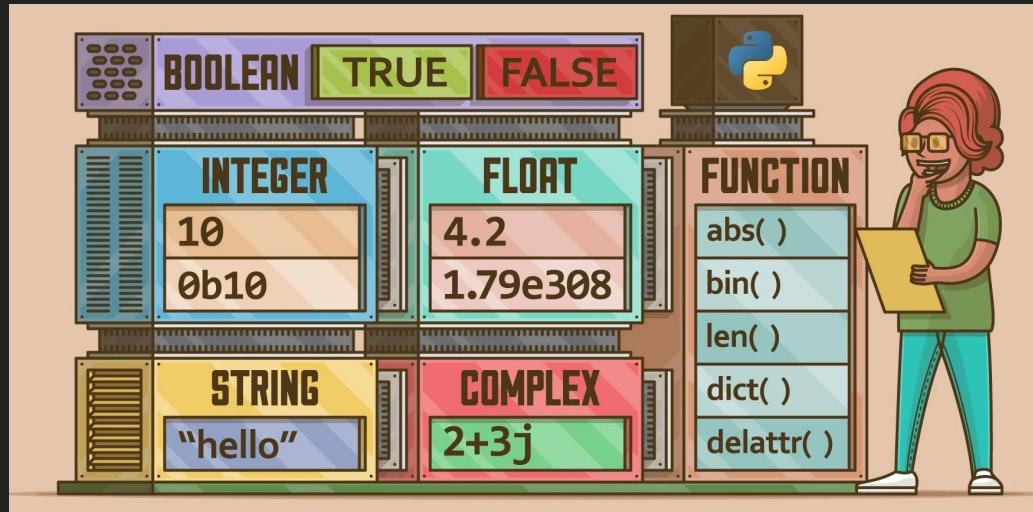
È fondamentale nell'Intelligenza Artificiale



Variabili e Tipi di Dato

Le variabili su Python **NON** si dichiarano, sono allocate automaticamente.

Nei linguaggi di programmazione si inizia a contare sempre da 0.



Variabili e Tipi di Dato

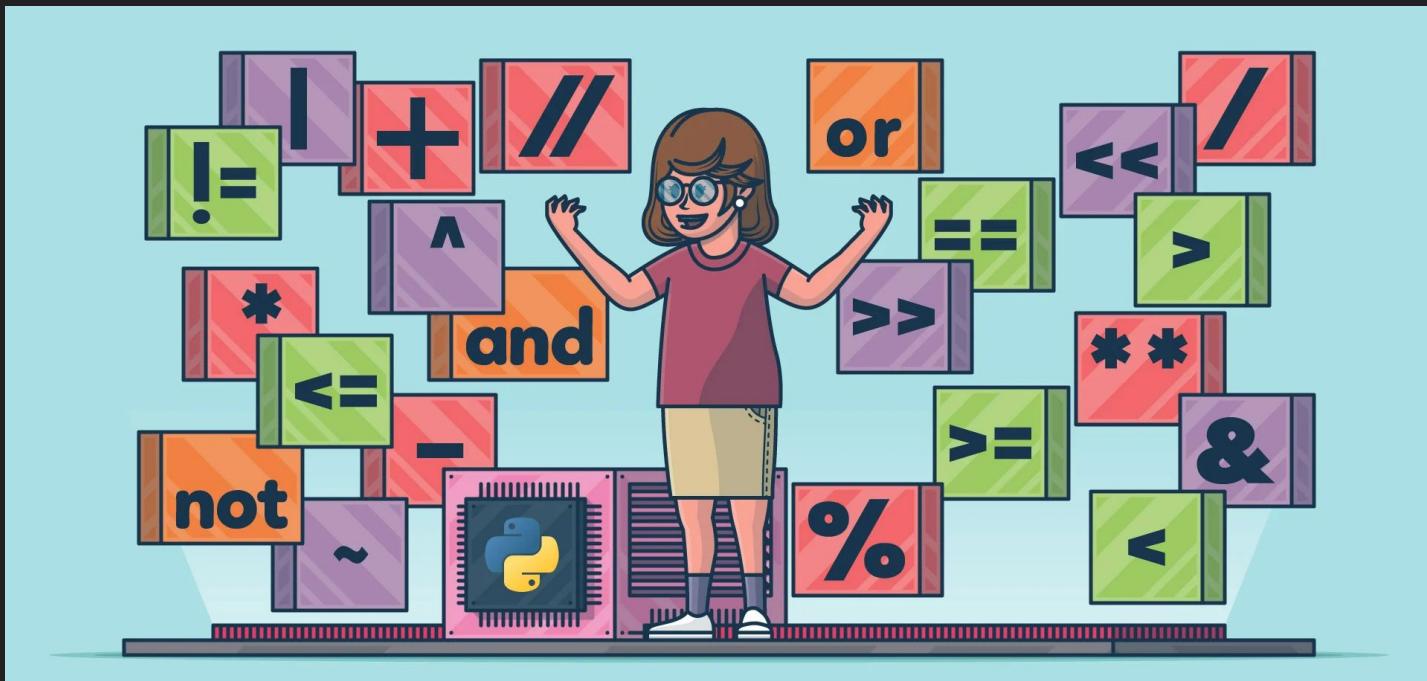
Python



```
nome = "Mario"    # Stringa
età = 25          # Intero
altezza = 1.75    # Float
maggiorenne = True # Booleano

print(nome)
print(età)
print(altezza)
print(maggiorenne)
```

Operatori aritmetici



Operatori aritmetici

Python



```
x = 10
y = 3
somma = x + y
differenza = x - y
quoziente = x / y
resto = x % y

print(somma)
print(differenza)
print(quoziente)
print(resto)

print(x > y) # True
print(x == y) # False
```

Strutture di controllo

Eseguire blocchi di codice in base a condizioni.



Strutture di controllo

Python



```
età = 16
if età >= 18:
    print("Sei maggiorenne")
else:
    print("Sei minorenne")

voto = 7
if voto >= 8:
    print("Ottimo")
elif voto >= 6:
    print("Sufficiente")
else:
    print("Insufficiente")
```

Iteratori su sequenze: il For Loop

Per iterare su sequenze con una lunghezza prestabilita.



Iteratori su sequenze: il **For Loop**

Il **For Loop** ha una lunghezza **DEFINITA**

Python



```
for i in range(5): # Stampa da 0 a 4
    print(i)
```

Iteratori condizionali: il While Loop

Per eseguire un blocco di codice **finché una condizione è vera**.



Iteratori condizionali: il **While Loop**

Il **For Loop** ha una lunghezza **INDEFINITA!**

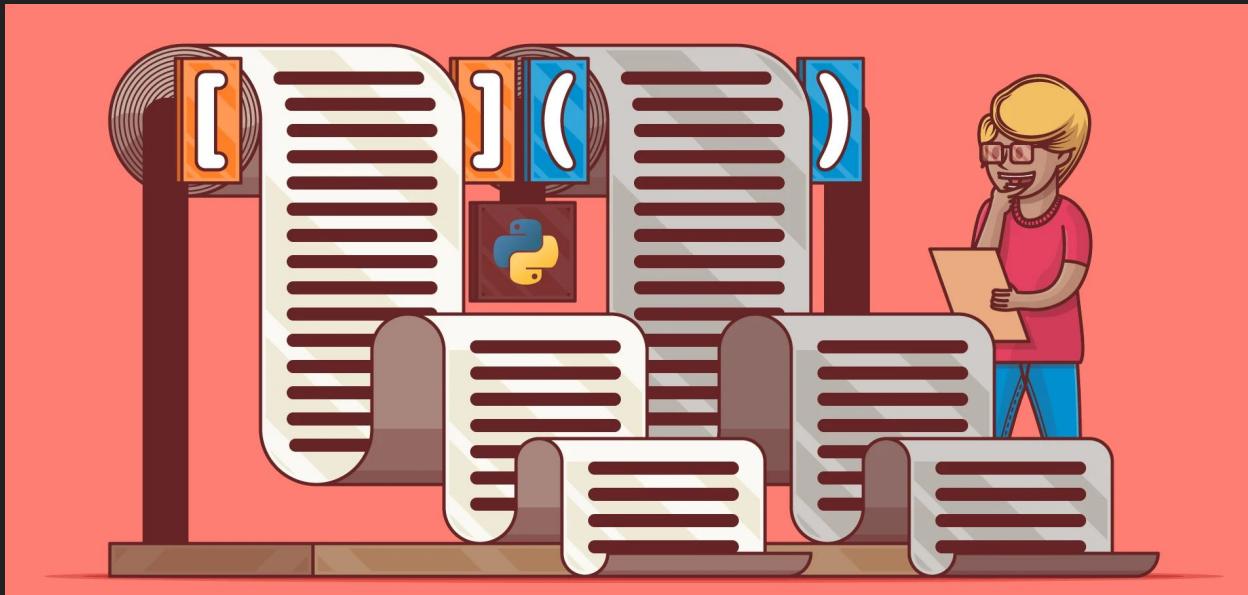
Con il **While Loop** si
possono fare danni.
**Da evitare quando
possibile**



```
i = 0
while i < 5:
    print(i)
    i += 1
```

Contenitori

In Python esistono le **liste** [*modificabili*] e le **tuple** (*non modificabili*).



Contenitori

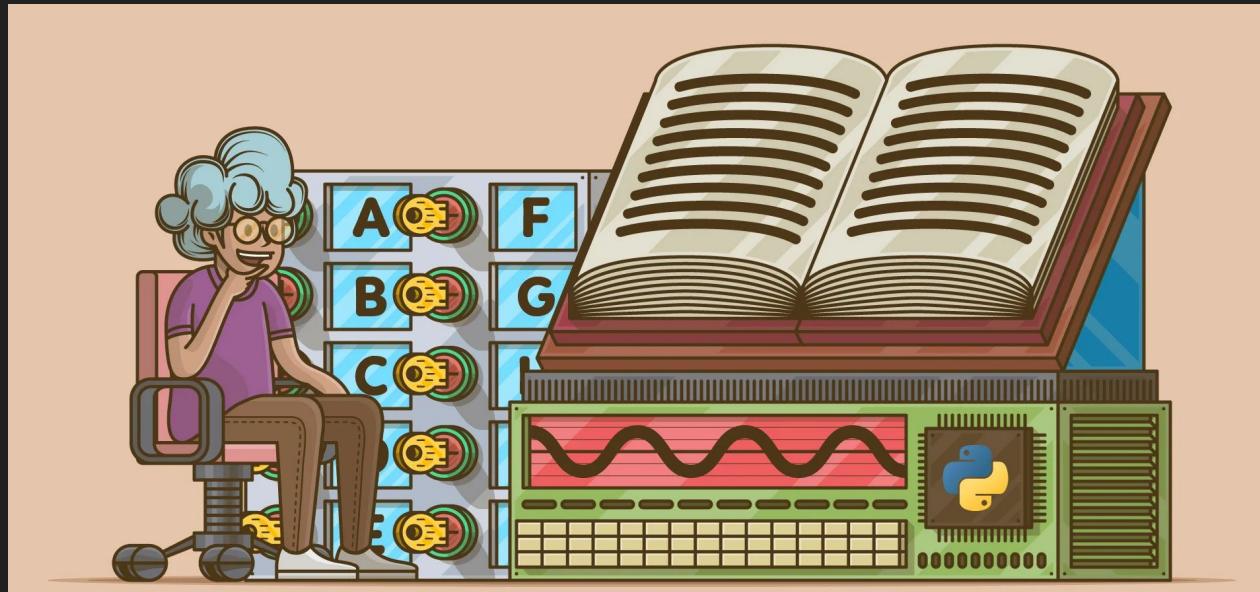
Python



```
numeri = [1, 2, 3, 4, 5]
print(numeri[0]) # Stampa 1
numeri.append(6)
print(numeri) #Stampa [1, 2, 3, 4, 5, 6]
```

Dizionari

I dizionari servono per memorizzare dei dati in coppie **{KEY:VALUE}**



Dizionari

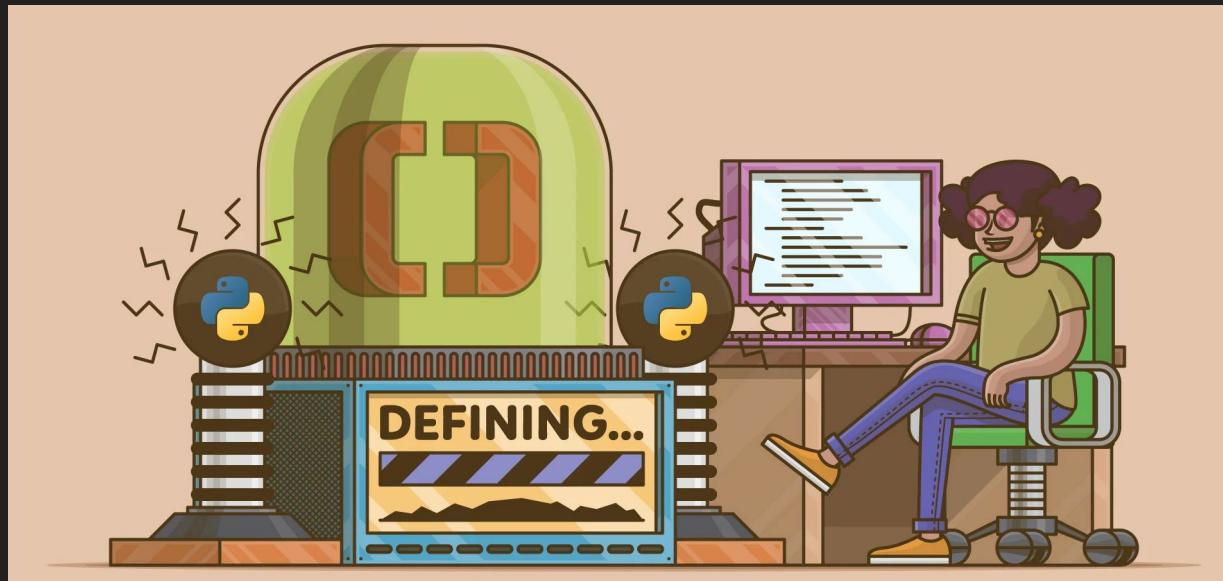
Python



```
# Creazione di un dizionario che rappresenta informazioni su una persona
persona = {
    "nome": "Mario",
    "cognome": "Rossi",
    "età": 30,
    "città": "Roma"
}
```

Funzioni

Le funzioni ci permettono di **assegnare** una **porzione di codice** e delle variabili chiamate parametri ad un **nome**.



Funzioni

```
def saluta(nome):
    """Questa funzione saluta la persona specificata.""" #Docstring, utile per la documentazione
    print("Ciao, " + nome + "!")

saluta("Alice") # Output: Ciao, Alice!

def somma(x, y):
    risultato = x + y
    return risultato

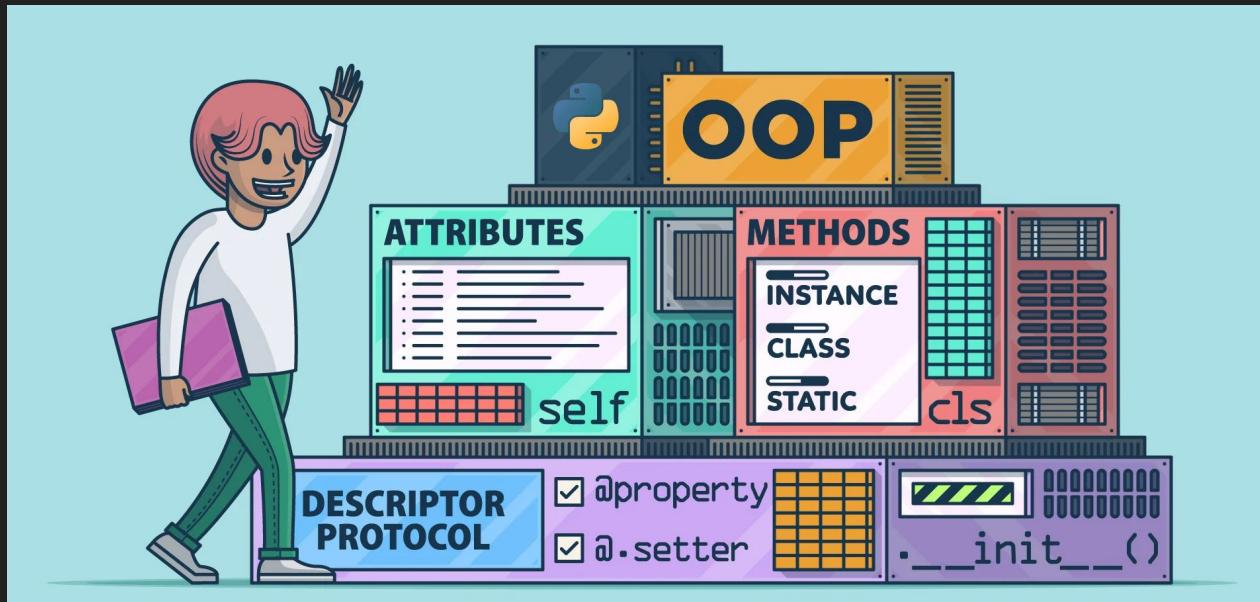
z = somma(5, 3)
print(z) # Output: 8

def controlla_maggiorenne(eta):
    if eta >= 18:
        return True
    else:
        return False

if controlla_maggiorenne(19):
    print("Maggiorenne")
```

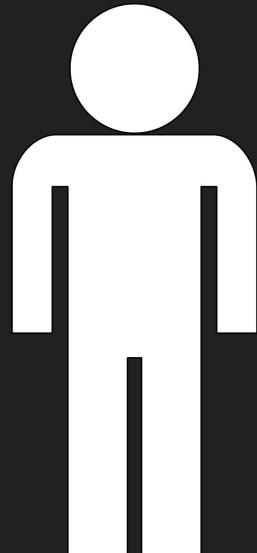
Classi

Le classi sono importantissime e distinguono un linguaggio di programmazione normale da uno **orientato agli oggetti (OOP)**.

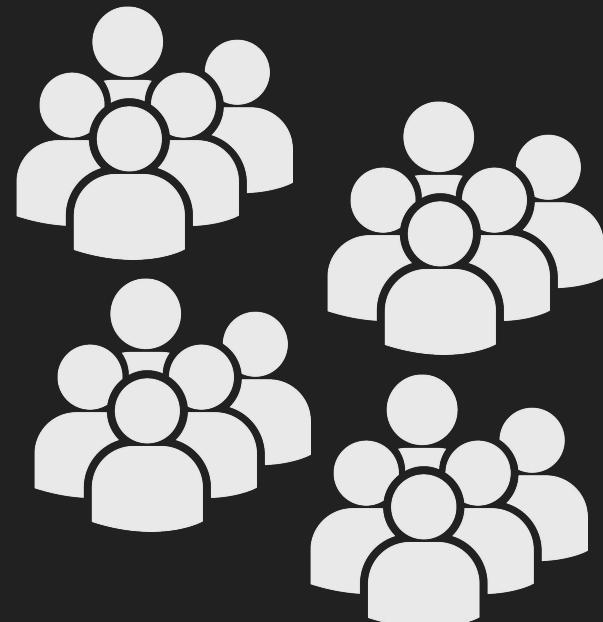


Classi

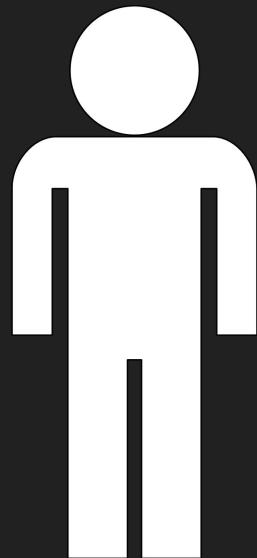
Le classi sono un modo per organizzare il codice e rappresentare concetti complessi.



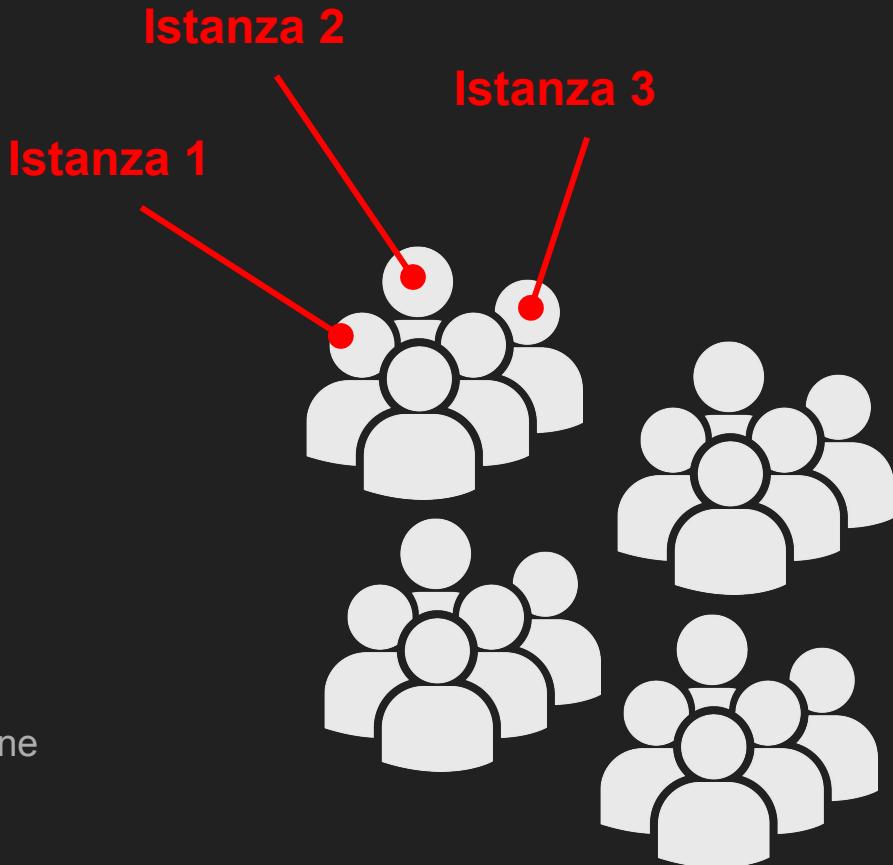
1. Nome e Cognome
2. Età
3. Anno di nascita
4. Sesso
5. Peso
6. Altezza
7. Istruzione
8. Salario
9. Stile di alimentazione
10. ...



Classi



1. Nome e Cognome
2. Età
3. Anno di nascita
4. Sesso
5. Peso
6. Altezza
7. Istruzione
8. Salario
9. Stile di alimentazione
10. ...



Classi

```
class Persona:  
    def __init__(self, nome, eta): #Costruttore, viene chiamato quando si crea un oggetto  
        self.nome = nome  
        self.eta = eta  
  
    def saluta(self):  
        print("Ciao, mi chiamo " + self.nome + " e ho " + str(self.eta) + " anni.")  
  
persona1 = Persona("Bob", 30) #Creazione di un oggetto di tipo Persona  
persona1.saluta() # Output: Ciao, mi chiamo Bob e ho 30 anni.  
  
persona2 = Persona("Eve", 25)  
persona2.saluta() # Output: Ciao, mi chiamo Eve e ho 25 anni.
```

Classi

Si riferisce all'istanza della classe in uso

```
class Persona:  
    def __init__(self, nome, eta): #Costruttore, viene chiamato quando si crea un oggetto  
        self.nome = nome  
        self.eta = eta  
  
    def saluta(self):  
        print("Ciao, mi chiamo " + self.nome + " e ho " + str(self.eta) + " anni.")  
  
persona1 = Persona("Bob", 30) #Creazione di un oggetto di tipo Persona  
persona1.saluta() # Output: Ciao, mi chiamo Bob e ho 30 anni.  
  
persona2 = Persona("Eve", 25)  
persona2.saluta() # Output: Ciao, mi chiamo Eve e ho 25 anni.
```

Kahoot!



FLY THROUGH DENSE FOREST

Feedback sulla lezione?

