ΣΥΝΑΡΤΗΣΕΙΣ

In [1]:
```
import random
students = [
    'Tzwrtzina',
    'Andreas',
    'Xristos',
    'Andromaxh',
    'Danah',
    'Antwnia',
    'Aris',
    'Maria',
    'Sofia',
    'Iwanna',
    'Aggelos',
]

def random_student():
    return random.choice(students)

rs = random_student
```

In [2]:
```
def foo(a,b):
    return (a+b)/2
```

In [3]:
```
rs()
```

Out[3]:
```
'Xristos'
```

In [13]:
```
foo(5,7)
```

Out[13]:
```
6.0
```

In [14]:
```
a = foo(5,7)
print (a)
```

```
6.0
```

In [15]:
```
a
```

Out[15]:
```
6.0
```

In [ ]:
```

```

In [16]:
```
def goo(a,b):
    print ((a+b)/2)
```

In [18]:
```
a = goo(5,7)
print (a)
```

```
6.0
None
```

In [20]:
```
a is None
```

Out[20]:
```
True
```

In [ ]:

In [ ]:

In [17]:
```
rs()
```

Out[17]:
```
'Tzwrtzina'
```

In [ ]:

In [7]:
```
goo(5,7)
```

```
6.0
```

In [10]:
```
foo(5,7) + 6
```

Out[10]:
```
12.0
```

In [12]:
```
goo(5,7) + 6
```

```
6.0

---------------------------------------------------------------------
------------
TypeError                                 Traceback (most recen
t call last)
<ipython-input-12-bbc21bb61727> in <module>()
----> 1 goo(5,7) + 6

TypeError: unsupported operand type(s) for +: 'NoneType' and 'i
nt'
```

In [21]:
```
def foo(a,b):

    c = a+10
    return c+b
```

In [23]:
```
foo(1,9)
```

Out[23]:
```
20
```

In [ ]:

In [37]:
```
rs()
```

Out[37]:
```
'Xristos'
```

Όταν μία συνάρτηση δεν έχει return, τότε κάνει return None!

In [28]:
```
def prime(n):
    for i in range(2,n):
        if n%i == 0:
            return False
    #return True
```

In [26]:
```
prime(15)
```

Out[26]:
```
False
```

In [29]:
```
prime(20)
```

Out[29]:
```
False
```

In [33]:
```
aa = prime(23)
```

In [34]:
```
print(aa)
```

```
None
```

In [35]:
```
print(prime(23))
```

```
None
```

In [36]:
```
print (prime(20))
```

```
False
```

In [45]:
```
def foo():
    return 4
```

In [47]:
```
foo(3)
```

```
---------------------------------------------------------------------
------------
TypeError                                 Traceback (most recen
t call last)
<ipython-input-47-bb81d0344e3d> in <module>()
----> 1 foo(3)

TypeError: foo() takes 0 positional arguments but 1 was given
```

In [ ]:
```

```

In [40]:
```
foo()
```

Out[40]:
```
4
```

In [41]:
```
def f(a, b=4):
    return a+b
```

In [43]:
```
f(10)
```

Out[43]:
```
14
```

In [48]:
```
f(10, 3)
```

Out[48]:
```
13
```

In [49]:
```
f(10,3,4)
```

```
---------------------------------------------------------------------
------------
TypeError                                 Traceback (most recen
t call last)
<ipython-input-49-76b5f97020f9> in <module>()
----> 1 f(10,3,4)

TypeError: f() takes from 1 to 2 positional arguments but 3 wer
e given
```

In [50]:
```
f(10)
```

Out[50]:
```
14
```

In [51]:
```
def f(a, b):
    return a+b
```

In [52]:
```
f(1)
```

```
---------------------------------------------------------------------
------------
TypeError                                 Traceback (most recen
t call last)
<ipython-input-52-281ab0a37d7d> in <module>()
----> 1 f(1)

TypeError: f() missing 1 required positional argument: 'b'
```

In [70]:
```
import random


random.random()
```

Out[70]:
```
0.23432783893358866
```

In [71]:
```
rs()
```

Out[71]:
```
'Sofia'
```

In [94]:
```
def f(a):
    print ('-->', a)

    a=5
    print ('-->', a)
    return a


a = 2
b = f(a)
print ('==>', a)
print ('==>', b)
```

```
--> 2
--> 5
==> 2
==> 5
```

In [96]:
```
def g(a):
    a.append(5)

a = [2]
g(a)
print (a)
```

```
[2, 5]
```

In [103]:
```
def g(b):
    b.append(5)
    #return None

a = [2]
g(a)
print (a)
```

```
[2, 5]
```

In [106]:
```
def g(b):
    b.append(5)
    #return None

a = [2]
αυτό_που_επέστρεψε_η_g = g(a)
print (αυτό_που_επέστρεψε_η_g)
```

```
None
```

In [107]:
```
def f(a,b):
    print (a)
    print (b)


x=1
y=2
#f(a,b)

f(x,y)
```

```
1
2
```

In [110]:
```
def f(n):

    return n//2, n%2


a,b = f(10)
print(a)
print(b)
```

```
5
0
```

In [111]:
```
print (f(10))
```

```
(5, 0)
```

In [158]:
```
def f(n):

    return n//2, n%2


a = f(10)

print (a)
```

```
(5, 0)
```

Variable unpacking

In [157]:
```
def f(n):

    return n//2, n%2


a,b = f(10)

print (a)
```

```
5
```

In [150]:
```
def f(n):

    return n//2, n%2
```

In [151]:
```
f(10)
```

Out[151]:
```
(5, 0)
```

In [152]:
```
c = f(10)
print (c)
```

```
(5, 0)
```

In [153]:
```
a,b = (5,0)
print (a)
print (b)
```

```
5
0
```

In [154]:
```
c = (5,0)
a,b = c
print (a)
print (b)
```

```
5
0
```

In [155]:
```
c = f(10)
a,b = c
print (a)
print (b)
```

```
5
0
```

In [ ]:
```
a,b = f(10)
```

In [143]:
```

```

In [144]:
```
type(f(10))
```

Out[144]:
```
tuple
```

In [145]:
```
a,b = (1,10)
```

In [146]:
```
c = f(10)
a,b = c
```

In [147]:
```
a,b = (1,10)
```

In [148]:
```
a,a = (1,10)
print (a)
```

```
10
```

In [149]:
```
a=1
a=10
print (a)
```

```
10
```

In [ ]:
```

```

In [130]:
```
a
```

Out[130]:
```
(5, 0)
```

In [131]:
```
len(a)
```

Out[131]: 2

In [132]:
```
a=1
b=5
```

In [133]:
```
a,b = (1,5)
```

In [134]:
```
print (a)
print (b)
```

```
1
5
```

In [135]:
```
a = (1,2)
```

In [136]:
```
list(a)
```

Out[136]: [1, 2]

In [137]:
```
a= 'mitsos'
```

In [139]:
```
a_l = list(a)
```

In [140]:
```
a_l = [a]
```

In [141]:
```
a_l
```

Out[141]: ['mitsos']

In [114]:
```
a
```

Out[114]: (5, 0)

In [115]:
```
type(a)
```

Out[115]: tuple

In [116]:
```
a = [1,2]
a += [4]
a
```

Out[116]: [1, 2, 4]

In [121]:
```
a = (1,2)
```

In [122]:
```
[1,2] + [5,6]
```

Out[122]:
```
[1, 2, 5, 6]
```

In [123]:
```
(1,2) + (5,6)
```

Out[123]:
```
(1, 2, 5, 6)
```

In [126]:
```
(1,2).append
```

```
---------------------------------------------------------------------
------------
AttributeError                            Traceback (most recen
t call last)
<ipython-input-126-92a0c13399be> in <module>()
----> 1 (1,2).append

AttributeError: 'tuple' object has no attribute 'append'
```

In [ ]:
```

```

In [124]:
```
a = [1,2]
a[0] = 5
```

In [125]:
```
a = (1,2)
a[0] = 5
```

```
---------------------------------------------------------------------
------------
TypeError                                 Traceback (most recen
t call last)
<ipython-input-125-eb7ab375b278> in <module>()
      1 a = (1,2)
----> 2 a[0] = 5

TypeError: 'tuple' object does not support item assignment
```

In [159]:
```
def f(n):
    return n/2
```

In [160]:
```
f(9)
```

Out[160]:
```
4.5
```

In [161]:
```
a=9
f(a)
```

Out[161]:
```
4.5
```

In [162]:
```
n=9
f(n)
```

Out[162]: 4.5

In [163]:
```
def f(a):
    return a.upper()
```

In [164]:
```
f('mitsos')
```

Out[164]: 'MITSOS'

In [166]:
```
a = 'mitsos'
f(a)
```

Out[166]: 'MITSOS'

In [167]:
```
def f(a,b):
    return a*b
```

In [168]:
```
f(3,5)
```

Out[168]: 15

In [169]:
```
f('hello ', 3)
```

Out[169]: 'hello hello hello '

In [178]:
```
f([5,6,7], 3)
```

Out[178]: [5, 6, 7, 5, 6, 7, 5, 6, 7]

In [181]:
```
f(3+5j, 2+3j)
```

Out[181]: (-9+19j)

In [180]:
```
3+5j
```

Out[180]: (3+5j)

In [182]:
```
a= [1,2,3]
```

In [184]:
```
[1,2,3] * [5,6,7]
```

```
---------------------------------------------------------------------
------------
TypeError                                 Traceback (most recen
t call last)
<ipython-input-184-4f0211182fde> in <module>()
----> 1 [1,2,3] * [5,6,7]

TypeError: can't multiply sequence by non-int of type 'list'
```

In [185]:
```
f([1,2,3], [5,6,7])
```

```
---------------------------------------------------------------------
------------
TypeError                                 Traceback (most recen
t call last)
<ipython-input-185-2cdf4f97e85e> in <module>()
----> 1 f([1,2,3], [5,6,7])

<ipython-input-167-e1a731d4f4e3> in f(a, b)
      1 def f(a,b):
----> 2     return a*b

TypeError: can't multiply sequence by non-int of type 'list'
```

In [176]:
```
import numpy as np
```

In [177]:
```
np.array(a) * 5
```

Out[177]:
```
array([ 5, 10, 15])
```

In [187]:
```
def f(x):
    def g(y):
        return y+2

    return x + g(x)

f(1)
```

Out[187]:
```
4
```

In [191]:
```
def f(x):

    def g(y):
        return y+x

    return g(x+2)

f(1)
```

```
3 1
```

Out[191]: 4

In [188]:
```
rs()
```

Out[188]: 'Iwanna'

ΑΝΑΔΡΟΜΗ

N! = N *(N-1)! N!* = *N* (N-1) * (N-2)! ....

In [221]:
```
def fact(N):
    print ('I HAVE TO COMPUTE FACTORIAL FOR N=', N)

    if N==1:
        #print ('I am not calling fact! N=1')
        return 1

    #print ('Calling fact with argument ', N-1)
    #print ('Computing the factorial for:', N-1)
    return N * fact(N-1)
```

In [220]:
```
fact(2)
```

```
I HAVE TO COMPUTE FACTORIA FOR N= 2
I HAVE TO COMPUTE FACTORIA FOR N= 1
```

Out[220]: 2

fact(N) N *fact(N-1) N* N-1 *fact(N-2) N* N-1 *N-2* fact(N-3) N *N-1* N-2 *N-3* 1

In [224]:
```
def f():
    return 3
    # οτιδίποτε είναι κάτω από return αγνοείται!!!!
    print ('Hello')
f()
```

Out[224]: 3

In [222]:
```
rs()
```

Out[222]:
```
'Andreas'
```

In [225]:
```
a=3
type(a)
```

Out[225]:
```
int
```

In [226]:
```
a='mitsos'
type(a)
```

Out[226]:
```
str
```

In [227]:
```
a=True
type(a)
```

Out[227]:
```
bool
```

In [228]:
```
a=[1,2,3]
type(a)
```

Out[228]:
```
list
```

In [229]:
```
a=(1,2,3)
type(a)
```

Out[229]:
```
tuple
```

In [230]:
```
def a(x):
    return x+1
type(a)
```

Out[230]:
```
function
```

In [231]:
```
a
```

Out[231]:
```
<function __main__.a(x)>
```

In [234]:
```
def f1(x):
    return x+1

def f2(x):
    return x+2

def g(a, t):
    return a + t(a)
```

In [235]:
```
g(5, f1)
```

Out[235]:
```
11
```

In [ ]:
```
g(5, f2)
```

In [238]:
```
def f(a,b,*c):
    print (a)
    print (b)
    print (c)
```

In [239]:
```
f(1,2,3,4,5,6,7,8,9)
```

```
1
2
(3, 4, 5, 6, 7, 8, 9)
```

In [240]:
```
f(1,2,3,4)
```

```
1
2
(3, 4)
```

In [241]:
```
f(1,2,3)
```

```
1
2
(3,)
```

In [242]:
```
a=[3]
```

In [244]:
```
a=(1,)
print (a)
```

```
(1,)
```

In [246]:
```
a=[3,]
print (a)
a=[3]
print (a)
```

```
[3]
[3]
```

Σε lists, tuples, ... μπορείς να βάλεις κόμμα στο τέλος

In [247]:
```
a=[1,2,3]
print (a)

a=[1,2,3,]
print (a)
```

```
[1, 2, 3]
[1, 2, 3]
```

Φτιάξτε μία συνάρτηση η οποία θα παίρνει δύο παράμετρους. Θα επιστρέφει το άθροισμά τους. Η συνάρτηση πρέπει να ελέγχει αν η είσοδος είναι int ή float. Αν η είσοδος δεν είναι int ή float θα τυπώνει ένα μήνυμα λάθους.

In [258]:
```
def f(a,b):
    if not type(a) is int and not type(a) is float:
        return ('ERROR')

    if not type(b) is int and not type(b) is float:
        return ('ERROR')

    return a+b
```

In [267]:
```
def f(a,b):
    if not type(a) is int and not type(a) is float:
        return ('ERROR')

    elif not type(b) is int and not type(b) is float:
        return ('ERROR')

    return a+b
```

In [268]:
```
def f(a,b):
    if type(a) is int or type(a) is float:
        if type(b) is int or type(b) is float:
            return a+b

    return 'Error!'
```

In [274]:
```
def f(a,b):
    if not type(a) in [int, float]:
        return ('ERROR')

    if not type(b) in [int, float]:
        return ('ERROR')

    return a+b
```

In [276]:
```
def f(a,b):
    if type(a) not in [int, float]:
        return ('ERROR')

    if type(b) not in [int, float]:
        return ('ERROR')

    return a+b
```

In [277]:
```
def f(a,b):

    for c in [a,b]:
        if type(c) not in [int, float]:
            return ('ERROR')

    return a+b
```

In [272]:
```
f(5,6)
```

Out[272]:
```
11
```

In [273]:
```
f([5,3], 5)
```

Out[273]:
```
'ERROR'
```

In [237]:
```
rs()
```

Out[237]:
```
'Andromaxh'
```

In [ ]:
```

```