Python

Loriano Storchi

loriano@storchi.org

http::/www.storchi.org/

I linguaggi di programmazione

- Abbiamo visto che i linguaggi di programmaione sono:
  - Dichiarativi
    - Logici
    - Funzionali
  - Imperativi
    - Procedurali
    - Orientati agli oggetti

- Python e’ imperativo sia procedurale che object-oriented

I linguaggi di programmazione

- La tipizzazione: puo' essere **dinamica o statica e forte o debole**
- Tipizzazione debole e dinamica ad esempio:

```
[redo@banquo tipiz (master)]$ cat test.pl
$i = 2 + "3";
print $i, "\n";
[redo@banquo tipiz (master)]$ perl test.pl
5
[redo@banquo tipiz (master)]$
```

- Python tipizzazione dinamica e forte

L’interprete Python

- Tantissimo materiale/documentazione disponibile (infatti la presente introduzione e’ fortemente basata su:

  http://tdc-www.harvard.edu/Python.pdf
- Python lo trovate pronto all’uso sia su Linux che su Mac OS X. Per Windows potete trovare facilmente i binari al seguente URL: http://python.org/

- Tantissimi moduli (librerie) disponibili, nel nostro caso alcune utili/interessanti: numpy, matplotlib e pystat
- python 2.x vs python 3.x, noi useremo python 2.x la versione 3.x ha introdotto alcuni cambiamenti che non la rendono piu’ compatibile con la 2.x che rimane pero’ ancora la piu’ diffusa

L’interprete

```
[redo@virtualinux ~]$ python
Python 2.7.10 (default, Jun 20 2016, 14:45:40)
[GCC 5.3.1 20160406 (Red Hat 5.3.1-6)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> (4*5)/3
6
>>> exit()
[redo@virtualinux ~]$
```

Altra possibilita’ e’ semplicemente quella di scrivere il sorgente in un file ASCII e successivamente :

$ python file.py

Hello world

- Il classico programma usato per illustrare le basi sintattiche di un qualsiasi linguaggio di programmazione

```
[redo@banquo helloworld (master)]$ cat > hello.py
print "Hello World"
^C
[redo@banquo helloworld (master)]$ python hello.py
Hello World
[redo@banquo helloworld (master)]$
```

Oltre Hello, World

- Vediamo un semplice codice che ci permette di illustrare alcune delle caratteristiche base della sintassi python:

```
[redo@banquo helloworld (master)]$ cat oltrehw1.py
x = 34 - 23 # commentare il codice
y = "Hello"
z = 3.45

if z == 3.45 or y == "Hello":
    x = x + 1
    y = y + " World"

print "Valore di x: ", x
print "Valore di y: ", y

print z
[redo@banquo helloworld (master)]$ python oltrehw1.py
Valore di x:  12
Valore di y:  Hello World
3.45
[redo@banquo helloworld (master)]$
```

La basi minime

- Guadando il codice precedente possiamo capire alcune cose della sintassi del python:
  - Posso mettere commenti usando il carattere #
  - = e' usate per assegnare valori alle variabili mentre == e' l'operatore che server confrontare valori
  - Per fare operazioni fra numeri e variabili poso usare i nomali operatori +, -, *, /
  - Gli operatori logici invece sono : and, or, not
  - print e' il comando base che serve per stampare a video
  - Le variabili non devono essere dichiarate esplicitamente la prima volta che che assegno un valore la variabile viene creata e gli viene assegnato un tipo
  - L'operatore + puo' essere usato anche per concatenare le stringhe

Le basi minime

- Non c’e’ un carattere di fine linea, se una linea di codice deve essere spezzata su piu’ righe si usa \
- Di default i numeri sono interi quindi z = 5 / 2 dara’ come risultato 2

```
[redo@banquo helloworld (master)]$ cat oltrehw2.py
z = 5 / 2
print z

z = 5.0 / 2.0
print z
[redo@banquo helloworld (master)]$ python oltrehw2.py
2
2.5
[redo@banquo helloworld (master)]$
```

I Loop

- Per identificare blocchi di codice in python si usano gli spazi vuoti , e non ad esempio le { } come in C/C++
- Python e’ case sensitive

```
ca[redo@banquo helloworld (master)]$ cat oltrehw3.py
N = 0

for n in range(0,10):
  N = N + n
  print n

print "Valore finale: ", N
[redo@banquo helloworld (master)]$ python oltrehw3.py
0
1
2
3
4
5
6
7
8
9
Valore finale:  45
[redo@banquo helloworld (master)]$
```

If … then … else

- Il costrutto if then else in python

```
[redo@banquo helloworld (master)]$ cat oltrehw4.py
si = input ("inserisci un numero: " )
i = float(si)

if i < 0 :
    print "numero inferiore a zero"
elif i == 0:
    print "inseiro uguale a zero"
else:
    print "numero maggiore di zero"


[redo@banquo helloworld (master)]$ python oltrehw4.py
inserisci un numero: 0.5
numero maggiore di zero
[redo@banquo helloworld (master)]$
```

Esempio di procedure numeriche

- In generale e' possibile trovare un algoritmo risolutivo per molti problemi, ma non tutti (problema di computabilita'). Ad esempio calcolare le soluzioni di un'equazione di secondo grado:

INSERISCI VALORI DI A, B, C

D = (B*B)-(4 * A * C)

SE D >= 0.0

  SOL1 = (-1,0 * B + SQRT(D)) / (2,0 * A)

  SOL2 = (-1,0 * B - SQRT(D)) / (2,0 * A)

  STAMPA SOL1 E SOL2

Altrimenti

  STAMPA non ci sono soluzioni reali

Programma in Python

```
Import  math
ai = input("insert a:")
a = float(ai)
bi = input("insert b:")
b = float(bi)
ci = input("insert c:")
c = float(ci)
print "a = ", a, " b = ", b, " c = ", c
delta = math.pow(b, 2.0) - (4.0 * a * c)
if (delta >= 0):
  tn = math.sqrt(delta)
  sol1 = ((-1.0 * b) + tn) / (2.0 * a)
  sol2 = ((-1.0 * b) - tn) / (2.0 * a)
  print sol1, sol2
else:
  print "No real solutions"
```

ESERCIZIO 1

- Scrivere un programma in python che letti in input 10 numeri ne calcoli il valore medio e stampi a video il risultato

```
[redo@banquo esercizipython (master)]$ python ex1.py
inserisci il numero 1 45
inserisci il numero 2 67
inserisci il numero 3 84
inserisci il numero 4 2
inserisci il numero 5 4
inserisci il numero 6 6
inserisci il numero 7 7
inserisci il numero 8 8
inserisci il numero 9 9.0
inserisci il numero 10 13.0
la somma:  245.0
valore medio:  24.5
[redo@banquo esercizipython (master)]$
```