

Python 101

“giuliof”

GOLEM

23 maggio 2018





Figura 1: Guido van Rossum



Figura 2: I Monty Python

python™



Perché imparare a programmare?

Perché in Python

Vediamo degli esempi...

“Interprete” Python;

- Spesso preinstallato su Linux e MacOS
- La versione per Windows si scarica da <https://www.python.org/>
- Si suggerisce `python 2.7`

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 `python 2.7`



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 `python 2.7`



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 `python 2.7`



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!

Hello World

```
print 'Hello_World!'
print "Anche_questa_frase_sara'_scritta_dal_computer"
print "E'_una_buona_idea_non_scrivere lettere_con_gli_accenti"
```

Salviamo il file come "first.py"

Lo eseguiamo scrivendo sul terminale `$ python first.py`

```
Hello World!
Anche questa frase sara' scritta dal computer
E' una buona idea non scrivere lettere con gli accenti
```

Codifiche e codici magici

```
#coding: utf-8
#!/usr/bin/python2
print 'Hello_World!'
print "Anche_questa_frase_sarà_scritta_dal_computer"
print "Così_possiamo_usare_tutte_le_lettere_strane_àèìòù"
```

- `#coding: utf-8` permette di scrivere caratteri non-ASCII (lettere accentate ed altri simboletti). *"Fanne buon uso"*.
- `#!/usr/bin/python2` è uno *shabang*. Permette di lanciare lo script anche facendo doppio clic.

Codifiche e codici magici

```
#coding: utf-8
#!/usr/bin/python2
print 'Hello_World!'
print "Anche_questa_frase_sarà_scritta_dal_computer"
print "Così_possiamo_usare_tutte_le_lettere_strane_àèìòù"
```

- `#coding: utf-8` permette di scrivere caratteri non-ASCII (lettere accentate ed altri simboletti). *"Fanne buon uso"*.
- `#!/usr/bin/python2` è uno *shabang*. Permette di lanciare lo script anche facendo doppio clic.

Codifiche e codici magici

```
#coding: utf-8
#!/usr/bin/python2
print 'Hello_World!'
print "Anche_questa_frase_sarà_scritta_dal_computer"
print "Così_possiamo_usare_tutte_le_lettere_strane_àèìòù"
```

- `#coding: utf-8` permette di scrivere caratteri non-ASCII (lettere accentate ed altri simboletti). *"Fanne buon uso"*.
- `#!/usr/bin/python2` è uno *shabang*. Permette di lanciare lo script anche facendo doppio clic.

Commenti

- Proviamo ad inserire il carattere # all'inizio di una o più linee;
- Eseguiamo nuovamente il programma;
- I commenti aiutano a prendere appunti su cosa fa una parte di codice, potrebbero servire in futuro!

```
## Stampiamo un po' di frasi sullo schermo
print 'Hello_World!'

## Questa frase non mi piace piu'
#print "Anche questa frase sara' scritta dal computer"
print "E' _una_buona_idea_non_scrivere_lettere_con_gli_accenti"
```

Operazioni

```
# somma
print 324 + 755
# sottrazione
print 111-110
# divisione
print 30/5
# moltiplicazione
print 324*33
# tutto insieme
print (324*4)+38/(3-1)
# operazioni logiche
print 11 <= 11
print 3 == 3
```

Nota: conviene sempre utilizzare le parentesi nei calcoli matematici complessi

```
print "Scrivo", "qualcosa", "separando", "le", "singole", "parole"  
print "Sei_per_sette", 6*7, "piu'_due", (6*7)+2  
print "42_e'_minore_di_7?", 42 < 7
```

Quando si vogliono stampare a schermo più informazioni, separate da spazi, è sufficiente separarle con una virgola.

Assegniamo dei nomi a dei valori per sapere di cosa si tratta

```
## Applicazione banca
cognome = 'Verdirame'
nome = 'Augusto'
saldo = 100.0

print 'Nome_e_Cognome', nome, cognome
print 'Saldo_attuale', saldo

# faccio un prelievo
saldo = saldo - 10.0
print 'Prelievo:_nuovo_saldo', saldo

# interessi del 5%
interessi = 5.0
saldo = saldo + saldo * interessi / 100

print 'Nuovo_saldo', saldo
```


C'è un secondo modo per stampare a schermo le variabili

```
cognome = 'Verdirame'
nome = 'Augusto'
eta = 77
altezza = 1.70
occhi = 'marrone'
cod_fiscale = 'VRDGST40L08B157W'

print 'Scheda_utente'
print 'Nome:_%s_%s, _Eta:_%d' % (cognome,nome,eta)
print 'Occhi:_%s, _CF:_%s' % (occhi,cod_fiscale)
print 'Altezza:_%f' % altezza
```

Stampare i numeri

```
a = 12
b = 15.7
print 'In decimale: %d, ottale %o, esadecimale %x oppure %X' % (a, a, a, a)
print "%f_e' all'incirca %d" % (b, round(b))
print "La parte intera di %f_e' %d" % (b, int(b))
```

Specchietto riassuntivo

%d	Decimale
%f	Virgola mobile (float)
%o	Ottale
%x	Esadecimale (minuscolo)
%X	Esadecimale (maiuscolo)
%s	Stringa (testo)

- `round(x)` arrotonda il valore di `x`;
- `int(x)` tronca il valore di `x`;

- Salva il file come `stringhe.py`
- Eseguilo con `python stringhe.py`
- Modifica il programma per visualizzare l'altezza in centimetri

Possiamo creare un programma interattivo leggendo gli input

- Dalla tastiera
- Dal mouse
- Da un Arduino
- Da internet
- Da altri dispositivi...

Input da tastiera

```
print 'Come_ti_chiami?'
nome = raw_input()
print 'Quanti_anni_hai?'
eta = raw_input()
print 'Quanto_sei_alto?'
altezza = raw_input()

print 'Ti_chiami_%s,_hai_%s_anni_e_sei_alto_%s' % (nome, eta, altezza)
```

`raw_input` permette di leggere del testo inserito da tastiera (stringa), fino a che non si preme il tasto invio

Input da tastiera – forma compatta

```
nome = raw_input('Come_ti_chiami?')
eta = raw_input('Quanti_anni_hai?')
altezza = raw_input('Quanto_sei_alto?')

print 'Ti_chiami_%s,_hai_%s_anni_e_sei_alto_%s' % (nome,eta,altezza)
# L'istruzione successiva non funziona!
print "Fra_10_anni_avrai_%s_anni" % (eta+10)
```

Leggere dei numeri

```
nome = raw_input('Come ti chiami?')
eta = input('Quanti anni hai?')
altezza = input('Quanto sei alto?')

print 'Ti chiami %s, hai %d anni e sei alto %f' % (nome, eta, altezza)
print "Fra 10 anni avrai %d anni" % (eta+10)
```

Controllare l'input

```
print "Qual'e'_la_capitale_del_giappone?"
capitale = raw_input()

if capitale == 'tokyo':
    print "Risposta_esatta!"
else:
    print "Ritenta_sarai_piu'_fortunato!"
```


Il costrutto condizionale

```
if ... condizione .....
```

```
    istruzione1
```

```
    istruzione2
```

```
elif .... altra condizione .....
```

```
    istruzione 3
```

```
    istruzione 4
```

```
else:
```

```
    istruzione 5
```

```
    istruzione 6
```

- Python raggruppa le istruzioni in blocchi sfruttando l'**indentazione**
- È possibile utilizzare spazi (in numero a piacere) oppure tabulazioni, l'importante è essere coerenti

Confronti e condizioni logiche

```
a > b      # a e' piu' grande di b
a < b      # a e' piu' piccolo di b
a == b     # a e' uguale a b
a <= b     # a e' minore o uguale a b
a >= b     # a e' maggiore o uguale a b
a != b     # a e' minore o uguale a b

# Per unire insieme piu' confronti...
a > 0 and b > 0  # voglio SIA a SIA b positivi
a > 5 or a < -5  # scarto tutti i numeri fra -5 e 5
not (a > 0)     # voglio numeri negativi (zero compreso)

# condizioni particolari
True      # condizione sempre vera
False     # condizione sempre falsa
```

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

- 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 giuliof (giulio@glgprograms.it)
a partire dal materiale di Fiore e con contributi di lucam
utilizzando un template \LaTeX a cura di giomba.
Materiale rilasciato sotto GPL3 presso golem.linux.it