

Esercizi in Laboratorio

Informatica@SEFA 2018/2019 - Laboratorio 1

Massimo Lauria <massimo.lauria@uniroma1.it>
<http://massimolauria.net/courses/infosefa2018/>

Lunedì, 8 Ottobre 2018

Funzioni che restituiscono un valore

Una funzione può calcolare un valore e “restituirlo” al codice chiamante.

```
# Questa funzione restituisce un numero 1
def funzione_chiamata(N): 2
    """ Raddoppio il mio input""" 3
    return 2*N 4
5
# Questa funzione non restituisce nessun valore 6
def funzione_chiamante(): 7
    temp = funzione_chiamata(10) 8
    print("Dieci per due è:",temp) 9
10
    temp = funzione_chiamata(21) 11
    print("Ventuno per due è:",temp) 12
13
# Esegue la funzione chiamante, che a sua volta esegue la 14
# funzione chiamata. 15
funzione_chiamante() 16
```

return <espressione>

- ▶ Interrompe **immediatamente** l'esecuzione della funzione

```
return qualcosa 1  
print("Questa istruzione non verrà mai eseguita") 2
```

- ▶ La funzione chiamante ottiene il valore dell'espressione
- ▶ La funzione chiamante può (ma non deve) salvare il valore in una variabile o usarlo in un'espressione

```
V = funzione_chiamata(12) 1
```

```
( funzione_chiamata(11) * 51 ) + ( 5 // 2 ) 1
```

Esercizi e test

Tre esercizi per oggi

1. scrivere **un** programma python contenente
 - le funzioni che risolvono i tre esercizi
 - nient'altro
 - il file deve chiamarsi `lab01.py`
2. scrivete le vostre funzioni nel file `lab01.py`
3. scaricate il file test `test_lab01.py`
4. eseguite, nella cartella che contiene entrambi,

```
$ python3 test_lab01.py
```

5. migliorate fino a che non ottenete una cosa **COME**

```
.....  
-----  
Ran 8 tests in 0.001s  
  
OK
```

Esito di ogni test

La prima riga di output del test contiene, per esempio

```
.FF.E...
```

- ▶ . per ogni test passato
- ▶ F per ogni test fallito
- ▶ E per ogni test che non riesce ad essere eseguito

Esercizio 1

Scrivere una funzione

```
scontato(prezzo,sconto)
```

che, dato un prezzo ed uno sconto da 0 a 100, **restituisca** il prezzo scontato. Ad esempio se il prezzo è 500 e lo sconto è 20 (che vuol dire 20 per cento), allora il prezzo finale restituito è 400.

Esercizio 2

```
superficie_cilindro(raggio, altezza)
```

Essenzialmente una reimplementazione dell'esempio visto in classe. Dati l'altezza del cilindro ed il raggio della base, la funzione deve **restituire** la superficie del cilindro. E si deve utilizzare l'approssimazione 'math.pi'.

Esercizio 3

```
volume_cilindro(raggio,altezza)
```

Dati l'altezza del cilindro ed il raggio della base, la funzione deve **restituire** il volume del cilindro. E si deve utilizzare l'approssimazione 'math.pi'.

Esito di ogni test

La prima riga di output del test contiene

- ▶ . per ogni test passato
- ▶ F per ogni test fallito
- ▶ E per ogni test che non riesce ad essere eseguito

```
.....
```

```
-----  
Ran 8 tests in 0.001s
```

```
OK
```

Esempio di test fallito

```
.....F.
```

```
=====
FAIL: test_sconti (__main__.TestLab01Sconto)
-----
Traceback (most recent call last):
  File "test_lab01.py", line 54, in test_sconti
    msg="\nTEST FAIL> 456 euro con {}% di sconto è {}, e non {}.".format(s,val)
AssertionError: 0 != 455.999999999999994 within 7 places :
TEST FAIL> 456 euro con 0% di sconto è 455.999999999999994, e non 0.
-----
```

Esempio di test non eseguito (causa errore)

```
.....EE
```

```
=====
ERROR: test_sconti (__main__.TestLab01Sconto)
-----
```

```
Traceback (most recent call last):
```

```
  File "test_lab01.py", line 51, in test_sconti
```

```
    val_calcolato = scontato(456,s)
```

```
  File "/Users/massimo/lavori/didattica/2018.IntroInformatica/lectures/lab01.py", line 45, in scontato
```

```
    return prezzo*(100.0-sconto)/0.0
```

```
ZeroDivisionError: float division by zero
```

```
=====
ERROR: test_sconti_zero (__main__.TestLab01Sconto)
-----
```

```
Traceback (most recent call last):
```

```
  File "test_lab01.py", line 58, in test_sconti_zero
```

```
    val_calcolato = scontato(0,s)
```

```
  File "/Users/massimo/lavori/didattica/2018.IntroInformatica/lectures/lab01.py", line 45, in scontato
```

```
    return prezzo*(100.0-sconto)/0.0
```

```
ZeroDivisionError: float division by zero
-----
```