

Corso Introduttivo di Python

Lezione 1

Jacopo, Luca

GOLEM

08/05/2019





Figura 1: Guido van Rossum



Figura 2: I Monty Python

python™



“Interprete” Python;

- Spesso preinstallato su Linux e MacOS
- La versione per Windows si scarica da <https://www.python.org/>
- Si suggerisce python3.6

Un terminale (riga di comando);

- Su Windows: cmd, PowerShell;
- Su MacOS: “Terminale”
- Su Linux: ...

Editor di testi (Geany, Sublime, Notepad++);

- Geany si scarica da <https://www.geany.org/>. Installare geany e geany-plugins!



“Interprete” Python;

- Spesso preinstallato su Linux e MacOS
- La versione per Windows si scarica da <https://www.python.org/>
- Si suggerisce python3.6



Un terminale (riga di comando);

- Su Windows: cmd, PowerShell;
- Su MacOS: “Terminale”
- Su Linux: ...



Editor di testi (Geany, Sublime, Notepad++);

- Geany si scarica da <https://www.geany.org/>. Installare geany e geany-plugins!



“Interprete” Python;

- Spesso preinstallato su Linux e MacOS
- La versione per Windows si scarica da <https://www.python.org/>
- Si suggerisce python3.6






Un terminale (riga di comando);

- Su Windows: cmd, PowerShell;
- Su MacOS: “Terminale”
- Su Linux: ...



Editor di testi (Geany, Sublime, Notepad++);

- Geany si scarica da <https://www.geany.org/>. Installare geany e geany-plugins!

-  “Interprete” Python;
 - Spesso preinstallato su Linux e MacOS
 - La versione per Windows si scarica da <https://www.python.org/>
 - Si suggerisce `python3.6`
-  Un terminale (riga di comando);
 - Su Windows: `cmd`, `PowerShell`;
 - Su MacOS: “Terminale”
 - Su Linux: ...
-  Editor di testi (Geany, Sublime, Notepad++);
 - Geany si scarica da <https://www.geany.org/>. Installare `geany` e `geany-plugins`!

Una raccomandazione:

Digitare carattere per carattere!

Interprete Python

Per avviare l'interprete di python in modalità terminale è sufficiente digitare

```
$ python
Python 3.7.3 (default, Mar 26 2019, 21:43:19)
[GCC 8.2.1 20181127] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Su Windows questo potrebbe portare a un errore di questo genere

```
C:\Users\jacop>python
"python" non è riconosciuto come comando interno o esterno,
un programma eseguibile o un file batch.
```

In questo caso, è sufficiente aggiungere la cartella di installazione di Anaconda alla variabile di ambiente PATH.

Python come calcolatrice - Operazioni di base

```
>>>1+2
3
>>>1-2
-1
>>>1.1+2
3.1
>>>3*4
12
>>>3.2*3
9.6
>>>3/2
1.5
>>>3//2
1
>>>3.0/2
1.5
>>>2**8
256
>>>27%8
3
```

Python come calcolatrice - Operatori relazionali e logici

```
>>>True == True
True
>>>False != False
False
>>>1 == 1
True
>>>3 >= 4
False
>>>3 < 4
True
>>>-2 < 3 < 4
True
>>>True and False
False
>>>True or False
True
>>>not False
True
>>>not (False or True)
False
```

Hello World!

```
>>>print('Hello World!')
Hello World!
>>>print(1+1)
2
>>>print('6 + 7 fa', 6+7)
6 + 7 fa 13
```

Intero

- Precisione infinita
- Si identifica dall'assenza del separatore decimale (1000)

Float

- Precisione finita (floating point)
- Si identifica dalla presenza del separatore decimale (1.0)

Booleano

- Solo due valori: True, False
- Un numero viene interpretato come True se è diverso da zero.

Stringa

- Sequenza di caratteri
- Non mutabile

Intero

- Precisione infinita
- Si identifica dall'assenza del separatore decimale (1000)

Float

- Precisione finita (floating point)
- Si identifica dalla presenza del separatore decimale (1.0)

Booleano

- Solo due valori: True, False
- Un numero viene interpretato come True se è diverso da zero.

Stringa

- Sequenza di caratteri
- Non mutabile

Intero

- Precisione infinita
- Si identifica dall'assenza del separatore decimale (1000)

Float

- Precisione finita (floating point)
- Si identifica dalla presenza del separatore decimale (1.0)

Booleano

- Solo due valori: True, False
- Un numero viene interpretato come True se è diverso da zero.

Stringa

- Sequenza di caratteri
- Non mutabile

Intero

- Precisione infinita
- Si identifica dall'assenza del separatore decimale (1000)

Float

- Precisione finita (floating point)
- Si identifica dalla presenza del separatore decimale (1.0)

Booleano

- Solo due valori: True, False
- Un numero viene interpretato come True se è diverso da zero.

Stringa

- Sequenza di caratteri
- Non mutabile

Intero

- Precisione infinita
- Si identifica dall'assenza del separatore decimale (1000)

Float

- Precisione finita (floating point)
- Si identifica dalla presenza del separatore decimale (1.0)

Booleano

- Solo due valori: True, False
- Un numero viene interpretato come True se è diverso da zero.

Stringa

- Sequenza di caratteri
- Non mutabile

Conversioni tra stringhe e interi

```
>>>int('15')
15
>>>int('FF', 16)
255
>>>int('10', 2)
2
>>>str(10)
'10'
```

Stringhe

```
>>>'ciao a tutti'  
'ciao a tutti'  
>>>"in questo modo si può inserire l'apostrofo"  
"in questo modo posso inserire l'apostrofo"  
>>>'in questo modo si possono inserire i doppi apici'  
'in questo modo si possono inserire i doppi apici'  
>>>'''così si può andare  
...a capo e scrivere stringhe lunghe'''  
'così si può andare\na capo e scrivere stringhe lunghe'  
>>>'ciao ' + 'mondo'  
'ciao mondo'  
>>>'na na ' *5  
'na na na na na na na na na na'
```

Tipizzazione dinamica: le variabili sono *puntatori* ad oggetti in memoria. Il tipo è una proprietà dell'oggetto piuttosto che di una variabile.

Forte controllo dei tipi a runtime: le conversioni implicite sono poche e limitate ai numeri.

```
>>>a = 1
>>>b = 2
>>>c = 0
>>>a+b
3
>>>a*b
2
>>>c = 'Totale della somma: ' + str(a+b)
>>>print(type(c), c)
<class 'str'> Totale della somma: 3
>>>a += 3
>>>c *= 2
>>>print(a, c)
4 Totale della somma: 3Totale della somma:3
```

Input da tastiera

```
>>>a = input()
ciao
>>>print(a)
ciao
>>>nome = input("Inserisci il tuo nome\n")
Inserisci il tuo nome
Jacopo
>>>print("Benvenuto", nome)
Benvenuto Jacopo
```

Eseguire un file .py

Da adesso in poi lasceremo da parte la modalità terminale, dedicandoci a scrivere programmi veri e propri su un editor di testo. I programmi, che avranno estensione .py, una volta completati saranno eseguiti tramite il seguente comando:

```
$ python nome_del_file.py
```

Prendere decisioni

Achtung! In python i blocchi sono definiti dall'indentazione.

```
# Cosa cambia?
if(condizione):
    istruzione_1
    istruzione_2

if(condizione):
    istruzione_1
istruzione_2
```

```
voto_esame = int(input("Inserire_voto"))

if(voto_esame < 18):
    print("Mi_dispiace_hai_bocciato_l'esame")
elif(18 < voto_esame < 26):
    print("Esame_passato!")
else:
    print("Ottimo_lavoro!")
```

Prendere decisioni

```
voto_scritto = int(input("Inserire_voto_scritto\n"))
voto_orale = int(input("Inserire_voto_orale\n"))

if((18 <= voto_scritto <= 30) and (18 <= voto_orale <= 30)):
    voto_finale = (voto_scritto + voto_orale)//2
    if(voto_finale == 30):
        print("Voto:_30_e_Lode")
    else:
        print("Voto:", voto_finale)
elif(voto_scritto > 30 or voto_orale > 30):
    print("Non_faccia_il_furbo!")
else:
    print("Si_presenti_al_prossimo_appello")
```

Prendere decisioni

La stringa vuota viene interpretata come False in un'istruzione condizionale. Il seguente programma prende in ingresso due stringhe e valuta se la seconda è contenuta nella prima.

```
stringa_1 = input("Inserisci_stringa_1")
stringa_2 = input("Inserisci_stringa_2")

if((not stringa_1) or (not stringa_2)):
    print("Una_delle_due_stringhe_è_vuota!")
elif(stringa_2 in stringa_1):
    print("La_stringa_2_è_contenuta_nella_stringa_1")
else:
    print("La_stringa_2_non_è_contenuta_nella_stringa_1")
```


Con una funzione è possibile riutilizzare una porzione di codice richiamandola tramite un nome identificativo. Abbiamo già visto alcuni esempi di funzioni

- `print()`
- `input()`
- `type()`

Altre funzioni predefinite (o built-in) sono:

- `abs()`
- `round()`
- `len()`
- `id()`

Per definire una funzione occorre rispettare la seguente sintassi

```
def nome_funzione(parametro_1, parametro_2):  
    #Istruzioni  
    return risultato  
  
# Richiamare la funzione dal codice  
r = nome_funzione(a, b) # a e b sono i valori assegnati ai parametri  
#...  
k = nome_funzione(c, d)
```

Creiamo una funzione che calcola superficie laterale, superficie totale e volume di un cilindro:

```
def sup_vol_cilindro(raggio, altezza = 1):
    pi = 3.1415
    if(raggio < 0 or altezza < 0):
        return 0,0,0
    else:
        area_base = pi * (raggio**2)
        supl = altezza * (2 * pi * raggio)
        supt = supl + area_base*2
        volume = area_base*altezza
        return supl, supt, volume

# Richiamare la funzione dal codice
print(sup_vol_cilindro(altezza = 5, raggio = 6))
a,b,c = sup_vol_cilindro(4)
```

Creiamo una funzione che verifica se un anno è bisestile. Sono bisestili:

- Gli anni non secolari il cui numero è multiplo di 4;
- Gli anni secolari il cui numero è multiplo di 400;

```
def is_bisestile(a):  
    if(anno <= 0):  
        return False  
    else:  
        if ((a%400 == 0) or ((a % 100 !=0) and (a%4 == 0))):  
            return True  
        else:  
            return False
```

Per casa eseguire gli esercizi da 1 a 3 al seguente link
<https://www.programmareinpython.it/esercizi-python/>

Esercizi per casa: Equazione di secondo grado

Un'equazione di secondo grado è un'equazione del tipo $ax^2 + bx + c = 0$. Se $a \neq 0$ l'equazione ammette due soluzioni:

$$\Delta = b^2 - 4ac$$
$$x_1, x_2 = \frac{-b \pm \sqrt{\Delta}}{2a}$$

Che sono distinte e reali se $\Delta > 0$, coincidenti se $\Delta = 0$ o complesse coniugate se $\Delta < 0$. Creare una funzione che, a partire dai coefficienti a, b, c di un'equazione di secondo grado, restituisca le radici dell'equazione. Si faccia attenzione a valutare correttamente tutti i possibili scenari, in particolare quello in cui a è nullo.

Si ricorda che la radice quadrata si può calcolare con un elevamento a potenza:

$$\sqrt{n} = n^{1/2}$$

Esercizi per casa: Due Numeri

Si scriva un programma in linguaggio python che legga due numeri da tastiera, detti A e B, e determini le seguenti informazioni, stampandole a video:

- determini se B è un numero positivo o negativo
- determini se A è un numero pari o dispari
- calcoli il valore di $A + B$
- determini quale scelta dei segni nell'espressione $(\pm A) + (\pm B)$ porta al risultato massimo, e quale è questo valore massimo.

Suggerimento: Nell'ultimo punto, il valore massimo della somma di A e B si può ottenere sommando il valore assoluto di A e di B.

Esercizi per casa: Classificazione triangolo

Si scriva un programma in linguaggio python che legga da tastiera i valori delle lunghezze dei tre lati di un triangolo (detti A, B e C), e determini:

- se il triangolo è equilatero (3 lati uguali)
- se il triangolo è isoscele (2 lati uguali)
- se il triangolo è scaleno (tutti i lati di lunghezze diverse)
- se il triangolo è rettangolo

Esercizio: il quizzone

Realizzare un programma che ponga, una dopo l'altra, 3 domande e ne verifichi la risposta. Al termine viene stampato il punteggio, +1 per ogni risposta corretta, -0.5 per ogni risposta sbagliata.

Per semplificare la vita, riportiamo alcuni metodi (Spoiler! Parleremo più avanti di cosa è un metodo) applicabili alle stringhe che possono fare comodo:

- `isupper()`: restituisce True se tutti i caratteri sono maiuscoli;
- `islower()`: restituisce True se tutti i caratteri sono minuscoli;
- `lower()`: restituisce una stringa identica a quella in ingresso con caratteri minuscoli;
- `upper()`: restituisce una stringa identica a quella in ingresso con caratteri maiuscoli;

La sintassi per applicare un metodo a un oggetto è:

```
string.method()
a = 'ciao'
a.islower()      # True
a.isupper()      # False
b = a.upper()    # b = 'CIAO'
```

- Note sulla migrazione a Python 3

http://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html

- Installazione di Python su Windows e altri trucchetti interessanti

<https://gist.github.com/mallegrini/>

- Quick reference python3 https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf

Grazie per l'attenzione!

GOLEM - Gruppo Operativo Linux Empoli



GOLEM – Gruppo Operativo Linux Empoli
presso “La Vela – Margherita Hack”
via Magolo, 32 – 50053 Empoli (FI)
tutti i martedì sera dalle 21.30 alle 24.00



Crediti & licenza

Questa serata è offerta da Jacopo e Luca a partire dal materiale di giuliof (GOLEM), utilizzando un template \LaTeX a cura di giomba.

Materiale rilasciato sotto GPL3 presso golem.linux.it