



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά  
μαθήματα ΠΠ

# Εισαγωγή

## στους Η/Υ και τις Εφαρμογές

Ενότητα 5: Επεξεργασία δεδομένων με τη γλώσσα  
προγραμματισμού python

Μανώλης Τζαγκαράκης, Βικτωρία Δασκάλου  
Σχολή Οργάνωσης και Διοίκησης Επιχειρήσεων  
Τμήμα Οικονομικών Επιστημών

# Σκοποί ενότητας

- Να πραγματοποιηθεί μία εισαγωγή στις δυνατότητες της γλώσσας `python`
- Να παρουσιαστούν οι βασικές δυνατότητες των γλωσσών προγραμματισμού για την επεξεργασία δεδομένων



# Περιεχόμενα ενότητας

1. Χαρακτηριστικά της γλώσσας `python`
2. Προγραμματισμός με `python`
  - Υπολογισμοί και μεταβλητές
  - Συμβολοσειρές
  - Έλεγχος ροής
  - Λίστες
  - Αρχεία
  - Συναρτήσεις
  - Γραφικές παραστάσεις



Η γλώσσα python

# Η γλώσσα python

- Σύντομη ιστορία
  - Δημιουργήθηκε από τον Ολλανδό Guido van Rossum το 1989 (της έδωσε το όνομα των Monty Python)
  - 2 εκδόσεις:
    - Python 2.0 (Οκτώβρης 2000) με τελευταίες εκδόσεις 2.6 & 2.7
    - Python 3.0 (Δεκέμβριος 2008): ιστορικά η πρώτη γλώσσα προγραμματισμού που σπάει την προς τα πίσω συμβατότητα

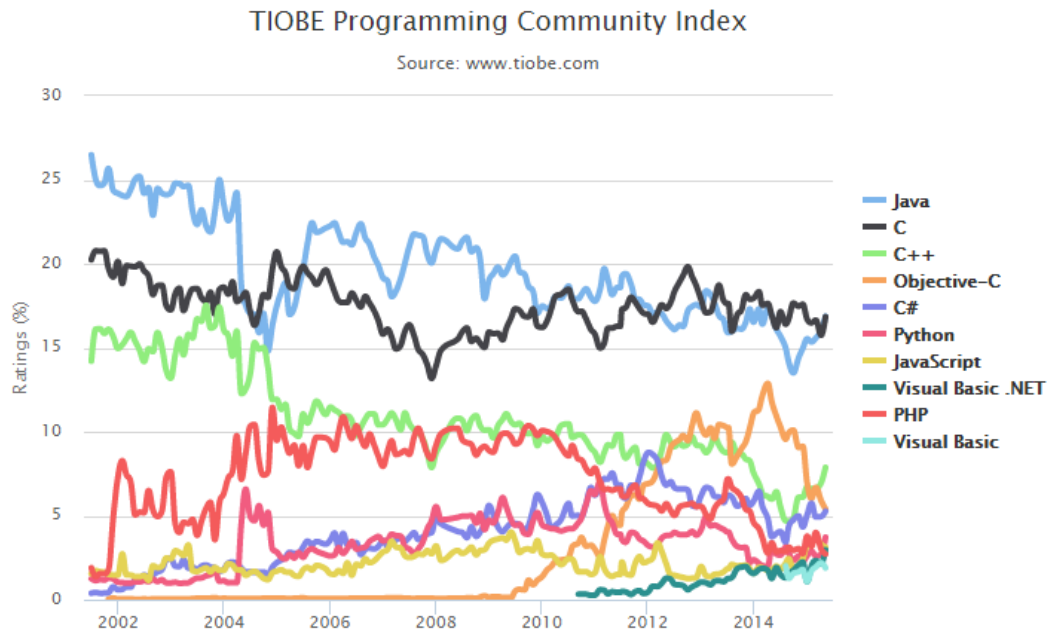


Guido van Rossum

Πηγή :  
[https://en.wikipedia.org/wiki/Guido\\_van\\_Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum)



# python: Η σημασία της



## Οι 10 πιο διαδεδομένες γλώσσες προγραμματισμού

By TIOBE Software B.V. [CC BY-SA 4.0 (<http://creativecommons.org/licenses/by-sa/4.0/>)], via Wikimedia Commons

Πηγή: <https://commons.wikimedia.org/wiki/File:Tiobeindex.png>

Επίκαιρη έκδοση διαθέσιμη [εδώ](#)

- Στις 10 πιο διαδεδομένες γλώσσες προγραμματισμού
- Συζητείται ευρέως για web, dbs & academic computing ([πηγή](#))
- Ευρέως διαδεδομένη ως γλώσσα εκμάθησης προγραμματισμού σε πανεπιστημιακό επίπεδο ([πηγή](#))

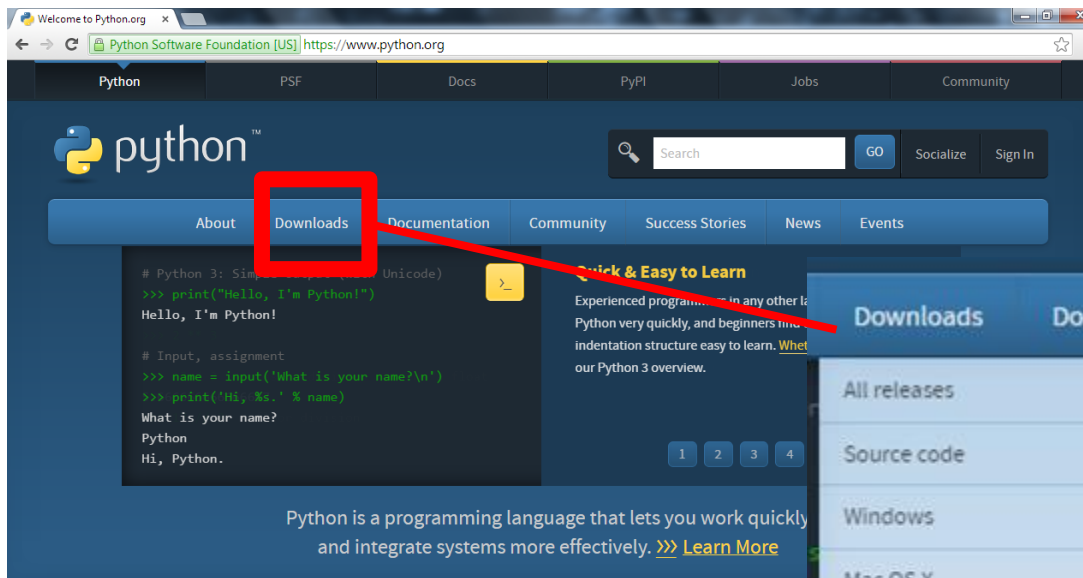
# Κύρια χαρακτηριστικά της python

- Ανοικτός κώδικας
- Προσπάθεια χρήσης όρων κοντά στη φυσική γλώσσα (αγγλικά)->ευκολία στην κατανόηση
- Χρησιμοποιεί διερμηνευτή εντολών και παρουσιάζει άμεσα όποια λάθη προγραμματισμού
- Φορητή σε διαφορετικά περιβάλλοντα
- Μότο: *"there should be one—and preferably only one—obvious way to do"*  
αντί "there is more than one way to do it"



# Εγκατάσταση python

1) Επισκέπτομαι τη σελίδα <http://www.python.org/>



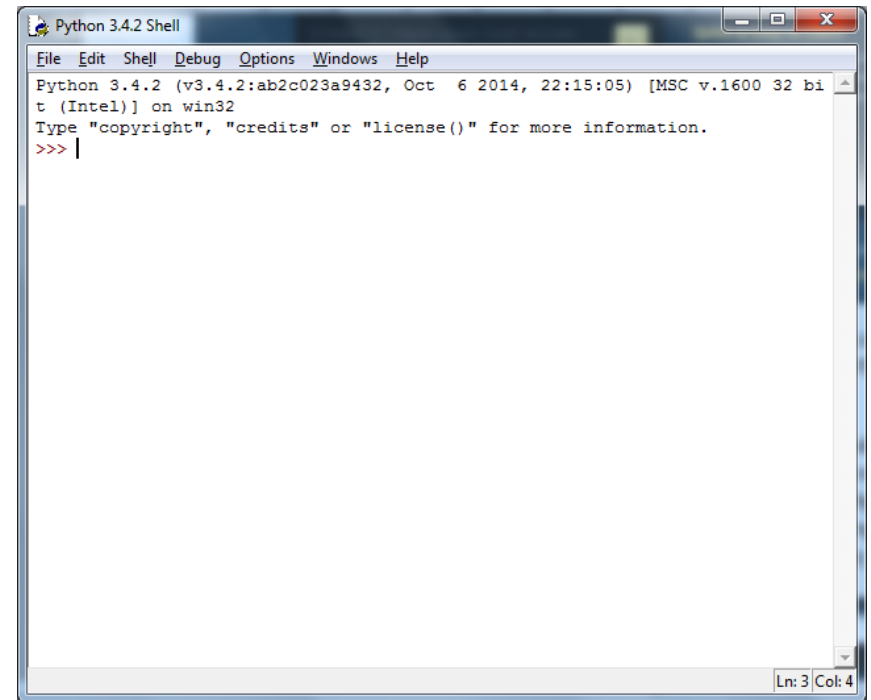
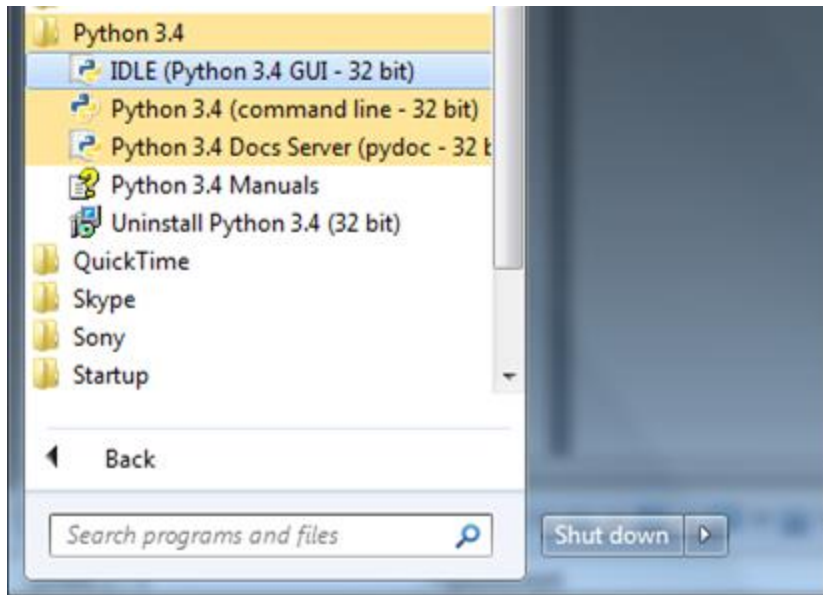
2) Download Python 3.4.2





# Εγκατάσταση python (συνέχεια)

3) Κατεβάζω και εκτελώ το αρχείο εγκατάστασης με την κλασική διαδικασία. Μετά εκτελώ το IDLE (από το Start menu):



Υπολογισμοί και μεταβλητές

# Υπολογισμοί και τελεστές

```
>>> (2+5) * 5
35
>>> 2 ** 2
4
>>> 14 / 4
3.5
>>> 14 // 4
3
>>> 14 % 4
2
>>> 2 ** 1000
```

**	ύψωση σε δύναμη
/	διαίρεση
//	πηλίκο ευκλείδιας διαίρεσης
%	υπόλοιπο ευκλείδιας διαίρεσης
*	πολλαπλασιασμός
+	πρόσθεση
-	αφαίρεση

```
1071508607186267320948425049060001810561404811705533
60744375038837035105112493612249319837881569585812
75946729175531468251871452856923140435984577574698
57480393456777482423098542107460506237114187795418
21530464749835819412673987675591655439460770629145
71196477686542167660429831652624386837205668069376
```



# Μεταβλητές

- Ονόματα:
  - Γράμματα, ψηφία, ή underscores `_`, ξεκινούν πάντα από γράμμα
- Δεσμευμένες λέξεις
- Case sensitive
- Ορθά ονόματα μεταβλητών
  - π.χ. `theWordOfWarcraft`



# Μεταβλητές και εκχώρηση

## Μεταβλητή:

- ένας “κουβάς” όπου τα δεδομένα αποθηκεύονται
- το όνομα της μεταβλητής μάς δίνει τη δυνατότητα αναφοράς στα δεδομένα του “κουβά”, στην τιμή της μεταβλητής

## Εκχώρηση:

- `var b1=5`: με το `=` πραγματοποιείται αποθήκευση τιμών στη μεταβλητή, δεν αποτελεί ισότητα
- `a=a+a`: η χρήση του ονόματος μεταβλητής στο δεξί μέρος της εκχώρησης αναφέρεται στην τιμή της μεταβλητής



# Βασικοί τύποι μεταβλητών

**Συμβολοσειρές (str):** (σε μονά ή διπλά εισαγωγικά)

```
myst r='Hello Nikos',  
grGM="Καλημέρα",  
yourPhone='2610459220'
```

**Ακέραιοι (int):**

```
a=6, b=1234, c=-567
```

**Πραγματικοί (float):**

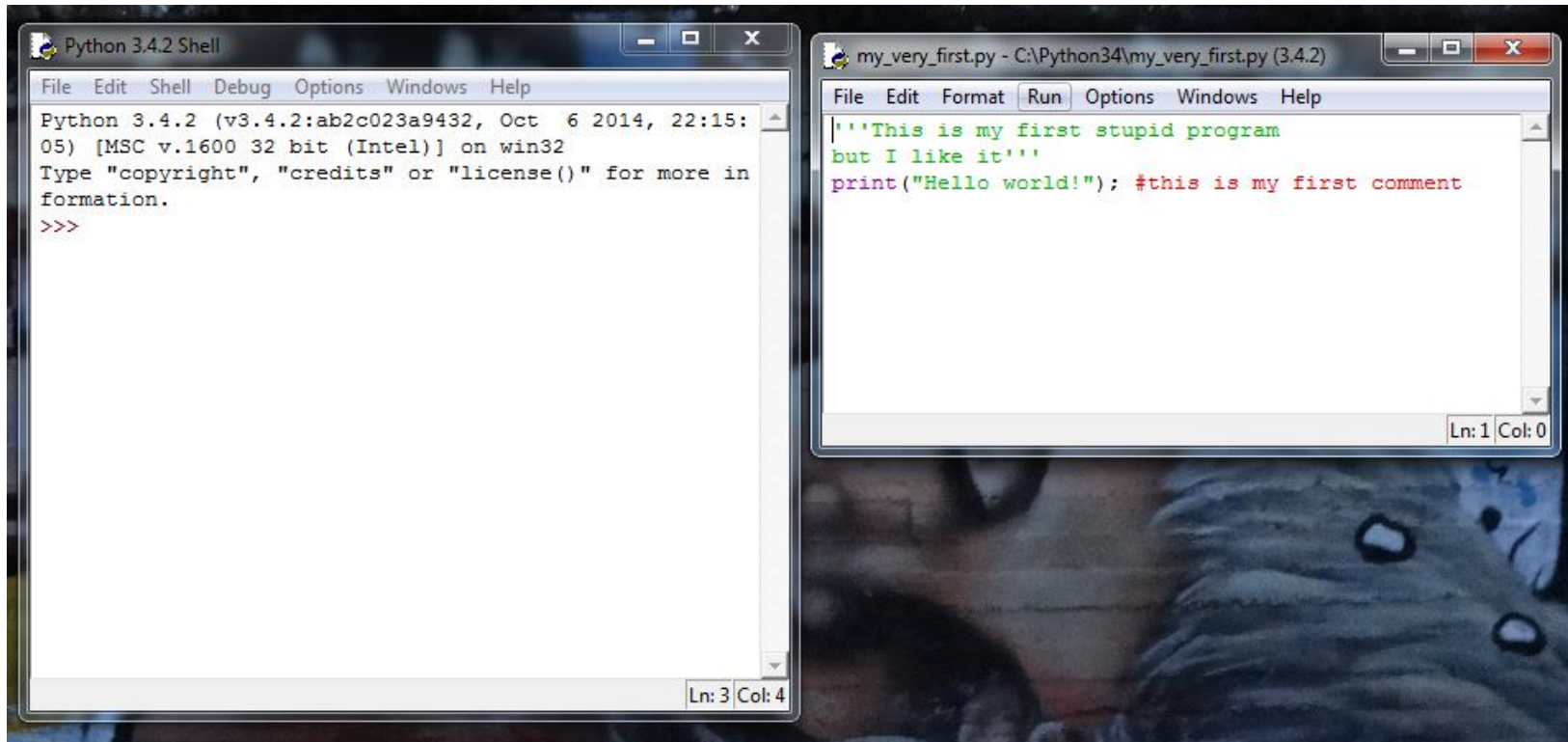
```
x=6.2, y=52.3E-4, z=-567.56789
```

**Boolean:** True, False



# Το πρώτο πρόγραμμα

IDLE->File->New File



The image shows two overlapping windows from the Python 3.4.2 IDLE environment. The left window is the 'Python 3.4.2 Shell' with a menu bar (File, Edit, Shell, Debug, Options, Windows, Help) and a text area containing the following text:

```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:15:05) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more in
formation.
>>>
```

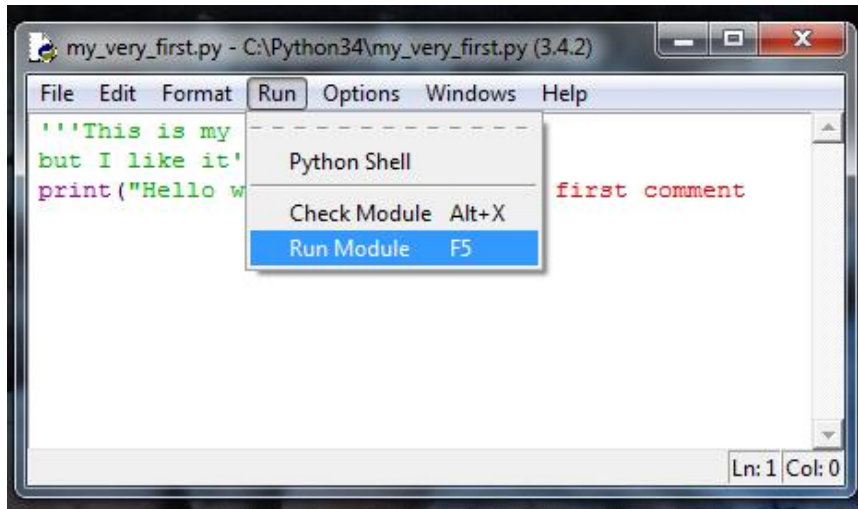
The status bar at the bottom right of this window shows 'Ln: 3 Col: 4'. The right window is the 'my\_very\_first.py - C:\Python34\my\_very\_first.py (3.4.2)' editor with a menu bar (File, Edit, Format, Run, Options, Windows, Help) and a text area containing the following code:

```
'''This is my first stupid program
but I like it'''
print("Hello world!"); #this is my first comment
```

The status bar at the bottom right of this window shows 'Ln: 1 Col: 0'. The background of the slide features a close-up image of a blue and white patterned fabric.



# Το πρώτο πρόγραμμα-Εκτέλεση



my\_very\_first.py - C:\Python34\my\_very\_first.py (3.4.2)

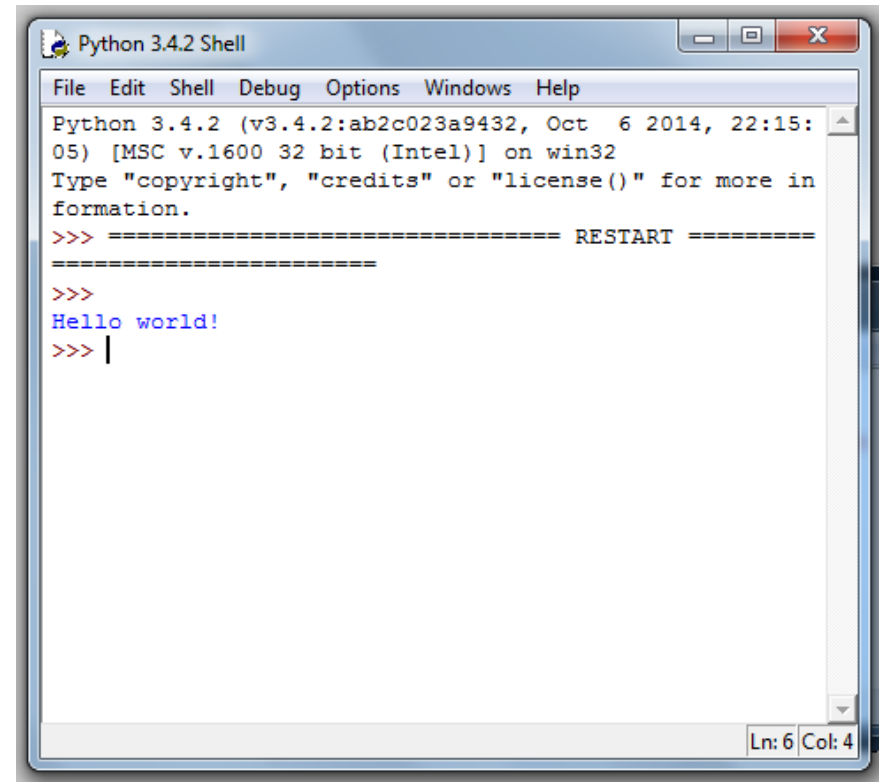
```
File Edit Format Run Options Windows Help
```

```
'''This is my  
but I like it'  
print("Hello w
```

Python Shell  
Check Module Alt+X  
Run Module F5

first comment

Ln: 1 Col: 0



Python 3.4.2 Shell

```
File Edit Shell Debug Options Windows Help
```

```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:15:  
05) [MSC v.1600 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more in  
formation.  
>>> ===== RESTART =====  
>>>  
Hello world!  
>>> |
```

Ln: 6 Col: 4





# Βήματα για δημιουργία προγράμματος

1. Στο IDLE Shell επιλέγω *File->New file*
2. Εμφανίζεται το παράθυρο του IDLE επεξεργαστή (IDLE editor)
3. Δημιουργούμε τον κώδικα του προγράμματος με εντολές της *python*
4. Εκτελούμε το πρόγραμμα επιλέγοντας στο παράθυρο του επεξεργαστή *Run->Run Module* (ή F5) και μας ζητά να αποθηκεύσουμε το πρόγραμμά μας
5. Στο παράθυρο του IDLE Shell εμφανίζει *RESTART* και ξεκινά η εκτέλεση του προγράμματος μας



# Συμβολοσειρές

```
>>> 'very ' + 'hot'
'very hot'
>>> 3*'very ' + 'hot'
'very very very hot'
>>> '7'+ '2'
'72'
>>> type('dog')
<class 'str'>
>>> type('7')
<class 'str'>
>>> type(7)
<class 'int'>
```

```
>>> justaTest = '''Say,
"I'm in!"
This is line 3'''
>>> print(justaTest)
Say,
"I'm in!"
This is line 3
>>>
```

- Συμβολοσειρές: γράμματα και ψηφία σε μονά ή διπλά εισαγωγικά
- Χρήση '''Συμβολοσειρά''' (τριπλά εισαγωγικά) για πολλαπλές γραμμές
- Χρήση + για συνένωση συμβολοσειρών



# Είσοδος με την input()

```
>>> a=input("Δώσε τον πρώτο #: ")
Δώσε τον πρώτο #: 5
>>> b=input("Δώσε τον δεύτερο #: ")
Δώσε τον δεύτερο #: 10
>>> a+b
'510'
>>> int(a)+int(b)
15
>>> x=float(input('Δώσε έναν
πραγματικό: '))
Δώσε έναν πραγματικό: 567.1234
>>> print(x)
567.1234
```

*input(prompt)*

- Όρισμα: Το κείμενο που εμφανίζεται στο χρήστη για την είσοδο (prompt)
- Επιστρέφει τη συμβολοσειρά που έδωσε ο χρήστης
- Είσοδος ακεραίων:  
`int(input("Δώσε ακέραιο: "))`
- Είσοδος πραγματικών:  
`float(input("Δώσε πραγματικό: "))`



# Έξοδος με την print()

```
>>> person = input('Enter your name: ')
Enter your name: Victoria
>>> print('Hello', person, '!')
Hello Victoria !
>>> print('Hello ', person, '!', sep='')
Hello Victoria!
>>> print('Hello\n', person, '\n!', sep='')
Hello
Victoria
!
```

**print**(\*objects, sep=' ', end='\n',  
file=sys.stdout, flush=False)

Παίρνει ως κύριο όρισμα αντικείμενα (*objects*) που τα εκτυπώνει σε αρχείο (*file*) διαχωριζόμενα από τη συμβολοσειρά *sep*, τελειώνοντας σε *end* προκαθορισμένο *file: sys.stdout* (standard output=οθόνη)

Ο χαρακτήρας '\n': αλλαγή γραμμής (new line)



# Βασικές μαθηματικές συναρτήσεις

```
from math import *
```

Χρήσιμες συναρτήσεις:

- `pi` - επιστρέφει προσέγγιση του  $\pi$
- `radians(x)` - μετατρέπει τις  $x$  μοίρες σε radians
- `sin(x)` - επιστρέφει το ημίτονο των  $x$  radians
- `cos(x)` - επιστρέφει το συνημίτονο των  $x$  radians
- `tan(x)` - επιστρέφει την εφαπτομένη των  $x$  radians
- `sqrt(x)` - επιστρέφει την τετραγωνική ρίζα του  $x$



# Εργασίες: μεταβλητές & υπολογισμοί

1. Γράψτε πρόγραμμα που να διαβάσει 5 πραγματικούς αριθμούς και να υπολογίζει το μέσο όρο τους
2. Γράψτε πρόγραμμα που να διαβάσει τα  $\alpha$ ,  $\beta$ ,  $\gamma$  του τριωνύμου  $(\alpha x^2 + \beta x + \gamma)$  και να το υπολογίζει για ένα συγκεκριμένο  $x$  που θα δίνει ο χρήστης
3. Γράψτε πρόγραμμα που να διαβάσει βάση και ύψος και να υπολογίζει το εμβαδό του τριγώνου και του ορθογωνίου
4. Ένα κυκλικό συντριβάνι βρίσκεται σε ορθογώνια αυλή. Γράψτε πρόγραμμα που να διαβάσει πλάτος και μήκος της αυλής, διάμετρο του κύκλου (σε m) και να υπολογίζει πόσο θα πληρωθεί η πλήρης κάλυψη της αυλής με πλακάκια διάστασης (25x35cm) που κοστίζουν 15 ευρώ/ένα [χρήση `round()`]
5. Ένας αμερικάνος ταξιδιώτης έρχεται στην Ελλάδα με συγκεκριμένο συνάλλαγμα σε δολάρια. Γράψτε πρόγραμμα που να διαβάσει το ποσό και να υπολογίζει σε πόσα ευρώ αντιστοιχούν (η τρέχουσα ισοτιμία ως σταθερά, αγνοώντας τα λεπτά) και πόσα χαρτονομίσματα των 50, 20, 10 και 5 ευρώ θα πάρει.



Συμβολοσειρές

# Συμβολοσειρές: Λειτουργίες (1)

- `yourString.upper()` - η συμβολοσειρά σε κεφαλαία
- `yourString.lower()` - η συμβολοσειρά σε πεζά
- `yourString.capitalize()` - η συμβολοσειρά με το πρώτο γράμμα κεφαλαίο
- `yourString.title()` - η συμβολοσειρά με το πρώτο γράμμα κάθε λέξης κεφαλαίο
- `yourString.replace(x, y)` - η συμβολοσειρά με αντικατάσταση του χαρακτήρα `x` με χαρακτήρα `y`
- `len(yourString)` - το μήκος της συμβολοσειράς

Περισσότερες συναρτήσεις σε συμβολοσειρές:

- <https://docs.python.org/3.4/library/stdtypes.html#string-methods>





# Συμβολοσειρές: Λειτουργίες (1)-συν.

```
>>> yourString='The answer to the ultimate question of life, the universe and
    everything is 42.'
>>> yourString.upper()
'THE ANSWER TO THE ULTIMATE QUESTION OF LIFE, THE UNIVERSE AND EVERYTHING IS 42.'
>>> yourString.lower()
'the answer to the ultimate question of life, the universe and everything is 42.'
>>> yourString.capitalize()
'The answer to the ultimate question of life, the universe and everything is 42.'
>>> yourString.title()
'The Answer To The Ultimate Question Of Life, The Universe And Everything Is 42.'
>>> yourString.replace('a', 'A')
'the Answer to the ultimAte question of Life, the Universe And Everything is 42.'
```



# Συμβολοσειρές-Επιπλέον λειτουργίες

myStr	T	h	e		a	n	s	w	e	r		i	s		4	2
δείκτης	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Κάθε συμβολοσειρά είναι ένας πίνακας χαρακτήρων

- `myStr[start:stop]` - το τμήμα της συμβολοσειράς από `start` έως `stop-1`
- `myStr[start:]` - το τμήμα της συμβολοσειράς από `start` έως το τέλος
- `myStr[:stop]` - το τμήμα της συμβολοσειράς από αρχή έως το `stop-1`
- `myStr[:]` - όλη η συμβολοσειρά
- `myStr[-x]` - ο `x`-στός χαρακτήρας μετρώντας από δεξιά
- `myStr[-x:]` - οι τελευταίοι `x` χαρακτήρες
- `myStr[:-x]` - όλη η συμβολοσειρά εκτός των τελευταίων `x` χαρακτήρων



Έλεγχος ροής

# Έλεγχος ροής προγράμματος - if

```
if <συνθήκη>:  
    # εντολές  
else:  
    # εντολές
```

```
vathmos= int(input("Ποιός  
είναι ο βαθμός σου; "))  
if vathmos>= 5:  
    print("Περνάς :-)")  
else:  
    print("Απορρίπτεισαι :-(")
```

Τελεστές σύγκρισης:	== ίσο !=, <> διάφορο >, >= <, <=
Λογικοί τελεστές:	not, π.χ. not(a) and, π.χ. (a and b) or, π.χ. (a or b)
Τελεστές μέλους	in, not in, έλεγχοι συμμετοχής, μία μεταβλητή μέρος μίας δομής δεδομένων

**Προσοχή: Οι εσοχές (indent) έχουν σημασία! Καθορίζουν τις εντολές που περιέχει κάθε block**



# Περισσότερα για το if

```
if <συνθήκη>:
```

```
    <εντολές>
```

```
elif <συνθήκη2>:
```

```
    <εντολές>
```

```
else:
```

```
    <εντολές>
```

```
kairos = input("Πώς είναι ο καιρός σήμερα; ")
```

```
if kairos == "βροχερός":
```

```
    print("Πάρε καλύτερα μία ομπρέλλα")
```

```
elif kairos == "κρύος":
```

```
    print("Πάρε ένα παλτό")
```

```
elif kairos == "ηλιόλουστος":
```

```
    print("Φόρεσε το αντηλιακό σου")
```

```
else:
```

```
    print("Απόλαυσε τη μέρα, ότι καιρό κι αν έχει!")
```



# Επανάληψη - for

```
for <μεταβλητή> in
    <εύρος>:
        <εντολές>

for counter in range(5):
    print("hello
world")
print("outside for")
```

range(από, έως, βήμα)

- από, έως: προαιρετικά
- έως : υποχρεωτικό
- από, έως, βήμα: **ακέραιοι**

Παραδείγματα:

- range(10): [0,1,2,3,4,5,6,7,8,9]
- range(1, 7): [1,2,3,4,5,6]
- range(0, 30, 5): [0,5,10,15,20,25]
- range(5, -1, -1): [5,4,3,2,1,0]



# Επανάληψη - while

```
while <συνθήκη_αληθής>:  
    <εντολή1>  
    <εντολή2>
```

```
number = 23  
running = True  
while running:  
    guess = int(input('Εισάγετε έναν ακέραιο αριθμό : '))  
    if guess == number:  
        print('Συγχαρητήρια, τον μαντέψατε.')  
        running = False # while stops here  
    elif guess < number:  
        print('Όχι, είναι λίγο μεγαλύτερος.')  
    else:  
        print('Όχι, είναι λίγο μικρότερος.')  
else:  
    print('Ο βρόχος while τερματίστηκε.')
```



# Επανάληψη while με break, continue

```
while True:
    s = input('Εισάγετε κάτι : ')
    if s == 'quit':
        break
    if len(s) < 3:
        print('Πολύ μικρό')
        continue
    print('Το μήκος των εισαχθέντων είναι επαρκές')
# Προσθέστε οτιδήποτε άλλο εδώ
```





# Εργασίες: if, for, while

1. Γράψτε πρόγραμμα που ζητά από τον χρήστη δύο τιμές και αν το άθροισμά τους είναι πάνω από το 100 να εμφανίζει κατάλληλο μήνυμα
2. Γράψτε πρόγραμμα που ζητά από το χρήστη αριθμό και υπολογίζει αν είναι άρτιος ή περιττός
3. Γράψτε πρόγραμμα που ζητά από το χρήστη τις συντεταγμένες  $(x_1, y_1)$ ,  $(x_2, y_2)$  δύο σημείων μίας ευθείας  $y=ax+\beta$  και υπολογίζει τα  $a$  και  $\beta$  της
4. Γράψτε πρόγραμμα που ζητά από το χρήστη τον αριθμό των πραγματικών που θα διαβάσει στη συνέχεια για να βρεί τον μέσο όρο τους
5. Γράψτε πρόγραμμα που ζητά από το χρήστη το  $n$  και υπολογίζει το  $n!$
6. Γράψτε πρόγραμμα που δημιουργεί τους 20 πρώτους αριθμούς από την [ακολουθία Φιμπονάτσι](#)
7. Γράψτε πρόγραμμα που ζητά από το χρήστη τα  $a, \beta, \gamma$  του τριωνύμου  $(ax^2+bx+\gamma)$  και υπολογίζει τις ρίζες του



Λίστες

# Λίστες

- Δυναμική δομή δεδομένων για διατεταγμένη συλλογή στοιχείων
- Όχι απαραίτητα του ίδιου τύπου
- Τα στοιχεία μέσα σε []
- Χρησιμοποιούμε λίστες στην python για την αναπαράσταση πινάκων

```
zoa= ["άλογο", "σκυλί", "κότα", 1, 2, 3]
```

```
pinakas2D= [ [ 2, 3, 5] , [ 1, 4, 7 ] ]
```



# Λίστες: δημιουργία και κατάτμηση

```
>>> shoplist =
    ['μήλο', 'μάνγκο', 'καρότο', 'μπανάνα', 'αχλάδι', 'σύκο']
>>> shoplist[0]
'μήλο'
>>> shoplist[0:2]
['μήλο', 'μάνγκο']
>>> shoplist[0:6:2] #from 0 to 5, step 2
['μήλο', 'καρότο', 'αχλάδι']
>>> shoplist[6:2:-1] #from 6 to 3, step -1
['σύκο', 'αχλάδι', 'μπανάνα']
>>> shoplist[-1] # 1 before len
'σύκο'
>>> shoplist[1:-2] #from 1 to 2nd before len
['μάνγκο', 'καρότο', 'μπανάνα']
```



# Διατρέχω μία λίστα με for & in

```
>>> for each in  
      shoplis:
      print(each)
```

```
μήλο  
μάνγκο  
καρότο  
μπανάνα  
αχλάδι  
σύκο  
>>>
```

```
>>> for each in  
      shoplis:
          if each[0]=='μ':
              print(each)
```

```
μήλο  
μάνγκο  
μπανάνα  
>>>
```



# Διατρέχω λίστες 2D

```
# Δημιουργία λίστας 2d
a = [ [ 2, 3, 5] , [ 1, 4, 7 ] ]
print ("Before: a =", a)
# βρίσκω διαστάσεις λίστας
rows = len(a)
cols = len(a[0])
# διατρέχω τη λίστα και προσθέτω 1 σε κάθε στοιχείο
for row in range(rows):
    for col in range(cols):
        a[row][col] += 1
# εκτυπώνω τα νέα στοιχεία της λίστας
print ("After: a =", a)
```



# Λίστες

## Μέθοδοι:

len(list)  
list.reverse()  
list.append()  
list.insert(position,object)  
list.pop(position)  
list.index(item)  
list.sort()  
list.remove(item)  
list.count(item)

```
>>> shoplist
['μήλο', 'μάνγκο', 'καρότο', 'μπανάνα', 'αχλάδι', 'σύκο']
>>> len(shoplist)
6
>>> shoplist.reverse()
>>> shoplist
['σύκο', 'αχλάδι', 'μπανάνα', 'καρότο', 'μάνγκο', 'μήλο']
>>> shoplist.append('πορτοκάλι') # add at the end
>>> shoplist
['σύκο', 'αχλάδι', 'μπανάνα', 'καρότο', 'μάνγκο', 'μήλο', 'πορτοκάλι']
>>> shoplist.insert(3,'σταφύλι') # insert before position 3
>>> shoplist
['σύκο', 'αχλάδι', 'μπανάνα', 'σταφύλι', 'καρότο', 'μάνγκο', 'μήλο',
'πορτοκάλι']
>>> shoplist.pop(5) # delete item from position 5
'μάνγκο'
>>> shoplist
['σύκο', 'αχλάδι', 'μπανάνα', 'σταφύλι', 'καρότο', 'μήλο', 'πορτοκάλι']
```



# Εργασίες σε λίστες

1. Γράψτε πρόγραμμα που να δημιουργεί λίστα, η οποία θα περιέχει τους άρτιους αριθμούς από το 1 έως το 50. Θα εκτυπώνει τη λίστα και το μήκος της.
2. Γράψε πρόγραμμα που να διαβάζει συμβολοσειρά με λέξεις που χωρίζονται με τον χαρακτήρα ';' και να τις εκχωρεί ως στοιχεία σε λίστα, την οποία θα εκτυπώνει.
3. Γράψτε πρόγραμμα που θα διαβάζει το πλήθος των θετικών αριθμών, τους οποίους θα διαβάζει στη συνέχεια και θα τοποθετεί σε μία λίστα. Ακολουθώς:
  - a. Εκτυπώνει τον ελάχιστο, το μέγιστο και το μέσο όρο της
  - b. Εκτυπώνει ένα (κάθετο) ιστόγραμμα της από χαρακτήρες '\*'
  - c. Εκτυπώνει την αθροιστική της λίστα. Π.χ. η [1,2,3] έχει ως αθροιστική την [1,3,6].
4. Γράψτε πρόγραμμα που δημιουργεί λίστα από συμβολοσειρές που θα πληκτρολογεί ο χρήστης έως τη λέξη 'end'. Στη συνέχεια, θα την ταξινομή και θα ζητά από το χρήστη να δώσει συμβολοσειρά, την οποία θα αφαιρεί από τη λίστα (αν υπάρχει).
5. Γράψτε πρόγραμμα που κάθε στοιχείο της δισδιάστατης λίστας  $a = [ [ 2, 3, 5 ] , [ 1, 4, 7 ] ]$  θα το αντικαθιστά με το τετράγωνό του.





Αρχεία

# Αρχεία δεδομένων σε μορφή κειμένου

Σε διάφορα δημόσια αποθετήρια υπάρχουν δεδομένα σε μορφή πίνακα αποθηκευμένα ως αρχεία κειμένου τα οποία θέλουμε να επεξεργαστούμε με `rython`:

- Αρχεία κειμένου, δεδομένα σε γραμμές ή σε στήλες και χωρισμένα με τον χαρακτήρα `tab`
  - Αρχεία συνήθως με κατάληξη `.txt`
- Αρχεία κειμένων με δομημένη μορφή CSV (Comma Separated Value), χωρισμένα με χαρακτήρα κόμμα `“,”` ή άλλους χαρακτήρες όπως το `“;”` (ελληνικό ερωτηματικό)
  - Αρχεία συνήθως με κατάληξη `.csv`

*Όλα τα αρχεία κειμένου επεξεργάζονται με απλούς επεξεργαστές κειμένου όπως το MS notepad*

# Διάβασμα από αρχείο κειμένου (1)

```
with open("animals.txt", mode="r", encoding="utf-8") as my_file:
    for line in my_file:
        print(line)
```

Βελτίωση: `print(line.rstrip("\n"))`

## Μορφή αρχείου animals.txt

```
antelope
bear
cat
dog
elephant
fox
```

`open(file, mode='r', buffering=-1, encoding=None, errors=None, newline=None, closefd=True, opener=None)`

- `file`: όνομα αρχείου
- `mode`:
  - **w** - ανοίγει το αρχείο για εγγραφή (διαγράφει τα υπάρχοντα περιεχόμενα)
  - **r** - ανοίγει το αρχείο για ανάγνωση
  - **a** - ανοίγει το αρχείο και γράφει στο τέλος του διατηρώντας τα υπάρχοντα περιεχόμενα
- `encoding`: για αρχεία text, το προκαθορισμένο εξαρτάται από την πλατφόρμα & χρησιμοποιούμε το UTF8



# Διάβασμα από αρχείο κειμένου(2)

- `myfile.read()` -> διαβάζει όλες τις γραμμές σε μία συμβολοσειρά `'antelope\nbear\ncat\ndog\nelephant\nfox'`
- `myfile.readlines()` -> διαβάζει όλες τις γραμμές σε μία λίστα  
`['antelope', 'bear', 'cat', 'dog', 'elephant', 'fox']`
- `myfile.read().splitlines()` -> διαβάζει όλες τις γραμμές σε μία συμβολοσειρά και τη χωρίζει σε γραμμές  
`['antelope\n', 'bear\n', 'cat\n', 'dog\n', 'elephant\n', 'fox']`
- `for line in myfile` -> διαβάζει μία-μία τις γραμμές του αρχείου ως συμβολοσειρές στη μεταβλητή `line`



# Διάβασμα από αρχείο CSV

## Μορφή αρχείου kalesmenoi.csv

```
"Οικ. Παπαδόπουλος";"Γαμπρού";2;3  
"Οικ. Φίδις";"Νύφης";1;1  
"Οικ. Ακρίδας";"Κουμπάρου";2;2  
"Οικ. Σαλούστρος";"Κουμπάρου";2;0  
"Οικ. Παπαγιάννη";"Νύφης";2;0
```

```
import csv  
with open('D:\TOE\mathima\kalesmenoi.csv', mode='r', newline='', encoding='utf-8',) as f:  
    reader=csv.reader(f, delimiter=';', quotechar='"', quoting=csv.QUOTE_NONNUMERIC)  
    for row in reader:  
        print(row)
```

1. Χρήση βιβλιοθήκης csv για αρχεία τύπου CSV (Comma Separated Values)
2. Άνοιγμα του αρχείου με την `open()`
3. Χρήση συνάρτησης `csv.reader()` για διάβασμα από αρχείο. Ορισμός του `delimiter`, `quotechar` και μετατροπή `non-quoted` σε `float`
4. Χρήση του `for in` για πρόσβαση σε μία-μία τις γραμμές



# Διάβασμα από αρχείο CSV : εύρεση sum στήλης

```
import csv
names=[]
poiou=[]
megaloi=[]
paidia=[]
with open('D:\TOE\mathima\kalesmenoi.csv', mode='r', newline='',encoding='utf-8',) as f:
    reader=csv.reader(f,delimiter=';',quotechar='"', quoting=csv.QUOTE_NONNUMERIC)
    for row in reader:
        name,poianou,adult,child=row
        names.append(name)
        poiou.append(poianou)
        megaloi.append(adult)
        paidia.append(child)
print(sum(megaloi))
print(sum(paidia))
```



# Διάβασμα από αρχείο CSV: εύρεση sum στήλης (2ος) τρόπος

```
import csv
with open('D:\TOE\mathima\kalesmenoi.csv', mode='r', newline='', encoding='utf-8',) as f:
    reader=csv.reader(f, delimiter=';', quotechar='"', quoting=csv.QUOTE_NONNUMERIC)
    names, poiou, megaloi, paidia=zip(*reader)
print(sum(megaloi))
print(sum(paidia))
```

- Χρήση συνάρτησης `zip` (\*πίνακας) και χωρίζει σε στήλες



# Βιβλιοθήκη Στατιστικής

```
from statistics import *
```

mean() ->Αριθμητικός μέσος

median() ->Διάμεσος

mode() ->Επικρατούσα τιμή διακριτών μεταβλητών

pstdev() ->Τυπική απόκλιση πληθυσμού

pvariance() ->Διακύμανση πληθυσμού

stdev() ->Τυπική απόκλιση δείγματος

variance() ->Διακύμανση δείγματος

+ Βασικές  
συναρτήσεις  
βαθμωτών  
τύπων

- max()
- min()
- sum()





# Έξοδος σε αρχείο csv

## Μορφή αρχείου εξόδου

```
"name";"poianou";"megaloi";"paidia"  
"Οικ. Παπαδόπουλος";"Γαμπρού";2;3  
"Οικ. Φίδις";"Νύφης";1;1  
"Οικ. Ακρίδας";"Κουμπάρου";2;2
```

```
import csv  
data = [['name', 'poianou', 'megaloi', 'paidia'], ['Οικ. Παπαδόπουλος', 'Γαμπρού', 2.0, 3.0]]  
with open('myOutput.csv', mode='w', newline='') as myWFile:  
    writer = csv.writer(myWFile,  
                        delimiter=';', quotechar='"', quoting=csv.QUOTE_NONNUMERIC)  
    for line in data:  
        writer.writerow(line)
```

1. Άνοιγμα του αρχείου με την `open()` με `mode=w` (γράψιμο)
2. Χρήση συνάρτησης `csv.writer()` για γράψιμο στο αρχείο. Ορισμός του `delimiter`, `quotechar` και μετατροπή `non-quoted` σε `float`
3. Χρήση του `for in` για πρόσβαση σε μία-μία τις γραμμές
4. Χρήση `writerow` για γράψιμο γραμμής στο αρχείο



# Παράδειγμα δημόσιων δεδομένων: Απογραφή πληθυσμού

- Έστω αρχείο με πληροφορίες για τον πληθυσμό της Ελλάδας, όπως αυτές προέκυψαν κατά την απογραφή του 2011 και δημοσιεύτηκαν από την Ελληνική Στατιστική Αρχή (<http://www.statistics.gr/>\*)
- Το αρχείο περιέχει στοιχεία του πληθυσμού για κάθε διαμέρισμα, περιφέρεια, δήμο και δημοτική ενότητα της Ελλάδας κατά φύλλο και ομάδες ηλικιών

*\*Το φύλλο εργασίας είναι διαθέσιμο στη σελίδα*

<http://www.statistics.gr/portal/page/portal/ESYE/PAGE-census2011tables> (Πίνακες Απογραφής 201, Πίνακας 4α: Απογραφή Πληθυσμού 2011. Μόνιμος Πληθυσμός κατά φύλλο και ομάδες ηλικιών Μεγάλες Γεωγραφικές Ενότητες (NUTS 1), Αποκεντρωμένες Διοικήσεις, Περιφέρειες (NUTS 2), Περιφερειακές Ενότητες, Δήμοι, Δημοτικές Ενότητες )

# Τμήμα αρχείου απογραφής ως CSV

## Δημιουργία Αρχείου apograph.csv

```
"level";"code";"descr";"population"  
0;"          ";"ΣΥΝΟΛΟ ΧΩΡΑΣ";10815197  
1;"1  ";"ΒΟΡΕΙΑ ΕΛΛΑΔΑ";3110596  
2;"11 ";"ΑΠΟΚΕΝΤΡΩΜΕΝΗ ΔΙΟΙΚΗΣΗ ΜΑΚΕΔΟΝΙΑΣ - ΘΡΑΚΗΣ";2490051  
3;"111 ";"ΠΕΡΙΦΕΡΕΙΑ ΑΝΑΤΟΛΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ ΚΑΙ ΘΡΑΚΗΣ";608182  
4;"11101";"ΠΕΡΙΦΕΡΕΙΑΚΗ ΕΝΟΤΗΤΑ ΡΟΔΟΠΗΣ";112039  
5;"1110101";"ΔΗΜΟΣ ΚΟΜΟΤΗΝΗΣ";66919
```

- Κάθε γραμμή του αρχείου αφορά μία γεωγραφική περιοχή
- Οι στήλες της κάθε γραμμής του αρχείου:
  - Επίπεδο γεωγραφικής περιοχής (π.χ. Οι Δήμοι χαρακτηρίζονται στο επίπεδο 5)
  - Κωδικός
  - Περιγραφή
  - Πληθυσμός



# Εργασίες σε αρχεία και λίστες

Στο eclass στο χώρο του Μαθήματος Έγγραφα->Σημειώσεις->Python υπάρχει το αρχείο kalesmenoι.csv

Γράψτε σταδιακά πρόγραμμα σε python που να επεξεργάζεται το αρχείο CSV, το οποίο επεκτεινόμενο σε κάθε νέο ερώτημα να κάνει τα ακόλουθα:

1. Να διαβάζει το αρχείο γραμμή-γραμμή και να την εκτυπώνει στην οθόνη
2. Να αγνοεί την πρώτη γραμμή και να εκτυπώνει τις υπόλοιπες στην οθόνη
3. Να εκχωρεί τις τιμές των στηλών του αρχείου csv σε κατάλληλες δομές τύπου λίστας
4. Να εκχωρεί μόνο τις τιμές των καλεσμένων της νύφης σε κατάλληλες δομές τύπου λίστας
5. Να βρίσκει το όνομα της οικογένειας με το μεγαλύτερο αριθμό παιδιών που είναι καλεσμένοι της νύφης



# Διάβασμα από csv σε λεξικό (1)

```
# Βρίσκει το μέσο όρο & τυπική απόκλιση του πληθυσμού δήμου του αρχείου apograph.csv
import csv,sys
from statistics import *
dhmoi=[]
filename='apograph.csv'
with open(filename, mode='r', newline='',encoding='utf-8',) as f:
    reader=csv.DictReader(f,delimiter=';',quotechar='', quoting=csv.QUOTE_NONNUMERIC)
    try:
        for row in reader:
            if row['level']==5:
                dhmoi.append(row['population'])
        print('Μέσος δήμων: '+'{:0.2f}'.format(mean(dhmoi)))
        print('Τυπική απόκλιση δήμων: '+'{:0.2f}'.format(stdev(dhmoi)))
    except csv.Error as e:
        sys.exit('file %s, line %d: %s' % (filename, reader.line_num, e))
```



# Διάβασμα από csv σε λεξικό (2)

## Νέα στοιχεία:

- Χρήση `csv.DictReader`: Διαβάζει σε δομή λεξικού για κάθε γραμμή:  

```
{'level': 6.0, 'population': 6350.0, 'descr':  
'ΔΗΜΟΤΙΚΗ ΕΝΟΤΗΤΑ ΜΑΡΩΝΕΙΑΣ', 'code': '111010402'}
```
- Χρήση `try` για χειρισμό των λαθών
- Χρήση `format()` για μορφοποιημένες εκτυπώσεις.

Δες Format Cookbook

<https://mkaz.com/2012/10/10/python-string-format/>



# Ερωτήματα

- Ποιός είναι ο μικρότερος πληθυσμός δήμου;
- Ποιός είναι ο μέσος όρος του πληθυσμού των δημοτικών ενοτήτων;
- Ποιά είναι η τυπική απόκλιση του πληθυσμού όλων των δήμων;
- Ποιά είναι η διακύμανση και η τυπική απόκλιση του πληθυσμού των ατόμων με ηλικία μικρότερη από ή ίση από 59 ετών ανεξαρτήτου φύλου των δημοτικών ενοτήτων;

Συναρτήσεις



# Ορισμός συνάρτησης

```
>>> def is_prime(n):  
    if n<2:  
        return('no')  
    for x in range(2,n-1):  
        if (n%x)==0.0:  
            return('no')  
    return('yes')  
  
>>> for t in range(20):  
    print(t,is_prime(t))
```

```
0 no  
1 no  
2 yes  
3 yes  
4 no  
5 yes  
6 no  
7 yes  
8 no  
9 no  
10 no  
11 yes  
12 no  
13 yes  
14 no  
15 no  
16 no  
17 yes  
18 no  
19 yes
```



# Ορισμός συνάρτησης (συν.)

- `def` όνομα\_συνάρτησης(μεταβλητή) :
- `return` -> για επιστροφή τιμής
- κλήση: όνομα\_συνάρτησης (όρισμα)

Μαθηματικά προβλήματα για εξάσκηση:

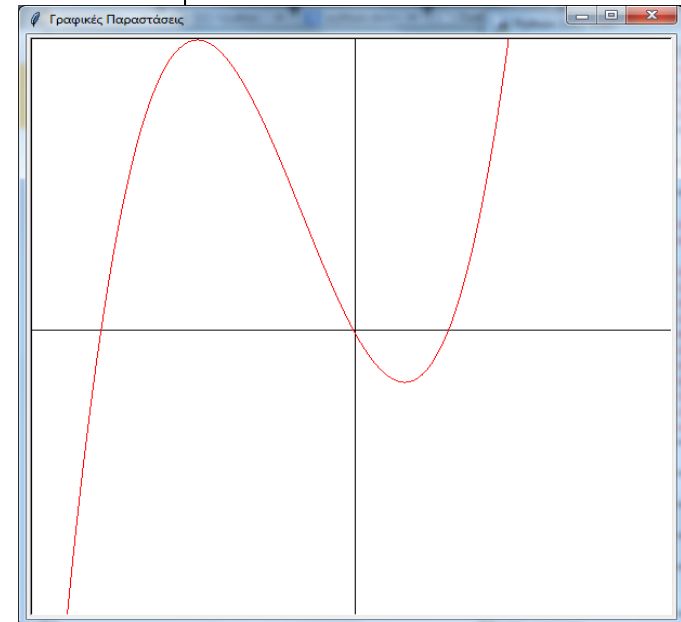
- <https://projecteuler.net>



Γραφικές παραστάσεις

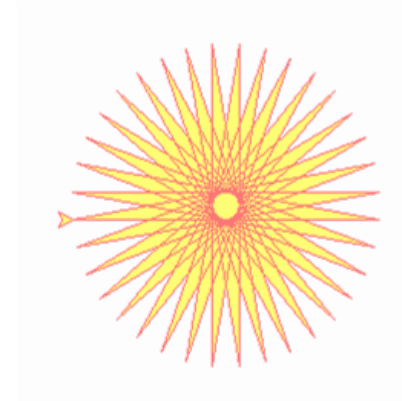
# turtle: Απλές γραφικές παραστάσεις (1)

```
from turtle import *
def f(x):
    return 0.0001*x**3+0.015*x**2-2*x-3
reset() # δημιουργία παραθύρου
speed('fastest') # ταχύτητα σχεδίασης
title('Γραφικές Παραστάσεις') # τίτλος παραθύρου
setup(600,600,0,0) # μέγεθος και θέση παραθύρου
penup() # η γραφίδα δεν ζωγραφίζει
# ο άξονας X
goto(-400,0) # πήγαινε στη θέση (-400,0)
pendown() # η γραφίδα ζωγραφίζει
goto(400,0) # πηγαίνοντας στη θέση (400,0)
# ο άξονας Y
penup()
goto(0,400)
pendown()
goto(0,-400)
penup()
# αλλαγή χρώματος
color('red')
goto(-300,f(-300))
down()
# η γραφίδα ζωγραφίζει συνεχώς στα σημεία [x, f(x)]
for x in range (-300,300):
    goto(x, f(x))
```



# turtle: Απλές γραφικές παραστάσεις (2)

```
from turtle import *  
color('red', 'yellow')  
begin_fill()  
while True:  
    forward(200)  
    left(170)  
    if abs(pos()) < 1:  
        break  
end_fill()  
done()
```

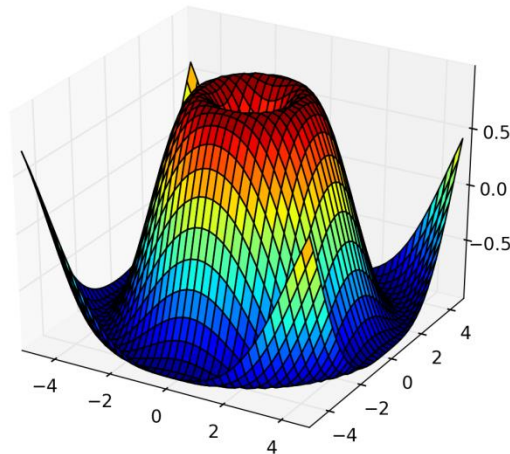


*Το αποτέλεσμα*



# Περισσότερες παραστάσεις...

Χρήση εξειδικευμένων βιβλιοθηκών με εξαιρετικές δυνατότητες, π.χ. Matplotlib



Γράφημα που δημιουργήθηκε μέσω της βιβλιοθήκης Matplotlib  
Πηγή :  
<https://commons.wikimedia.org/wiki/File:Matplotlib1.hires.png>



# Βιβλιογραφία

1. Εισαγωγή στους Υπολογιστές με τη Γλώσσα Python, Ν. Αβούρης - Κ. Σγάρμπας - Σ. Καξίρας - Μ. Κουκιάς - Β. Παλιουράς, 2013
2. A byte of Python (Ελληνικά)  
[http://dide.flo.sch.gr/Plinet/Meetings/Meeting23/A\\_Byte\\_of\\_Python-el.pdf](http://dide.flo.sch.gr/Plinet/Meetings/Meeting23/A_Byte_of_Python-el.pdf)
3. Python Schools, <http://www.pythonschool.net/>
4. Non-Programmer's Tutorial for Python 3,  
[http://en.wikibooks.org/wiki/Non-Programmer%27s\\_Tutorial\\_for\\_Python\\_3/Print\\_version](http://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3/Print_version)
5. Hands-on Python Tutorial, Dr. A. N. Harrington, Loyola Uni. Chicago  
<http://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/index.html>
6. Python Programming [http://en.wikibooks.org/wiki/Python\\_Programming](http://en.wikibooks.org/wiki/Python_Programming)
7. Python Documentation, <https://docs.python.org/3.4/contents.html>



Τέλος Υπο-ενότητας



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημειώματα

# Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0



# Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών, Μανώλης Τζαγκαράκης, Βικτωρία Δασκάλου, Σχολή Οργάνωσης και Διοίκησης Επιχειρήσεων, Τμήμα Οικονομικών Επιστημών. «Εισαγωγή στους Η/Υ και Εφαρμογές. Επεξεργασία δεδομένων με τη γλώσσα προγραμματισμού python». Έκδοση: 1.0. Πάτρα 2015. Διαθέσιμο από τη δικτυακή διεύθυνση:

<https://eclass.upatras.gr/courses/ECON1242/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.



# Σημείωμα Χρήσης Έργων Τρίτων

Το Έργο αυτό κάνει χρήση των ακόλουθων έργων:

## Εικόνες/Σχήματα/Διαγράμματα/Φωτογραφίες

Εικόνα 1: Guido van Rossum, By Doc Searls (2006oscon\_203.JPG) [CC BY-SA 2.0 (<http://creativecommons.org/licenses/by-sa/2.0>)], via Wikimedia Commons, Πηγή : [https://en.wikipedia.org/wiki/Guido\\_van\\_Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum)

Εικόνα 2: Οι 10 πιο διαδεδομένες γλώσσες προγραμματισμού, By TIOBE Software B.V. [CC BY-SA 4.0 (<http://creativecommons.org/licenses/by-sa/4.0>)], via Wikimedia Commons, Πηγή: <https://commons.wikimedia.org/wiki/File:Tiobeindex.png>

Εικόνα 3: Γράφημα που δημιουργήθηκε μέσω της βιβλιοθήκης Matplotlib, By <http://matplotlib.org/> (<http://matplotlib.org/>) [CC BY-SA 3.0 (<http://creativecommons.org/licenses/by-sa/3.0>)], via Wikimedia Commons, Πηγή : <https://commons.wikimedia.org/wiki/File:Matplotlib1.hires.png>

