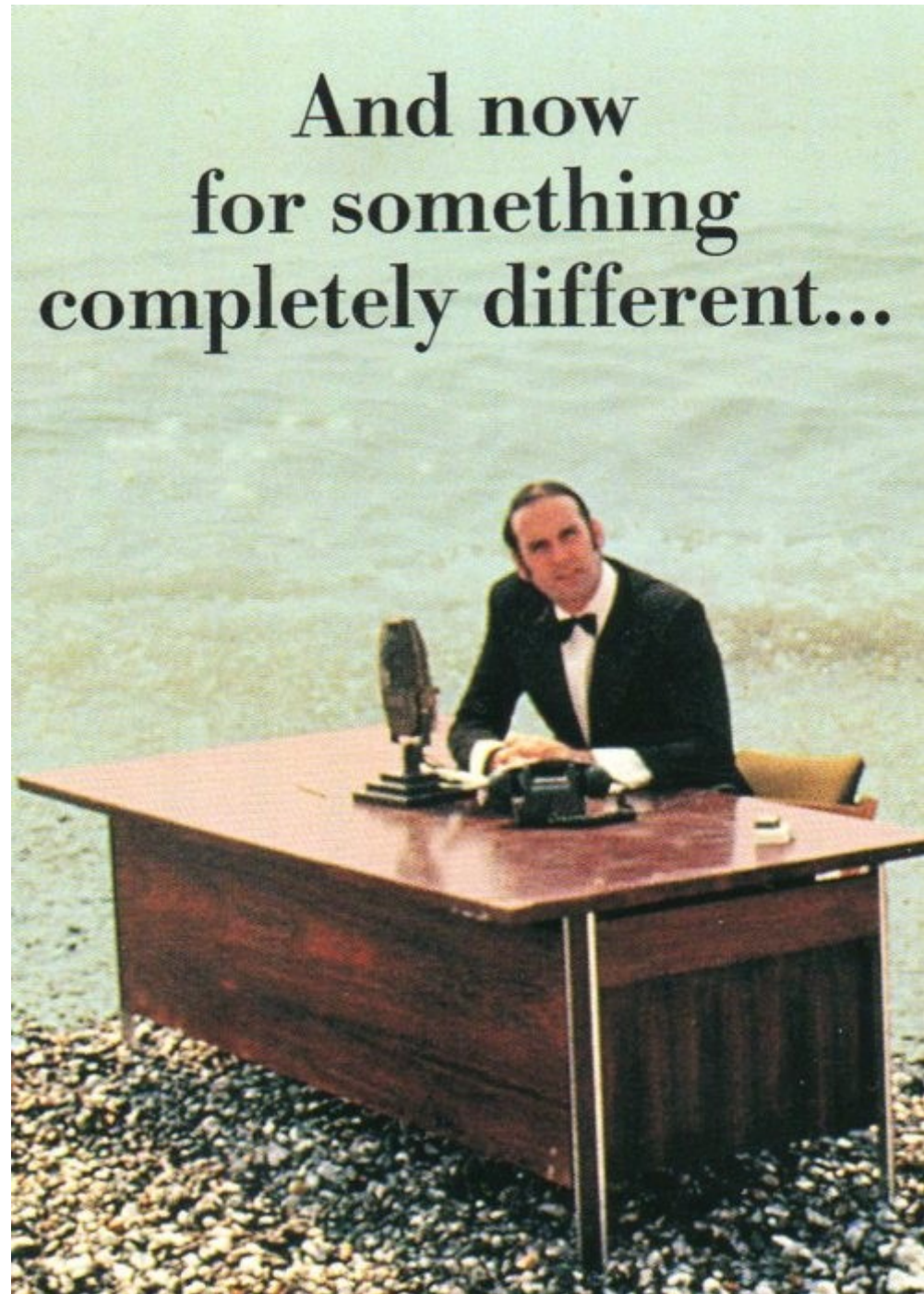


Εισαγωγή στον
Προγραμματισμό
με την
Python 3

Βασίλης Βασιλάκης
vassilakis1978@gmail.com

Γιώργος Μπουκέας
boukeas@gmail.com

Απρίλιος 2015



Γιατί Python; (στην εκπαίδευση)



Προγραμματιστικά Χαρακτηριστικά

Συντακτική Απλότητα

Υποχρεωτική Στοίχιση

Δομές Δεδομένων

Συνδυασμός

Προγραμματιστικών

Παραδειγμάτων

Δομή Παρουσίασης

Βασικά Διαδικαστικά Χαρακτηριστικά

Μεταβλητές & Εκφράσεις
Δομή Επιλογής & Επανάληψης
Εξάσκηση »

Τμηματικός Προγραμματισμός

Συναρτήσεις
Βιβλιοθήκες
Εξάσκηση »

Δομές Δεδομένων

Συμβολοσειρές (strings) *
Λίστες
Εξάσκηση »
Πλειάδες (tuples)
Λεξικά (dicts)
Αρχεία

Python Interpreter + IDE



Online Python IDE

<http://repl.it/languages/Python3>

The screenshot shows the repl.it Python3 online IDE interface. The browser window title is "repl.it - Python3 - Mozilla Firefox". The address bar contains "repl.it - Python3". The main interface features the repl.it logo, "Python3" text, and navigation links for "languages" and "examples", along with a "Log In" button. Below the navigation bar are three buttons: "share", "save", and "run". The code editor area is currently empty, with a line number "1" visible. To the right of the code editor is a terminal window with a "clear" button. The terminal output shows the Python version and environment: "Python 3.4.0 (default, Apr 11 2014, 13:05:11) [GCC 4.8.2] on linux". At the bottom of the page, there are links for "about us", "API", "about Python3", and "help".

ανάπτυξη προγραμμάτων
(πατήστε run για εκτέλεση)

γραμμή εντολών
(διαλογική αλληλεπίδραση)

about us API about Python3 help

Μεταβλητές

int
float
bool
str

```
>>> type(3.14159)  
<class 'float'>  
>>> type(True)  
<class 'bool'>
```

Χωρίς δήλωση τύπου

```
num = 13  
title = "Python seminar"
```

```
>>> type(title)  
<class 'str'>
```

Πολλαπλή εκχώρηση

```
a = b = c = 0  
name, age = "Άλαν", 13
```

Βασικοί Τελεστές

Αριθμητικοί

+ - * / ** // %

```
>>> 120 // 13
```

```
9
```

```
>>> 13 ** 3
```

```
2197
```

Συγκριτικοί

> >= < <= == !=

Λογικοί

and or not

Είσοδος Έξοδος

συναρτήσεις
print()
input()

```
print("Καλημέρα.")  
Καλημέρα.
```

```
print("1, 2", end=" ", "  
print("3, 4")  
1, 2, 3, 4
```

```
name = input("Όνομα; ")  
Όνομα; Άλαν  
print("Καλημέρα", name)  
Καλημέρα Άλαν
```


Μετατροπή Τύπου (casting)

```
>>> num = input("X: ")  
X: 13  
>>> num ** 2  
TypeError: ...
```

```
>>> num = int(input("X: "))  
X: 13  
>>> num ** 2  
169
```

Σχόλια

σχόλιο μιας γραμμής

"""

σχόλιο
πολλαπλών
γραμμών

"""

Δομή Επιλογής

η **στοίχιση** δεν
είναι προαιρετική
εσοχή =
εμφώλευση

μην παραλείπετε το
σύμβολο :

```
if συνθήκη:  
    εντολές
```

```
if συνθήκη:  
    εντολές  
else:  
    εντολές
```

```
if συνθήκη:  
    εντολές  
elif συνθήκη:  
    εντολές  
...  
else:  
    εντολές
```

Δομή Επιλογής

```
if x > max:  
    max = x
```

```
if x > y:  
    max = x  
else:  
    max = y  
print(max)
```

```
if x > max:  
    max2 = max  
    max = x  
elif x > max2:  
    max2 = x
```

Δομή Επανάληψης

while
for

η **for** διατρέχει
τα στοιχεία
μιας ακολουθίας
(π.χ. μιας συμβολοσειράς
ή μιας λίστας)

while συνθήκη:
εντολές

for μεταβλητή **in** ακολουθία:
εντολές

Δομή Επανάληψης

μια επανάληψη μπορεί
να διακοπεί με την `break`

```
num = int(input("X: "))  
while num != 0:  
    print(1/num)  
    num = int(input("X: "))
```

```
while True:  
    num = int(input("X: "))  
    if num == 0:  
        break  
    print(1/num)
```

Δομή Επανάληψης

συχνά η `for`
χρησιμοποιείται σε
συνδυασμό με την
ακολουθία `range`
`range(a,b,step)`

το `b` δεν
συμπεριλαμβάνεται
στην ακολουθία

```
for i in range(10):  
    print(i, end=" ")
```

0 1 2 3 4 5 6 7 8 9

```
for i in range(5,50,10):  
    print(i, end=" ")
```

5 15 25 35 45

```
for c in "Καλημέρα"  
    print(c, end="-")
```

Κ-α-λ-η-μ-έ-ρ-α-

Εξάσκηση

Εικασία Collatz

Υλοποιήστε τη διαδικασία που περιγράφεται.

Διαβάστε από το χρήστη το αρχικό x και εμφανίστε την σειρά των αριθμών που προκύπτει.

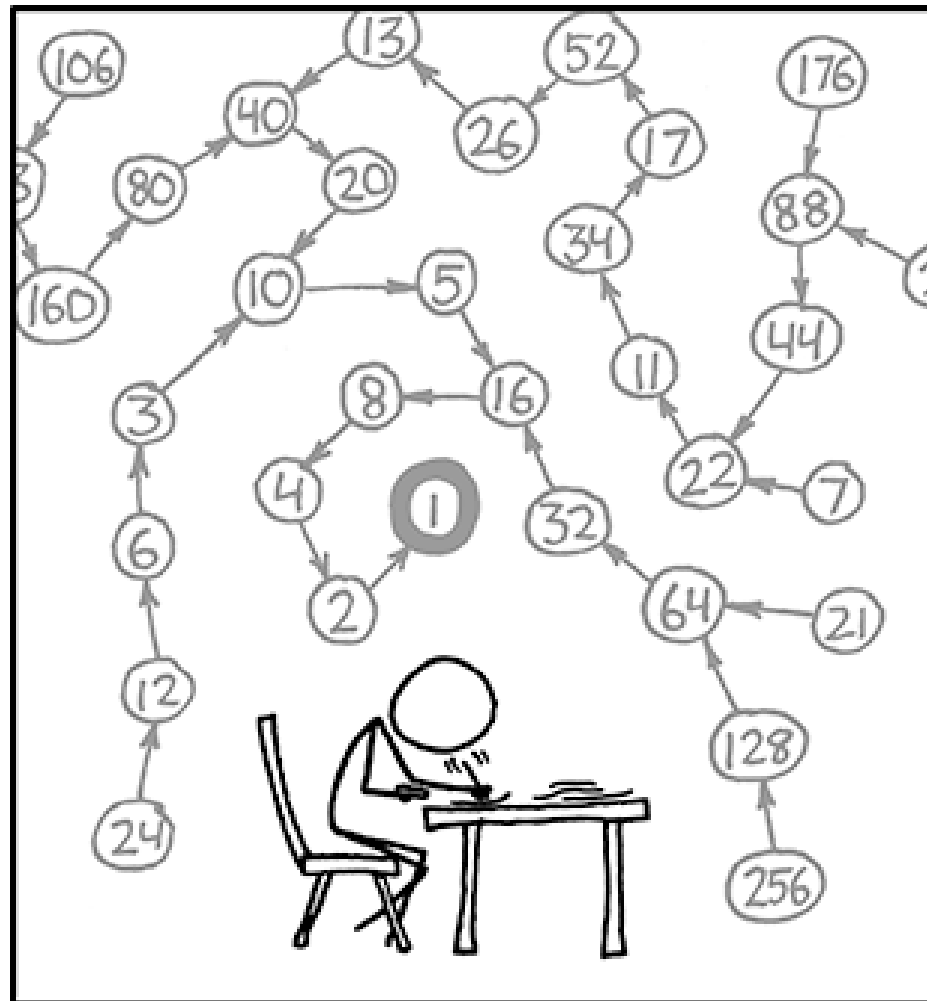
Επιλέξτε έναν θετικό ακέραιο x .

Αν είναι άρτιος διαιρέστε τον με το 2, ενώ αν είναι περιττός πολλαπλασιάστε τον με το 3 και προσθέστε άλλη μια μονάδα.

Επαναλάβετε τη διαδικασία με τον αριθμό που θα προκύψει.

Από οποιονδήποτε αριθμό x κι αν ξεκινήσετε, θα καταλήξετε τελικά στο 1.

Εξάσκηση Εικασία Collatz



THE COLLATZ CONJECTURE STATES THAT IF YOU PICK A NUMBER, AND IF IT'S EVEN DIVIDE IT BY TWO AND IF IT'S ODD MULTIPLY IT BY THREE AND ADD ONE, AND YOU REPEAT THIS PROCEDURE LONG ENOUGH, EVENTUALLY YOUR FRIENDS WILL STOP CALLING TO SEE IF YOU WANT TO HANG OUT.

Εξάσκηση

Εικασία Collatz

Λύση

```
x = int(input("X: "))
while x > 1:
    if x % 2:
        x = 3*x + 1
    else:
        x = x // 2
    print(x, end=" ")
print()
```

```
>>>
```

```
X: 13
```

```
40 20 10 5 16 8 4 2 1
```

Συναρτήσεις

Μια συνάρτηση πρέπει να έχει οριστεί πριν χρησιμοποιηθεί.

Τα αποτελέσματα επιστρέφονται με την **return**

```
def min(a,b):  
    if a < b:  
        return a  
    else:  
        return b
```

```
def minmax(a,b):  
    if a < b:  
        return a,b  
    else:  
        return b,a
```

Παραδείγματα κλήσης:

```
print(min(y,1))  
x, y = minmax(13,27)
```

Βιβλιοθήκες (modules)

```
import random  
dice = random.randint(1,6)  
import time  
time.sleep(60)
```

Εξάσκηση Εικασία Collatz με συναρτήσεις

Κατασκευάστε συνάρτηση που να ελέγχει αν ένας ακέραιος είναι περιττός.

Κατασκευάστε συνάρτηση που να δέχεται σαν παράμετρο έναν ακέραιο και να επιστρέφει τον επόμενο αριθμό που προκύπτει από τη διαδικασία του Collatz.

Χρησιμοποιήστε αυτές τις συναρτήσεις στον κώδικα που γράψατε προηγουμένως.

Επιλέξτε ένα τυχαίο σημείο εκκίνησης μεταξύ 2 και 32.

Εξάσκηση

Εικασία Collatz με συναρτήσεις

Λύση

```
import random
def isOdd(x):
    return x % 2 == 1
def next(x):
    if isOdd(x):
        return 3*x + 1
    else:
        return x//2
x = random.randint(2,32)
print(x, end=" ")
while x > 1:
    x = next(x)
    print(x, end=" ")
print()
```

Δομές Δεδομένων

Συμβολοσειρές
(Strings)

Λίστες
(Lists)

Πλειάδες
(Tuples)

Λεξικά
(Dicts)

ακολουθιακές
δομές



Συμβολοσειρές (strings)

immutable

```
>>> word = "Python"
```

0	1	2	3	4	5
P	y	t	h	o	n
-6	-5	-4	-3	-2	-1

```
>>> word[0]
```

```
'P'
```

```
>>> word[-1]
```

```
'n'
```

```
>>> word[0] = 'p'
```

```
TypeError: ...
```


Συμβολοσειρές (strings)

0	1	2	3	4	5
P	y	t	h	o	n
-6	-5	-4	-3	-2	-1

slicing
[a:b:step]

επιστρέφει *τμήμα*
ενός string

το b δεν
συμπεριλαμβάνεται
στο τμήμα

```
>>> word[1:4]
```

```
'yth'
```

```
>>> word[1:]
```

```
'ython'
```

```
>>> word[:4]
```

```
'Pyth'
```

```
>>> word[::2]
```

```
'Pto'
```

```
>>> word = 'p' + word[1:]
```

Συμβολοσειρές (strings)

Τελεστές + * in
Συνάρτηση len()

```
>>> s = "Monty " + "Python's"  
>>> print(s)  
Monty Python's
```

```
>>> "a" * 10  
aaaaaaaaaa
```

```
>>> "Python" in s  
True  
>>> "Circus" in s  
False
```

```
>>> len(s)  
14
```

Συμβολοσειρές (strings)

Διάσχιση

```
word = "Python"  
for c in word:  
    print(2*c, end="")  
PPyytthhoonn
```

```
word = "Python"  
for i in range(len(word)):  
    print(2*word[i], end="")  
PPyytthhoonn
```

Συμβολοσειρές (strings)

Επισκόπηση

Τελεστές

+ * in [a:b:step]

Συναρτήσεις

len()

Μέθοδοι

str.count()

str.replace()

str.split()

Διάσχιση

for χαρακτήρας in
συμβολοσειρά:

Λίστες

mutable

```
>>> c = ['red', 'green', 'blue', 'yellow']
```

0	1	2	3
red	green	blue	yellow
-4	-3	-2	-1

```
>>> c[0] = 'orange'
```

```
>>> print(c)
```

```
['orange', 'green', 'blue', 'yellow']
```

Λίστες

ομοιότητες με τις
συμβολοσειρές
τελεστές
+ * in [a:b:step]
συναρτήσεις
len()

```
>>> len(c)
4
```

```
>>> "brown" in c
False
```

```
>>> c[1:3]
['green', 'blue']
```

```
>>> ['brown', 'purple'] + c
['brown', 'purple', 'orange',
'green', 'blue', 'yellow']
```

```
>>> 2 * c
['orange', 'green', 'blue', 'yellow',
'orange', 'green', 'blue', 'yellow']
```

Λίστες

μέθοδοι
list.append()
list.sort()
διάσχιση
for

```
>>> fruits = ['banana', 'apple']
```

```
>>> fruits.append('orange')  
>>> print(fruits)  
['banana', 'apple', 'orange']
```

```
>>> fruits.sort()  
>>> print(fruits)  
['apple', 'banana', 'orange']
```

```
for fruit in fruits:  
    print(fruit.upper(), end=" ")  
APPLE BANANA ORANGE
```

Λίστες

μετατροπή
συμβολοσειράς σε
λίστα και
αντιστρόφως

```
>>> word = "Python"  
>>> list(word)  
['P', 'y', 't', 'h', 'o', 'n']
```

```
>>> words = ["Monty", "Python's", \  
             "Flying", "Circus"]  
>>> " ".join(words)  
"Monty Python's Flying Circus"
```


Λίστες

Συλλογή σε λίστα
(list comprehension)

Είναι σύνηθες να συλλέγουμε σε μια λίστα τιμές που προέρχονται από μια ακολουθία και ικανοποιούν μια συγκεκριμένη συνθήκη.

```
list = []  
for μεταβλητή in ακολουθία:  
    if συνθήκη:  
        list.append(έκφραση)
```

Στην Python προτείνεται το ιδίωμα:

```
[ έκφραση for στοιχείο in ακολουθία if συνθήκη ]
```

Λίστες

Συλλογή σε λίστα
(list comprehension)

Τα τετράγωνα των μονών αριθμών
από το 1 μέχρι και το 11.

```
sq = [x**2 for x in range(1,13,2)]
```

αντί για

```
sq = []  
for x in range(1,13,2):  
    sq.append(x**2)
```

Λίστες

Συλλογή σε λίστα
(list comprehension)

Το πρώτο γράμμα κάθε λέξης στη λίστα L που έχει περισσότερα από 4 γράμματα.

```
C = [wd[0] for wd in L if len(wd) > 4]
```

αντί για

```
C = []  
for wd in L:  
    if len(wd) > 4:  
        C.append(wd[0])
```

Λίστες

Επισκόπηση

Τελεστές

+ * in [a:b:step]

Συναρτήσεις

len(), sorted()

Μέθοδοι

list.append() list.reverse()

list.extend() list.sort()

list.insert()

Διάσχιση

for στοιχείο in λίστα:

Συλλογή σε λίστα

[έκφραση

for στοιχείο in ακολουθία

if συνθήκη]

Εξάσκηση Οι πλανήτες

*Κατασκευάστε μια λίστα με τα ονόματα των **8 πλανητών**.*

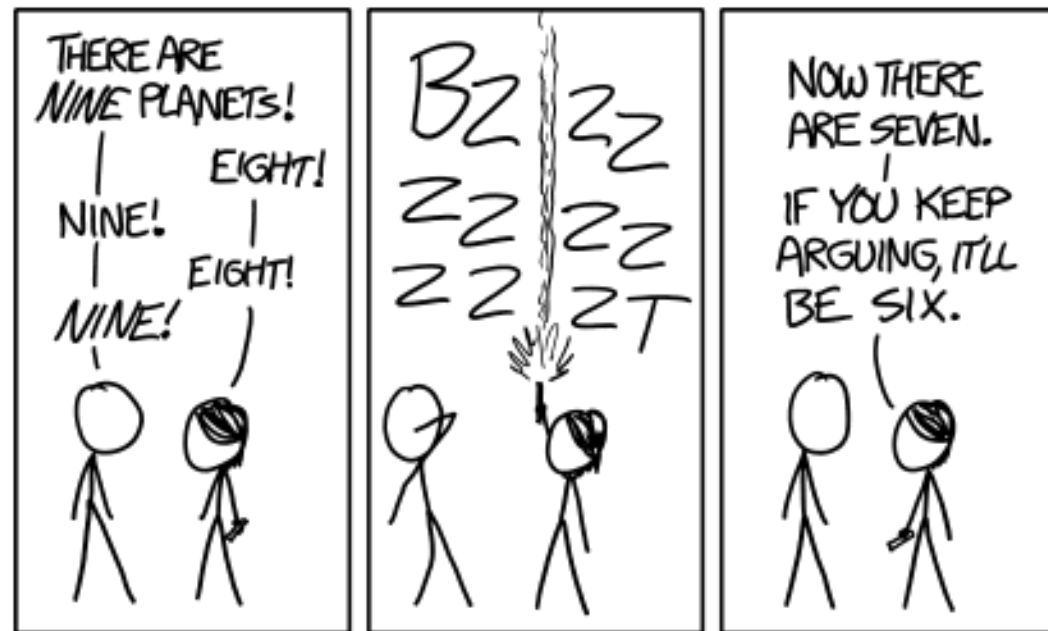
Σκονάκι: Ερμής, Αφροδίτη, Γη, Άρης, Δίας, Κρόνος, Ουρανός, Ποσειδώνας.

Διαβάστε από το χρήστη έναν ακέραιο (1-8) και εμφανίστε το όνομα του αντίστοιχου πλανήτη, τον προηγούμενό του (αν υπάρχει) και τον επόμενο του (αν υπάρχει).

Κατασκευάστε μια λίστα με τα ονόματα των πλανητών σε αντίστροφη σειρά.

Κατασκευάστε μια λίστα με τα ονόματα των πλανητών που τελειώνουν σε 'ς'.

Εξάσκηση Οι πλανήτες



Εξάσκηση

Οι πλανήτες

Λύση

```
planets = ["Ερμής", ... ]  
# είσοδος: αριθμός του πλανήτη  
pos = int(input("Πλανήτη; "))  
# όνομα πλανήτη  
print(planets[pos-1])  
# όνομα προηγούμενου  
if pos > 1:  
    print("Πριν:", planets[pos-2])  
# όνομα επόμενου  
if pos < len(planets):  
    print("Μετά:", planets[pos])  
# λίστα με αντίστροφη σειρά  
planets[::-1]  
# λίστα "αρσενικών" πλανητών  
[p for p in planets if p[-1] == 'ς']
```

Πλειάδες (Tuples)

immutable

```
>>> days = ('Δευ', 'Τρι', 'Τετ', \
            'Πεμ', 'Παρ', 'Σαβ', 'Κυρ')
```

```
>>> days[0]
'Δευ'
```

```
>>> days[0] = 'Κυρ'
TypeError: ...
```


Λεξικά (Dicts)

πίνακες
κατατερματισμού
(hashtables)
πίνακες συσχετισμού
(associative arrays)

mutable

ζεύγη
κλειδιών: τιμών

```
dict = { key1:val1, key2:val2, ... }
```

```
>>> p = { 'Βασίλης': '6947654321', \  
         'Γιώργος': '6938765432' }
```

```
>>> p['Γιώργος']  
'6938741111'
```

```
>>> p['Γιώργος'] = '6938234567'
```

```
>>> p['Σπύρος'] = '6976543210'
```

Αρχεία (κειμένου)

Άνοιγμα

```
myfile = open("test.txt", "r+", \
              encoding="utf-8")
```

Ανάγνωση

```
for line in myfile:  
    lines.append(line)
```

```
lines = myfile.readlines()
```

Εγγραφή

```
myfile.write("Τηλέφωνα")
```

Κλείσιμο

```
myfile.close()
```

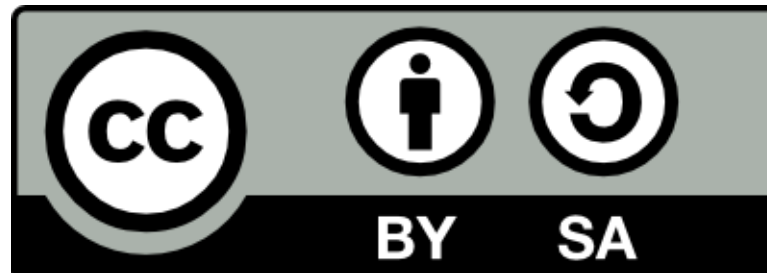
Ευχαριστούμε!

Άδεια Χρήσης

Το υλικό διατίθεται με άδεια **CC-BY-SA 4.0**
Creative Commons

Αναφορά Δημιουργού - Παρόμοια Διανομή 4.0
(<http://creativecommons.org/licenses/by-sa/4.0/>).

Η αναφορά θα πρέπει να γίνεται ως εξής:
«*Εισαγωγή στον Προγραμματισμό με Python*».
Βασίλης Βασιλάκης, Γιώργος Μπουκέας, 2015.



Η εισαγωγική εικόνα είναι αναφορά στους
Monty Pythons, από τους οποίους πήρε το
όνομά της η γλώσσα.

Τα σκίτσα προέρχονται από το xkcd.com