



Python: Συναρτήσεις

Ενότητες

- Συναρτήσεις
- Ορισμός και κλήση συναρτήσεων
- Παράμετροι σε συνάρτηση
- Επιστροφή συνάρτησης
- Εμβέλεια
- Ορίσματα συναρτήσεων
- Αναδρομή
- Συνάρτηση main()
- Αρθρώματα (Modules)
- Παραδείγματα

Συναρτήσεις

- Οι συναρτήσεις στον προγραμματισμό είναι επαναχρησιμοποιήσιμα μέρη προγραμμάτων
- Μια συνάρτηση μας επιτρέπει να δίνουμε ένα όνομα σε μια ομάδα εντολών και να την εκτελούμε χρησιμοποιώντας το όνομά της οπουδήποτε στο πρόγραμμά μας και για όσες φορές θέλουμε
- Μπορούν να δεχθούν είσοδο και να παραγάγουν έξοδο (επιστρεφόμενη τιμή)
- Μπορούν να αποθηκευτούν και σε αρχεία για μελλοντική χρήση
- Διευκολύνουν τη συγγραφή, ανάγνωση, κατανόηση και διόρθωση ενός προγράμματος

Ορισμός και Κλήση Συνάρτησης

- Ορισμός

```
def όνομα_συνάρτησης():  
    ... εντολές .....
```

```
def sayHello():           # ορισμός συνάρτησης  
    print('Hello')  
  
sayHello()                # κλήση συνάρτησης
```

- Κλήση

```
όνομα_συνάρτησης()
```

```
>>> -----  
>>>  
Hello
```

Ορισμός και Κλήση Συνάρτησης

- Παράδειγμα

```
1 def univ():
2     print('University of Ioannina')
3     print('Dept. of Computer Science & Engineering')
4
5 def sum10():
6     the_sum=0
7     for i in range(1,11):
8         the_sum+=i
9     return the_sum
10
11 univ()
12 x=sum10()
13 print(x)
14 y=3*sum10()-(4+sum10())/2
```

Παράμετροι σε συνάρτηση

- Μία συνάρτηση μπορεί να δεχθεί **παραμέτρους (formal parameters ή απλά parameters)**, οι οποίες αποτελούν τιμές στη συνάρτηση ως είσοδος
- Ορίζονται κατά την κλήση της συνάρτησης και ονομάζονται **ορίσματα (actual arguments ή απλά arguments)**

Ορισμός

```
def compare(a,b):  
    ... εντολές .....
```

Κλήση

```
compare(1,3)
```

Παράμετροι σε συνάρτηση

- Παράδειγμα

```
def compare(a, b): # ορισμός της συνάρτησης compare με παραμέτρους a, b
    if a > b:
        print(a, '>', b)
    elif a < b:
        print(a, '<', b)
    else:
        print(a, '=', b)

compare(7,9) # κλήση συνάρτησης με ορίσματα απευθείας τιμές
x = 12
y = 5
compare(x,y) # κλήση συνάρτησης με ορίσματα μεταβλητές
compare(x,12) # κλήση συνάρτησης με ορίσματα μεταβλητή και τιμή
```

```
>>> =====
>>>
7 < 9
12 > 5
12 = 12
```

Παράμετροι σε συνάρτηση

- Με απλά λόγια:
 - **στον ορισμό** μιας συνάρτησης έχουμε **παραμέτρους** (ονομασίες μεταβλητών)
 - **στην κλήση** μιας συνάρτησης έχουμε **ορίσματα** (τιμές ή μεταβλητές στις οποίες έχουν εκχωρηθεί τιμές)

Ορισμός

```
def compare(a,b):  
    ... εντολές .....
```

Κλήση

```
compare(1,3)
```


Παράμετροι σε συνάρτηση

- Πέρασμα μέσω αναφοράς (pass by reference)

```
def compare(a, b): # ορισμός της συνάρτησης compare με παραμέτρους a, b
    if a > b:
        print(a, '>', b)
    elif a < b:
        print(a, '<', b)
    else:
        print(a, '=', b)

x = 12
y = 5
compare(x, y) # κλήση συνάρτησης με ορίσματα μεταβλητές
```

- πριν την κλήση `compare(x, y)`

x → 12
y → 5

- κατά την κλήση `compare(x, y)`

x → 12 ← a
y → 5 ← b

Παράμετροι σε συνάρτηση

- Πέρασμα μέσω αναφοράς (pass by reference)

Τι θα τυπωθεί και γιατί,
όταν εκτελεστεί το πρόγραμμα;

```
def set10(x):  
    x = 10  
  
y = 0  
set10(y)  
print(y)
```

Επιστροφή συνάρτησης

- Μία συνάρτηση μπορεί να επιστρέφει και τιμή, η οποία μπορεί να εκχωρηθεί σε μία μεταβλητή ή ακόμη και να χρησιμοποιηθεί ως μέρος μιας έκφρασης
- Η επιστροφή τιμής γίνεται με την εντολή **return**

Επιστροφή συνάρτησης

- Παράδειγμα 1

```
def absoluteValue(x):  
    if x < 0:  
        return -x  
    else:  
        return x  
  
print(absoluteValue(5))  
print(absoluteValue(-5))
```

```
>>> -----  
>>>  
5  
5
```

Επιστροφή συνάρτησης

- Παράδειγμα 2

```
def add(x, y):  
    return x + y  
  
print(add(7, 5))  
print(add('Hello ', 'World'))
```

```
>>> -----  
>>>  
12  
Hello World
```

Επιστροφή συνάρτησης

- Παράδειγμα 3

```
def add2(x, y):  
    print(x + y)  
  
add2(7, 5)  
sum = add2(7, 5)  
print(sum)
```

```
>>> -----  
>>>  
12  
12  
None
```

Η None υποδεικνύει ότι η συνάρτηση add2 δεν επιστρέφει τιμή

Εμβέλεια

- Εμβέλεια μεταβλητών

Τι θα τυπωθεί και γιατί,
όταν εκτελεστεί το πρόγραμμα;

```
1 def inc(x):  
2     print(x)  
3     x=x+1  
4     print(x)  
5  
6 a=3  
7 print(a)  
8 inc(a)  
9 print(a)
```

Εμβέλεια

- Εμβέλεια μεταβλητών

Τι θα τυπωθεί και γιατί,
όταν εκτελεστεί το πρόγραμμα;

```
1 def inc(x):  
2     print(x)  
3     x=x+1  
4     print(x)  
5  
6 a=3  
7 print(a)  
8 inc(a)  
9 print(a)
```

```
3  
3  
4  
3
```


Εμβέλεια

- Η εμβέλεια μιας μεταβλητής ή μιας συνάρτησης αφορά στην περιοχή του προγράμματος στην οποία μπορεί να προσπελαστεί ή με άλλα λόγια να είναι ορατή
- Όταν δημιουργούμε μία μεταβλητή μέσα σε μία συνάρτηση, αυτή ονομάζεται **τοπική μεταβλητή (local variable)** και «υπάρχει» μόνο μέσα στη συνάρτηση, δεν μπορούμε να τη χρησιμοποιήσουμε έξω από αυτή
- Οι τοπικές μεταβλητές «ζουν» όσο διαρκεί η εκτέλεση της συνάρτησης στην οποία έχουν οριστεί, και καταστρέφονται, όταν ολοκληρωθεί η εκτέλεση της συνάρτησης
- Κάθε νέα κλήση μιας συνάρτησης δημιουργεί νέες τοπικές μεταβλητές

Εμβέλεια

- Παράδειγμα

```
def add3(x, y):  
    s = x + y  
    return s  
  
print(add3(7, 3))  
print(s)
```

```
>>> ----- RESTART -----  
>>>  
10  
Traceback (most recent call last):  
  File "C:/Users/User/Desktop/code/6.6.py", line 7, in <module>  
    print(s)  
NameError: name 's' is not defined
```

Εμβέλεια

- Οι μεταβλητές που δηλώνονται έξω από τις συναρτήσεις ενός προγράμματος ονομάζονται **καθολικές μεταβλητές (global variables)**
- Μπορούν να προσπελαστούν από οποιοδήποτε σημείο μέσα στο πρόγραμμα συμπεριλαμβανομένων και των συναρτήσεων

Εμβέλεια

- Άσκηση 1

Τι θα τυπωθεί και γιατί,
όταν εκτελεστεί το πρόγραμμα;

```
name = 'Mariza'
def sayHello():
    print('Hello ' + name)

def changeName(new_name):
    name = new_name

sayHello()
changeName('Katerina')
sayHello()
```

Εμβέλεια

- Άσκηση 1

Τι θα τυπωθεί και γιατί,
όταν εκτελεστεί το πρόγραμμα;

```
name = 'Mariza'  
def sayHello():  
    print('Hello ' + name)  
  
def changeName(new_name):  
    name = new_name  
  
sayHello()  
changeName('Katerina')  
sayHello()
```

```
>>> -----  
>>>  
Hello Mariza  
Hello Mariza
```

Εμβέλεια

- Άσκηση 2

Τι θα τυπωθεί και γιατί,
όταν εκτελεστεί το πρόγραμμα;

```
name = 'Mariza'  
def sayHello():  
    print('Hello ' + name)  
  
def changeName(new_name):  
    global name  
    name = new_name  
  
sayHello()  
changeName('Katerina')  
sayHello()
```

Εμβέλεια

- Άσκηση 2

Τι θα τυπωθεί και γιατί,
όταν εκτελεστεί το πρόγραμμα;

```
name = 'Mariza'
def sayHello():
    print('Hello ' + name)

def changeName(new_name):
    global name
    name = new_name

sayHello()
changeName('Katerina')
sayHello()
```

```
>>> =====
>>>
Hello Mariza
Hello Katerina
```

Ορίσματα Συναρτήσεων

- **Προεπιλεγμένα ορίσματα**
 - Σε μία συνάρτηση μπορούμε να έχουμε μερικές παραμέτρους προαιρετικές και να χρησιμοποιούμε προεπιλεγμένες τιμές, εάν ο χρήστης δεν θέλει να δώσει τιμές σε αυτές τις παραμέτρους
 - Ο καθορισμός προεπιλεγμένων τιμών ορισμάτων για παραμέτρους γίνεται τοποθετώντας μετά το όνομα της παραμέτρου στον ορισμό της συνάρτησης τον τελεστή εκχώρησης (=) ακολουθούμενο από την προεπιλεγμένη τιμή

Ορίσματα Συναρτήσεων

- Προεπιλεγμένα ορίσματα

```
def greet(name, greeting = 'Hello'):  
    print(greeting, name)  
  
greet('Antonis')  
greet('Antonis', 'Good bye')
```

```
>>> -----  
>>>  
Hello Antonis  
Good bye Antonis
```

Ορίσματα Συναρτήσεων

- **Ορίσματα με λέξεις κλειδιά**
 - Σε μία συνάρτηση μπορούμε να ορίσουμε μερικές παραμέτρους χρησιμοποιώντας λέξεις κλειδιά
 - Δίνουμε τιμές μόνο στις παραμέτρους που θέλουμε, οι υπόλοιπες θα παίρνουν τις προεπιλεγμένες τιμές

Ορίσματα Συναρτήσεων

- Ορίσματα με λέξεις κλειδιά

```
def func(a, b=1, c=2):  
    print('a =', a, 'και b =', b, 'και c =', c)  
  
func(5, 10)  
func(2, c=20)  
func(c=50, a=10)
```

```
>>> -----  
>>>  
a = 5 και b = 10 και c = 2  
a = 2 και b = 1 και c = 20  
a = 10 και b = 1 και c = 50
```

Ορίσματα Συναρτήσεων

- Παράδειγμα 1

```
1 def sentence(the_subject,the_verb,the_object):
2     return the_subject+' '+the_verb+' '+the_object
3
4 print(sentence('Peter','reads','a book'))
5
6 print(sentence(the_subject='Peter',the_verb='reads',
7               the_object='a book'))
8 print(sentence(the_verb='reads',the_object='a book',
9               the_subject='Peter'))
```

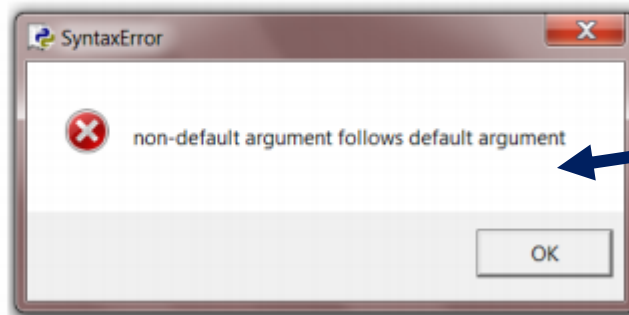
Ορίσματα Συναρτήσεων

- Παράδειγμα 2

```
1 def sentence_2(the_subject,the_verb,the_object='the newspaper'):  
2     return the_subject+' '+the_verb+' '+the_object  
3  
4 print(sentence_2(the_verb='reads',the_object='a book',  
5                 the_subject='Peter'))  
6 print(sentence_2(the_verb='reads',the_subject='Peter'))  
7  
8 print(sentence_2('Peter','reads'))
```

Τί παρατηρείτε;

```
1 def sentence_3(the_subject,the_verb='reads',the_object):  
2     return the_subject+' '+the_verb+' '+the_object
```



Η Python διαμαρτύρεται θυμίζοντάς μας ότι ακόμα και η ευελιξία πρέπει να έχει τα όριά της

Αναδρομή

- Η διαδικασία κατά την οποία μια συνάρτηση καλεί τον εαυτό της ονομάζεται **αναδρομή**, και η συνάρτηση λέγεται **αναδρομική συνάρτηση**
- Χαρακτηριστικό παράδειγμα ο υπολογισμός του παραγοντικού

$$n! = \begin{cases} 1 & \text{αν } n = 0 \\ n(n - 1)! & \text{αν } n > 0 \end{cases}$$

Αναδρομή

- Παράδειγμα: Υπολογισμός παραγοντικού

```
def factorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n-1)  
  
x = 4  
print(x, '! =', factorial(x))
```

```
>>> -----  
>>>  
4 ! = 24
```

Συνάρτηση main()

- Μία καλή πρακτική προγραμματισμού είναι ο ορισμός και η χρήση μιας τουλάχιστον συνάρτησης σε κάθε πρόγραμμα που γράφουμε και με όνομα το `main`
- Σε άλλες γλώσσες προγραμματισμού, όπως η C και η Java, η συνάρτηση `main` είναι μέρος της γλώσσας
- Η `main` δηλώνει την αρχή ενός προγράμματος και είναι προαιρετική στην Python

Συνάρτηση main()

- Παράδειγμα

```
def main():  
    pwd = input('Δώστε τον κωδικό πρόσβασης:')  
    if pwd == 'heraklion':  
        print('Είσοδος...')  
    else:  
        print('Λανθασμένος κωδικός πρόσβασης')  
  
main()
```

```
>>> ===== RESTART =====  
>>>  
Δώστε τον κωδικό πρόσβασης:heraklion  
Είσοδος...
```

Άρθρώματα (modules)

- Ένα άρθρωμα (module) είναι μία συλλογή σχετιζόμενων συναρτήσεων
- Ένα άρθρωμα αποθηκεύεται σε ένα αρχείο με κατάληξη **.py**
- Για να χρησιμοποιήσουμε ένα άρθρωμα σε ένα πρόγραμμα, θα πρέπει να το εισαγάγουμε με χρήση της εντολής **import**

Αρθρώματα (modules)

- Παράδειγμα

```
>>> import math
>>> print(math.pi)
3.141592653589793
>>> x = 3 + math.log10(100)
>>> print(x)
5.0
>>> degrees = 45
>>> radians = degrees / 360 * 2 * math.pi
>>> math.sin(radians)
0.7071067811865475
>>>
```

Παραδείγματα

- Παράδειγμα 1: Υπολογισμός απόλυτης τιμής

```
1 def absolute(x):  
2     if x<0:  
3         return -x  
4     else:  
5         return x
```

Παραδείγματα

- Παράδειγμα 2: Ύψωση σε δύναμη

```
1 def power(a,x):
2     if int(x)!=x:
3         return 'error: not supported'
4     elif x==0:
5         return 1
6     elif x>0:
7         p=1
8         for i in range(x):
9             p*=a
10        return p
11    elif a==0 and x<0:
12        return 'error: division by zero'
13    else:
14        p=1
15        for i in range(-x):
16            p/=float(a)
17        return p
```

Παραδείγματα

- Παράδειγμα 2: Λίστα θετικών και αρνητικών αριθμών

```
1 def pos_and_neg(L):
2     pos=[]
3     neg=[]
4     zero=0
5     for x in L:
6         if x>0:
7             pos.append(x)
8         if x<0:
9             neg.append(x)
10        if x==0:
11            zero+=1
12    return pos,neg,zero
```

Ασκήσεις

1. Γράψτε μια συνάρτηση που δέχεται ένα string, να το τυπώνει ανάποδα και να ενημερώνει για το πλήθος των κεφαλαίων και πεζών χαρακτήρων
2. Γράψτε ένα πρόγραμμα με τον ορισμό κατάλληλων συναρτήσεων για την εύρεση του μεγίστου μεταξύ 2 και 3 αριθμών
3. Γράψτε ένα πρόγραμμα με τον ορισμό κατάλληλων συναρτήσεων για την εύρεση του εμβαδού και τις περιμέτρου των ορθογωνίου παραλληλογράμμου, ορθογωνίου τριγώνου, κύκλου

Βιβλιογραφία

- Μανής, Γ., 2015. Εισαγωγή στον Προγραμματισμό με αρωγό τη γλώσσα Python. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: <http://hdl.handle.net/11419/2745>
- Αγγελιδάκης, Ν., 2015. Εισαγωγή στον προγραμματισμό με την Python, Ηράκλειο. [ηλεκτρ. βιβλ.] Διαθέσιμο στο: http://aggelid.mysch.gr/pythonbook/INTRODUCTION_TO_COMPUTER_PROGRAMMING_WITH_PYTHON.pdf