

## ΕΠΙΠΛΕΟΝ ΥΛΙΚΟ ΣΤΗΝ ΔΙΕΥΘΥΝΣΗ

[https://www.w3schools.com/python/python\\_functions.asp](https://www.w3schools.com/python/python_functions.asp)

### Δημιουργώντας τις δικές μας συναρτήσεις

Οι συναρτήσεις είναι μέρη προγραμμάτων που γράφονται μία φορά και χρησιμοποιούνται πολλές. Μας επιτρέπουν να δίνουμε ένα όνομα σε ένα σύνολο εντολών και να το εκτελούμε καλώντας το όνομά τους, οπουδήποτε στο πρόγραμμα και όσες φορές θέλουμε. Αυτή η διαδικασία ονομάζεται κλήση (call) της συνάρτησης.

Η python έχει αρκετές ενσωματωμένες συναρτήσεις, όπως την `int()`, τη `range`. `len()` κτλ

### Ορισμός συνάρτησης

Οι συναρτήσεις ορίζονται χρησιμοποιώντας τη λέξη `def`, από το `define` που σημαίνει ορίζω. Στη συνέχεια ακολουθεί ένα όνομα της συνάρτησης. Μετά προσθέτουμε ένα ζευγάρι παρενθέσεων που μπορούν να περικλείουν μερικά ονόματα μεταβλητών και η γραμμή τελειώνει με διπλή τελεία (:).

```
def <όνομα συνάρτησης> ( [ {λίστα παραμέτρων} ] ):  
    εντολές  
    [ return <αποτέλεσμα> ]
```

Οι αγκύλες [ ] σημαίνουν, πως, ό,τι περικλείεται μέσα σε αυτές, είναι προαιρετικό. Η λίστα των παραμέτρων μπορεί να είναι κενή. Μια συνάρτηση δεν είναι υποχρεωτικό να επιστρέφει κάποια τιμή.

Παραδείγματα:

```
def add(arg1, arg2):  
    result = arg1+arg2  
    return result  
def times3(arg):  
    ginomeno = 3*arg  
    return ginomeno  
  
>>> add(10, 18)  
28  
>>> add(10, 18.5)  
28.5  
>>> times3(10)  
30
```

Μπορούμε να συνδυάσουμε την κλήση συναρτήσεων, με το σκεπτικό ότι το αποτέλεσμα της μιας συνάρτησης μπορεί να αποτελέσει τα δεδομένα εισόδου μιας άλλης.

```
>>> times3(2.5)
7.5
>>> times3('python')
'python python python'
>>> times3( times3( 9 ' ) )
'9999999999'
>>> times3(add(add('ab','ba'),' '))
'abba abba abba'
```

Παρατήρησε ότι έχουμε ορίσει μία συνάρτηση, η οποία δέχεται όλους τους τύπους των ορισμάτων και η **λειτουργία της αλλάζει δυναμικά ανάλογα** με αυτά και αν δοθούν αριθμοί τους προσθέτει, ενώ αν δοθούν αλφαριθμητικά, τα συνενώνει.

### Κατηγορίες συναρτήσεων

Οι συναρτήσεις μπορούν να κατηγοριοποιηθούν με πολλούς τρόπους. Μια πρώτη κατηγορία συναρτήσεων είναι:

α) αυτές οι οποίες δεν τροποποιούν το αντικείμενο στο οποίο εφαρμόζονται, όπως:

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

β) εκείνες που μπορούν να αλλάξουν το αντικείμενο στο οποίο καλούνται

```
>>> b = [ 'a', 'b', 'c', 'd' ]
>>> print b.append('e')
None
>>> print b
['a', 'b', 'c', 'd', 'e']
```

## Παράμετροι συναρτήσεων

Μια συνάρτηση δέχεται δεδομένα μέσω των παραμέτρων και επιστρέφει τα αποτελέσματα μέσω άλλων ή και των ίδιων παραμέτρων στο πρόγραμμα ή σε άλλη συνάρτηση.

Οι παράμετροι καθορίζονται μέσα στο ζευγάρι των παρενθέσεων στον ορισμό της συνάρτησης και διαχωρίζονται με κόμμα. Όταν καλούμε τη συνάρτηση, δίνουμε και τις τιμές με τον ίδιο τρόπο, οι οποίες τιμές **ονομάζονται ορίσματα**. Τα ορίσματα δεν είναι υποχρεωτικά, δηλ. μπορεί να φτιάξουμε μια συνάρτηση χωρίς ορίσματα π.χ. η συνάρτηση `help()` που εμφανίζει ένα συγκεκριμένο κείμενο.

```
def ginomeno (a,b):  
    x = a * b  
    return x  
print ginomeno(5,10)
```

Σημείωση: Η μεταβίβαση παραμέτρων στην Python λειτουργεί με τέτοιο τρόπο, ώστε οποιαδήποτε αλλαγή στις παραμέτρους εντός της συνάρτησης **δεν έχει καμία επίδραση** στα ορίσματα μεταβλητές που έχουν οριστεί εκτός της συνάρτησης, όπως φαίνεται παρακάτω:

```
def increment(a, b):  
    a = a + 1  
    b = b + 1  
    return a+b  
# τέλος συνάρτησης  
z = 1  
w = 2  
  
q = increment( z, w )  
print z, w, q  
1 2 5
```

Τα ορίσματα `z`, `w` δεν αλλάζουν τιμή, παρόλο που οι αντίστοιχοι παράμετροι αυξάνονται εντός της συνάρτησης `increment`, όπως φαίνεται και από την τιμή που επιστρέφει η συνάρτηση και καταχωρείται στην `q`.