

Introduzione a Python

Francesco Pacenza

Prof. Giovambattista Ianni

Orario di ricevimento: Mercoledì dalle 15:30 alle 17:30 (previa comunicazione via email)

Email: `pacenza@mat.unical.it`

Corso di **Sistemi Operativi e Reti** - A.A. 2018/19

Dipartimento di Matematica e Informatica



October 3, 2018

INTRODUZIONE

Introduzione

Python:

- Python è un linguaggio di script pseudocompilato → Linguaggio Interpretato¹

L' **interprete** è un normale programma che va installato sulla propria macchina, e si occuperà di interpretare il codice sorgente e di eseguirlo

- Non c'è una fase di *compilazione* - *linking* che trasforma il sorgente in eseguibile
- Si ha a disposizione solo il sorgente che viene direttamente eseguito dall'interprete

¹ Il codice sorgente viene tradotto da un interprete in linguaggio macchina al momento dell'esecuzione senza necessità di compilatore

Vantaggi & Svantaggi di Python

Vantaggi

- **Portabilità** → Lo stesso programma potrà girare su diverse piattaforme purché vi sia installato l'interprete
- **Object Oriented** → Supporta le classi, l'ereditarietà e si caratterizza per il binding dinamico
- **Short Learning Curve** → Chiunque nell'arco di un paio di giornate può imparare ad usarlo e a scrivere le sue prime applicazioni

Vantaggi & Svantaggi di Python

Vantaggi

- **Portabilità** → Lo stesso programma potrà girare su diverse piattaforme purché vi sia installato l'interprete
- **Object Oriented** → Supporta le classi, l'ereditarietà e si caratterizza per il binding dinamico
- **Short Learning Curve** → Chiunque nell'arco di un paio di giornate può imparare ad usarlo e a scrivere le sue prime applicazioni

Svantaggi

- Uso dell'operatore `self`
- Indentazione:

```
def main():  
    one_to_ten = range(1,11)  
    for x in one_to_ten:  
        if x%2==0:  
            print("Il numero %d e' divisibile per 2" % (x))  
            # print("Il numero {x} e' divisibile per 2".format(x=x))  
            # print("Il numero " + str(x) + " e' divisibile per 2")  
  
if __name__ == '__main__':  
    main()
```

Vantaggi & Svantaggi di Python

Vantaggi

- **Portabilità** → Lo stesso programma potrà girare su diverse piattaforme purché vi sia installato l'interprete
- **Object Oriented** → Supporta le classi, l'ereditarietà e si caratterizza per il binding dinamico
- **Short Learning Curve** → Chiunque nell'arco di un paio di giornate può imparare ad usarlo e a scrivere le sue prime applicazioni

Svantaggi

- Uso dell'operatore `self`
- Indentazione:

```
def main():
    one_to_ten = range(1,11)
    for x in one_to_ten:
        if x%2==0:
            print("Il numero %d e' divisibile per 2" % (x))
            # print("Il numero {x} e' divisibile per 2".format(x=x))
            # print("Il numero " + str(x) + " e' divisibile per 2")

if __name__ == '__main__':
    main()
```



Costrutti in Python I

I principali costrutti Python sono i seguenti `if`, `for` e `while`

Costrutto if

```
if x < 0:
    print ("Negativo")
elif x == 0:
    print ("Zero")
else:
    print ("Positivo")
```

Costrutto for

```
for i in range(1,100):
    print (i)

words = [ 'cat', 'window', 'defenestrare' ]
for w in words:
    print(w, len(w))
```

Costrutti in Python II

Costrutto while

```
while True:
    print ("Loop Infinito")

x = 0
while x!= -1:
    print ("Il valore di x e' %d" % (x))
    x = int(input("Inserisci x: "))
```


Primi programmi in Python I

Hello World

```
def main():  
    print ("Hello world !")  
  
if __name__ == '__main__':  
    main()
```

Definizione di una classe e dei suoi metodi

```
class MyClass:  
    def __init__(self, x=0):  
        print("Costruttore con parametro x")  
        self.x = x  
  
    def printX(self):  
        print ("x = %d" %(self.x))
```

Primi programmi in Python II

Uso della classe e accesso ai campi

```
from MyClass import MyClass

def main():
    myClassInstance = MyClass(3)
    myClassInstance.printX()
    myClassInstance.x = 4
    print (myClassInstance.x)

    mySecondClassInstance = MyClass()

if __name__ == '__main__':
    main()
```

Esercizi I

- Si crei una classe **persona** che contenga tutti i dati relativi alla stessa (Codice fiscale, nome, cognome, data di nascita, . . .)
- Si crei una classe **esame** che abbia i seguenti campi pubblici: *Nome insegnamento, codice insegnamento, votoP*
- Si crei una classe **studente** che eredita da persona (*Campi richiesti: Matricola, lista esami superati*); la classe studente dovrà offrire i seguenti metodi pubblici
 - CalcolaMedia → Stampa in output la media degli esami dello studente
 - ControllaEsame → Dato un esame in input , controlla se lo stesso è presente nella lista degli esami superati dello studente
- Si crei un **main** di prova per testare il programma