

## Βασικές Ασκήσεις από τα Εργαστήρια της PYTHON

### Συναρτήσεις – Random generator

► Η Python επιτρέπει να χρησιμοποιούμε πολλές μεταβλητές οποιουδήποτε τύπου στην αριστερή πλευρά μίας καταχώρισης, αρκεί ο αριθμός των τιμών ή μεταβλητών που βρίσκονται στην δεξιά πλευρά να είναι ίδιος με τον αριθμό των μεταβλητών.

```
>>> a,b,c,d='a',2,3,'z'  
>>> a  
'a'  
>>> b  
2  
>>> c  
3  
>>> d  
'z'  
>>>
```

► Η αντιμετάθεση τιμών μπορεί να γίνει χωρίς χρήση ενδιάμεσης μεταβλητής όπως κάνουμε στην C ή σε άλλες γλώσσες.

```
a = 1  
b = 2  
  
print a,b  
a , b = b , a  
print a,b
```

► Αυτό το πρόγραμμα δημιουργεί λίστα με ονόματα (Strings) τα οποία δέχεται από το πληκτρολόγιο. Σταματάει όταν για όνομα δώσουμε τη λέξη 'end':

```
#program 5.1  
  
lista=[]  
name=input('name: ')  
while(name!='end'):  
    lista.append(name)
```

```
name=input('name: ')
print(lista)
```

► Η παρακάτω εκδοχή του προγράμματος δημιουργεί λίστα με ονόματα, όπως το προηγούμενο πρόγραμμα και επιπλέον ζητάει από τον χρήστη ένα όνομα. Στη συνέχεια διατρέχει τη λίστα και ενημερώνει εάν το όνομα που ζητήθηκε βρίσκεται στη λίστα ή όχι.

```
#program 5.2
lista=[]
name=input('add name: ')
while(name!='end'):
    lista.append(name)
    name=input('add name: ')
print(lista)
name=input('search for: ')
found=False
for n in lista:
    if(n==name):
        found=True
        break
if(found):
    print('found!')
else:
    print('not found')
```

Η αναζήτηση του String name στη λίστα lista θα μπορούσε να γίνει με τη βοήθεια της συνάρτησης any:

```
if any(name in s for s in lista):
    print('found!')
else:
    print('not found')
```

► Το πρόγραμμα αυτό βοηθάει να υπολογιστεί πόσα χαρτονομίσματα και κέρματα αντιστοιχούν σε χρηματικό ποσό. Να τροποποιήσετε το while, έτσι ώστε η συνθήκη του while να γίνει (amount!=0) αντί για (True).

```
# program5.3

while(True):
    amount=int(input('Poso: '))
    if(amount==0):
        break;
    p500=amount/500
    y=amount%500
    p200=y/200
    y=y%200
    p100=y/100
    y=y%100
    p50=y/50
    y=y%50
    p20=y/20
    y=y%20
    p10=y/10
    y=y%10
    p5=y/5
    y=y%5
    p2=y/2
    y=y%2
    p1=y

    print('500 - %d'%p500)
    print('200 - %d'%p200)
    print('100 - %d'%p100)
    print('50 - %d'%p50)
    print('20 - %d'%p20)
    print('10 - %d'%p10)
    print('5 - %d'%p5)
    print('2 - %d'%p2)
    print('1 - %d'%p1)

print('telos')
```

► Στο πρώτο μάθημα είδαμε ότι εάν κάνουμε `import` μία βιβλιοθήκη, τότε μπορούμε να χρησιμοποιήσουμε τον κώδικα τον οποίο περιλαμβάνει, με βάση το παρακάτω παράδειγμα:

```
import calendar

year=int(input("Βάλτε έτος:"))
calendar.prcal(year)
```

Εδώ η βιβλιοθήκη λέγεται **calendar** και η συνάρτησή της την οποία καλέσαμε **prcal**.

► Μπορούμε όμως να δηλώσουμε εύκολα δικές μας συναρτήσεις για να τυποποιήσουμε διαδικασίες. Αυτό γίνεται με τον παρακάτω τρόπο:

```
def maxi(x,y):
    if (x > y):
        return x
    else:
        return y

print(maxi(1,2))
```

Η συνάρτηση `maxi` δέχεται δύο παραμέτρους, συγκρίνει τις τιμές τους και επιστρέφει την μεγαλύτερη τιμή από τις δύο. Στο παράδειγμα η σύγκριση γίνεται μεταξύ 1 και 2.

► Ένα string με τριπλά εισαγωγικά είναι ένα doc string, το οποίο είναι διαθέσιμο σαν πραγματικό String μέσα στο πρόγραμμα και δεν είναι σχόλιο:

```
def test():
    '''Print an answer'''
    print(42)

test()
help(test)
```

Το παραπάνω πρόγραμμα όταν τρέξει καλεί την συνάρτηση `test()`, τυπώνει 42 και όταν κληθεί η `help(test)` επιστρέφει το doc string που υπάρχει μέσα:

```
>>> ===== RESTART =====
>>>
42
Help on function test in module __main__:

test()
  Print an answer

>>>
```

► Το `functionbasics.py` είναι ένα αρχείο `.py`, το οποίο περιέχει δύο συναρτήσεις. Προφανώς δεν μπορούμε να το "τρέξουμε" γιατί οι συναρτήσεις πρέπει να κληθούν για να κάνουν κάποια λειτουργία:

```
# functionbasics.py περιέχει 2 συναρτήσεις:  
  
def maxi(x,y):  
    if (x < y):  
        return y  
    else:  
        return x  
  
def mini(x,y):  
    if (x > y):  
        return y  
    else:  
        return x
```

Την `maxi` μπορούμε να την καλέσουμε με δύο τρόπους από ένα άλλο αρχείο `.py`.

1. Κάνοντας `import` μόνο την `maxi` από το αρχείο `functionbasics`, οπότε μπορούμε να την καλέσουμε σαν να είναι τοπική συνάρτηση:

```
#import the function maxi  
from functionbasics import maxi  
  
print(maxi(3,21))
```

ή

2. Κάνοντας `import` όλη την `functionbasics` και μετά χρησιμοποιώντας ολόκληρο το όνομα:

```
import functionbasics  
  
print(functionbasics.maxi(3,21))
```

► Γεννήτρια ψευδοτυχαίων αριθμών και επιλογής:

```
import random

i=random.random()           # Random float x, 0.0 <= x < 1.0
print(i)

f=random.uniform(1, 10)     # Random float x, 1.0 <= x < 10.0
print(f)

ir=random.randrange(10)     # Integer from 0 to 9
print(ir)

ei=random.randrange(0, 101, 2) # Even integer from 0 to 100
print(ei)

c=random.choice('abcdefghij') # Single random element
print(c)

items = [1, 2, 3, 4, 5, 6, 7]
print(items)
random.shuffle(items)
print(items)

sam=random.sample([1, 2, 3, 4, 5], 3) # Three samples without replacement
print(sam)
```