

ΚΕΦΑΛΑΙΟ 7 – ΣΥΝΑΡΤΗΣΕΙΣ ΣΤΗΝ PYTHON

ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ

Ένα υποπρόγραμμα είναι ένα κομμάτι προγράμματος που έχει γραφεί ξεχωριστά από το υπόλοιπο πρόγραμμα και επιτελεί ένα αυτόνομο έργο.

Κάθε υποπρόγραμμα πρέπει να έχει τα παρακάτω βασικά χαρακτηριστικά:

- α) Έχει μόνο ένα σημείο εισόδου από το οποίο δέχεται τα δεδομένα του.
- β) Το υποπρόγραμμα το οποίο καλεί ένα άλλο υποπρόγραμμα σταματάει την εκτέλεσή του όσο εκτελείται το καλούμενο υποπρόγραμμα. Μόνο ένα υποπρόγραμμα μπορεί να εκτελείται σε μια χρονική στιγμή.
- γ) Ο έλεγχος επιστρέφει στο υποπρόγραμμα το οποίο καλεί, όταν το καλούμενο υποπρόγραμμα σταματήσει να εκτελείται.

Καλές πρακτικές

1. Πριν ξεκινήσουμε να γράφουμε ένα πρόγραμμα, μελετάμε πώς αυτό μπορεί να αναλυθεί σε επιμέρους τμήματα και αποφασίζουμε για τα αντίστοιχα υποπρογράμματα.
2. Εξετάζουμε αν κάποια υποπρογράμματα, τα οποία έχουμε ήδη γράψει ή υπάρχουν σε έτοιμες βιβλιοθήκες προγραμμάτων, μπορούν να χρησιμοποιηθούν, για να κερδίσουμε χρόνο και να αποφύγουμε λάθη.
3. Προσπαθούμε κάθε υποπρόγραμμα να είναι όσο το δυνατόν πιο ανεξάρτητο από τα άλλα. Αυτό μας προφυλάσσει από λάθη στο πρόγραμμά μας και επιτρέπει τη χρήση του σε άλλα προγράμματα αργότερα.

Η Python παρέχει ένα μόνο τύπο υποπρογραμμάτων, τις συναρτήσεις, τις οποίες τις θεωρεί ως αντικείμενα.

Ορισμός και κλήση συνάρτησης

Ο ορισμός μιας συνάρτησης περιλαμβάνει το όνομά της και τις παραμέτρους εισόδου και καλείται από σημεία του προγράμματος μέσω της λειτουργίας που ονομάζεται κλήση (calling) της συνάρτησης. Η επικεφαλίδα ξεκινάει με τη δεσμευμένη λέξη def ακολουθούμενη από το όνομα της συνάρτησης, ένα ζευγάρι παρενθέσεων που προαιρετικά περικλείει μια διαχωριζόμενη με κόμματα λίστα παραμέτρων και τελειώνει με μια άνω κάτω τελεία. Κάτω από τη γραμμή αυτή τοποθετούνται, σε εσοχή, οι εντολές που καθορίζουν τη λειτουργία της συνάρτησης.

def όνομα(παράμετροι):

εντολές

Παραδείγματα

```
def hello(): # ορισμός συνάρτησης
```

```
    print ('Hello ΓΠ')
```

```
Hello() # κλήση συνάρτησης
```

Η συνάρτηση αυτή καλείται γράφοντας hello() και μας εμφανίζει στην οθόνη τη φράση Hello ΓΠ.

```
def add(par1, par2):
```

```
    result = par1 + par2
```

```
    return result
```

Η συνάρτηση αυτή επιστρέφει το άθροισμα των τιμών των δύο παραμέτρων . Το αποτέλεσμα αυτό ονομάζεται **επιστρεφόμενη τιμή**.

Χαρακτηριστικά συναρτήσεων

- Μια συνάρτηση πρέπει να έχει ορισθεί πριν χρησιμοποιηθεί.
- Οι εντολές (statements) μέσα στη συνάρτηση δεν εκτελούνται μέχρι να κληθεί.
- Μια κλήση συνάρτησης είναι μια παράκαμψη στη ροή εκτέλεσης. Αντί η ροή να πάει στην επόμενη εντολή, πηγαίνει στο σώμα (εντολές) της συνάρτησης, εκτελεί όλες τις δηλώσεις εκεί και μετά επιστρέφει για να συνεχίσει από εκεί που σταμάτησε.
- Η συνάρτηση επιστρέφει μία μόνο τιμή.

Κατηγορίες συναρτήσεων

α) αυτές που δεν τροποποιούν το αντικείμενο στο οποίο εφαρμόζονται:

```
>>> a = "Python"
>>> print (a)
Python
>>> print(a.upper( ))
PYTHON
```

β) αυτές που αλλάζουν το αντικείμενο στο οποίο καλούνται να ενεργήσουν:

```
>>> b = ['A', 'B', 'C', 'D']
>>> b.reverse()
>>> print b
['D', 'C', 'B', 'A']
```

γ) αυτές που επιστρέφουν αποτέλεσμα:

```
>>> def add2(par1, par2):
    result = par1 + par2
    return result
>>> print add2(1,2)
3
```

δ) αυτές που δεν επιστρέφουν κάποια τιμή αλλά εκτελούν ενέργειες μέσω των εντολών τους:
Πιθανή τέτοια συνάρτηση η... print

Μεταβλητές και παράμετροι

Ένα υποπρόγραμμα αποτελεί ένα ανεξάρτητο τμήμα προγράμματος που μπορεί να καλείται από οποιοδήποτε σημείο του προγράμματος ή άλλων υποπρογραμμάτων. Δέχεται τιμές από το τμήμα προγράμματος που το καλεί και μετά την εκτέλεση των εντολών του επιστρέφει σε αυτό νέες τιμές. Οι τιμές αυτές που μεταβιβάζονται από το ένα υποπρόγραμμα στο άλλο λέγονται παράμετροι και γενικά σε γλώσσες προγραμματισμού διακρίνονται σε **παραμέτρους Εισόδου** και **παραμέτρους Εξόδου**.

Παράμετροι συναρτήσεων

Μια συνάρτηση μπορεί να δεχθεί παραμέτρους, οι οποίες είναι τιμές που δίνονται / εισέρχονται στη συνάρτηση, έτσι ώστε αυτή να μπορεί να λειτουργήσει αξιοποιώντας τις.

Οι **παράμετροι** μοιάζουν με τις μεταβλητές, καθορίζονται μέσα στο ζευγάρι των παρενθέσεων, στον ορισμό της συνάρτησης και διαχωρίζονται με κόμμα, ενώ οι τιμές αυτών των μεταβλητών ορίζονται, όταν καλούμε τη συνάρτηση.

Οι τιμές με τις οποίες καλούμε μία συνάρτηση να εκτελεστεί λέγονται **ορίσματα**.

Παράδειγμα

```
def compare(x,y): #οι x, y είναι παράμετροι
    if x>y:
        print x,'>',y
    elif x==y:
        print x,'=',y
    else:
        print x,'<',y
compare (15,9) # κλήση συνάρτησης με ορίσματα τιμές
a=14
b=2
compare(a,b) # κλήση συνάρτησης με ορίσματα μεταβλητές
```

compare(10,b)) # κλήση συνάρτησης με ορίσματα μεταβλητές και τιμές.

Επιστρεφόμενη τιμή (return value)

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 η οποία σημαίνει: «επίστρεψε αμέσως τον έλεγχο ροής από τη συνάρτηση στο σημείο του προγράμματος που αυτή κλήθηκε και χρησιμοποίησε την τιμή της έκφρασης που ακολουθεί ως επιστρεφόμενη τιμή».

Παράδειγμα 1

```
def sum(x,y):  
    return x+y  
print sum(24,3)  
Print sum('Καλά', 'Χριστούγεννα')
```

Παράδειγμα 2

```
def apolith_timh(x):  
    if x>=0:  
        return x  
    else  
        return -x  
print apolith_timh(-25)  
print apolith_timh(25)
```

Εμβέλεια συναρτήσεων

Η εμβέλεια μιας μεταβλητής αναφέρεται στο τμήμα του προγράμματος στο οποίο μπορεί να προσπελαστεί ή με άλλα λόγια να είναι ορατή. Οι μεταβλητές μπορεί να είναι:

Απεριόριστης εμβέλειας Είναι ορατές και μπορούν να χρησιμοποιηθούν σε οποιοδήποτε τμήμα του προγράμματος ανεξάρτητα το που δηλώθηκαν. Χαρακτηρίζονται ως **καθολικές (global) μεταβλητές**.

Περιορισμένης εμβέλειας Ισχύουν μόνο για το υποπρόγραμμα στο οποίο δηλώθηκαν. Χαρακτηρίζονται ως **τοπικές (local) μεταβλητές**.

Τοπικές μεταβλητές

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.

ΠΡΟΣΟΧΗ!! Οι παράμετροι μιας συνάρτησης θεωρούνται επίσης τοπικές μεταβλητές. Αλλάζοντας την τιμή μιας μεταβλητής-παραμέτρου μέσα στη συνάρτηση, η αλλαγή αυτή εφαρμόζεται μόνο στην “περιοχή” της συνάρτησης και δεν αλλάζει την τιμή της μεταβλητής-ορίσματος στο πρόγραμμα απ' όπου κλήθηκε!

Παράδειγμα

```
def sum(x,y):  
    athr=x+y # Η athr είναι τοπική μεταβλητή  
    print 'Η τιμή της athr μέσα στη συνάρτηση είναι', athr  
    return athr  
athr=0  
sum(5,15)  
print 'Η τιμή της athr έξω από τη συνάρτηση είναι', athr  
Οι παράμετροι x,y και η μεταβλητή athr είναι τοπικές μεταβλητές στη συνάρτηση sum.
```

Όταν τρέξουμε το πρόγραμμα θα πάρουμε τα παρακάτω αποτελέσματα:

Η τιμή της athr μέσα στη συνάρτηση είναι 20

Η τιμή της athr έξω από τη συνάρτηση είναι 0

ΠΡΟΣΟΧΗ!!! Αν δεν δώσω την εντολή `athr=0` το πρόγραμμα θα «χτυπήσει» κατά την εκτέλεση του και θα μας εμφανίσει `name 'arthr' is not defined`.

Καθολικές (global) μεταβλητές

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

Αν θέλουμε μέσα σε μια συνάρτηση να αλλάξουμε την τιμή μιας καθολικής μεταβλητής, τότε πρέπει να δηλώσουμε στην Python ότι η μεταβλητή αυτή δεν είναι τοπική αλλά καθολική. Αυτό γίνεται με τη χρήση της **εντολής `global`** με την οποία γίνεται ξεκάθαρο ότι η μεταβλητή βρίσκεται σε ένα εξωτερικό τμήμα εντολών.

Παράδειγμα

```
name='black'
def f():
    global name
    name='white'
    print 'Μέσα στη συνάρτηση', name
print 'Έξω από τη συνάρτηση', name
f()
print 'Έξω από τη συνάρτηση', name
```

Αν τρέξουμε το πρόγραμμα αυτό θα μας εμφανίσει:

```
Έξω από τη συνάρτηση black
Μέσα στη συνάρτηση white
Έξω από τη συνάρτηση white
```