



ΜΑΘΑΙΝΩ & ΔΙΔΑΣΚΩ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΜΕ PYTHON

Μάθημα 1ο και Μάθημα 2ο
Αράπογλου Αριστείδης – Δήμητρα Λέκκα

ΠΕΡΙΕΧΟΜΕΝΑ

2

- ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΡΥΤΗΘΝ
- ΒΑΣΙΚΟΙ ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ
- ΜΕΤΑΒΛΗΤΕΣ – ΤΕΛΕΣΤΕΣ- ΕΝΤΟΛΕΣ
ΕΙΣΟΔΟΥ/ΕΞΟΔΟΥ
- ΜΟΡΦΗ ΠΡΟΓΡΑΜΜΑΤΟΣ
- ΣΥΜΒΟΛΟΣΕΙΡΕΣ
- ΔΟΜΕΣ ΕΛΕΓΧΟΥ
- ΣΥΝΑΡΤΗΣΕΙΣ

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΡΥΤΗΘΝ

3

- ΑΠΛΗ
- ΕΥΚΟΛΗ ΣΤΗ ΜΑΘΗΣΗ
- ΑΝΟΙΚΤΟΥ ΚΩΔΙΚΑ
- ΥΨΗΛΟΥ ΕΠΙΠΕΔΟΥ(Script)
- ΑΝΕΞΑΡΤΗΤΗ ΛΕΙΤΟΥΡΓΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ
- ΔΙΕΡΜΗΝΕΥΟΜΕΝΗ
- ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ

Χαρακτηριστικά Python

1. ΑΠΛΗ

4

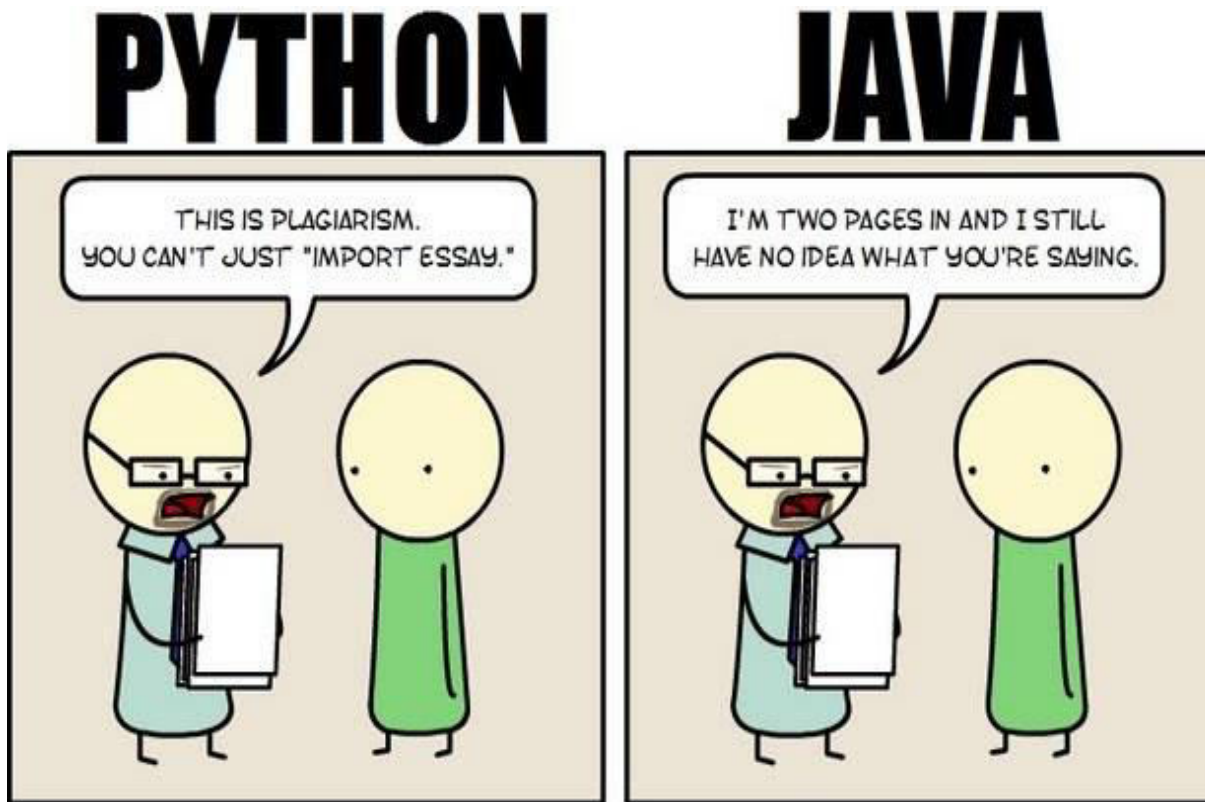


- Πρόκειται για μία ασυνήθιστα απλή στη σύνταξη γλώσσα που επιτρέπει να συγκεντρώνεται ο δημιουργός στη λύση του προβλήματος, αντί στη σύνταξη της ίδιας της γλώσσας.
- Τα προγράμματα γραμμένα σε Python είναι συντομότερα σε σχέση με άλλες μοντέρνες γλώσσες προγραμματισμού. Λιγότερος κώδικας σημαίνει λιγότερα λάθη.

Χαρακτηριστικά Python

1. ΑΠΛΗ

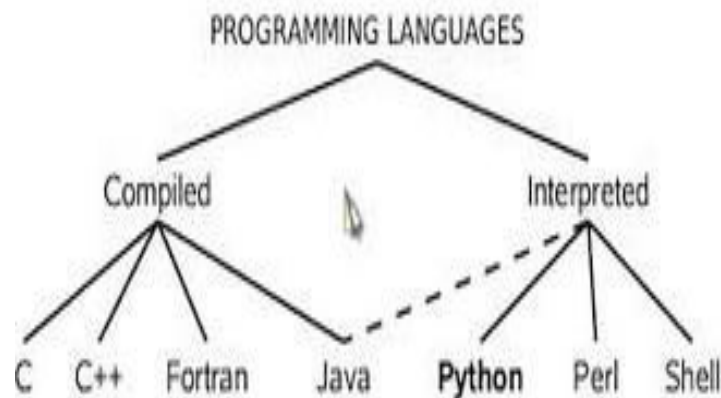
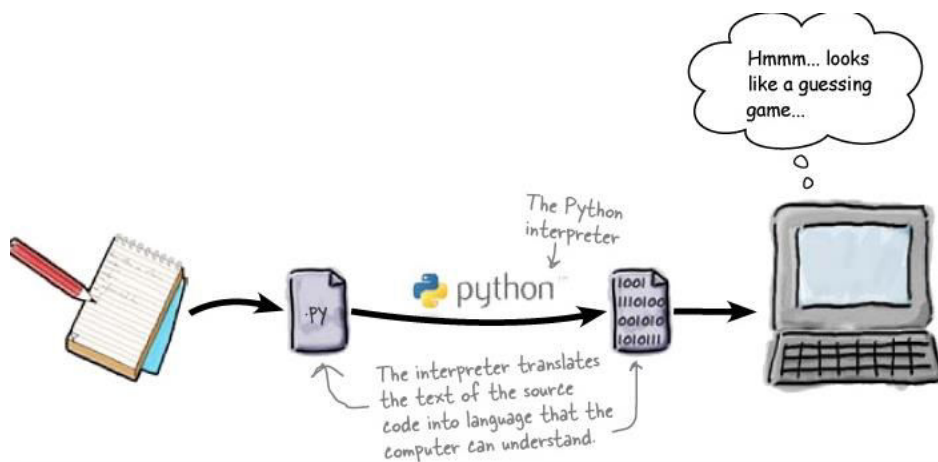
6



Χαρακτηριστικά Python

2. Χρησιμοποιεί Διερμηνευτή

7



Η Python χρησιμοποιεί διερμηνευτή (Interpreter)

Αυτό επιτρέπει μια άμεση διάδραση του κώδικα με τον προγραμματιστή, κάτι το οποίο είναι πολύ σημαντικό για μαθητές, που κάνουν τα πρώτα τους βήματα στον προγραμματισμό.

Χαρακτηριστικά Python

3.Ελεύθερη και Ανοικτού Κώδικα



8

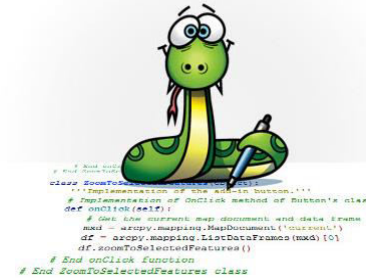
- Η Python είναι ένα Ελεύθερο Λογισμικό και Λογισμικό Ανοικτού Κώδικα (ΕΛ/ΛΑΚ) υποστηρίζοντας την ιδέα του διαμοιρασμού της γνώσης ανάμεσα σε κοινότητες.
- Η γλώσσα υποστηρίζεται και βελτιώνεται από μία ευρύτατη παγκόσμια κοινότητα.
- Για αυτό επεκτείνεται συνεχώς με πλούσιες βιβλιοθήκες προγραμματισμού (**modules**) για διάφορες εξειδικευμένες εφαρμογές.
- Πολλές διαδικτυακές εφαρμογές όπως το **Dropbox**, το **YouTube** και τμήματα της μηχανής **Google** έχουν γραφτεί σε Python.
- Σημαντική είναι επίσης η συνεισφορά της **ελληνικής** κοινότητας προγραμματιστών Python. Ήδη έχει ξεκινήσει να συμβάλει με ανοικτά έργα (project) για την εκπαιδευτική κοινότητα και στο μέλλον πιστεύουμε σε μία μεγαλύτερη συμβολή της μεταξύ των οποίων μπορεί να είναι ο πλήρης εξελληνισμός της.

Χαρακτηριστικά Python

4.Γλώσσα Υψηλού επιπέδου

9

- Η Python είναι μία γλώσσα προγραμματισμού **υψηλού επιπέδου**.



- Παρέχει εντυπωσιακές εναλλακτικές δυνατότητες έκφρασης για τη διαχείριση τύπων δεδομένων και μεταβλητών, που διευκολύνουν σε αρκετές περιπτώσεις στον προγραμματισμό.
 - Για παράδειγμα μία **μεταβλητή** μπορεί να χρησιμοποιηθεί **δυναμικά** μέσα σε ένα πρόγραμμα και να δείχνει σε τιμές με διαφορετικό τύπο δεδομένων (ακέραιες, κινητής υποδιαστολής, συμβολοσειρές κ.ά.).
- **Σημείωση:** Η ποικιλία διαφορετικών εκφράσεων σε συμβολικό επίπεδο μπορεί να αποτελέσουν εμπόδια για τον αρχάριο προγραμματιστή ως προς την κατανόηση βασικών προγραμματιστικών εννοιών, γι αυτό προτείνεται να εισαχθούν από τον εκπαιδευτικό σταδιακά.



Χαρακτηριστικά Python 5.Φορητή

Η Python, λόγω του ανοικτού της κώδικα, έχει υλοποιήσεις που μπορούν να λειτουργήσουν σε πολλές πλατφόρμες λειτουργικών συστημάτων και διαφορετικών συσκευών (PC, Raspberry, smart phone, tablet, arduino).

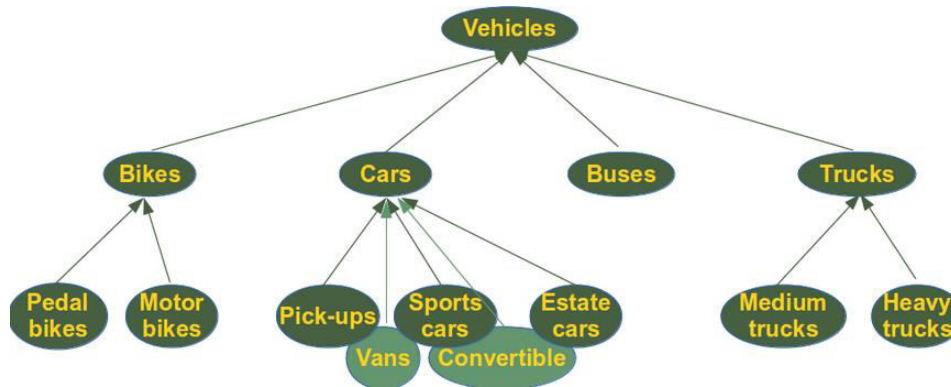
Τα προγράμμάτα μπορούν να λειτουργήσουν σε οποιαδήποτε από αυτές τις πλατφόρμες (Linux, Windows, FreeBSD κ.λπ), χωρίς να χρειάζονται καθόλου αλλαγές, αποφεύγοντας βέβαια τη χρήση χαρακτηριστικών που εξαρτώνται από κάθε σύστημα.



Χαρακτηριστικά Python

6. Αντικειμενοστρεφής

11

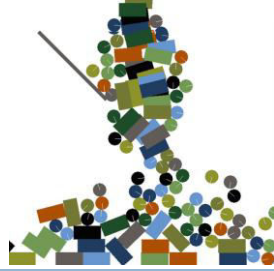


Η Python υποστηρίζει **δομημένο, συναρτησιακό** και **αντικειμενοστρεφή** προγραμματισμό (**object-oriented**).

Διαθέτει έναν πολύ ισχυρό, αλλά συγχρόνως απλό τρόπο για ανάπτυξη προγραμμάτων με αντικείμενα, ειδικά όταν συγκρίνεται με γλώσσες όπως η C++ και η Java.

Χαρακτηριστικά Python

7.Εκτεταμένες βιβλιοθήκες



12

- Η Python υποστηρίζει μια εκτεταμένη standard βιβλιοθήκη, η οποία περιλαμβάνει τμήματα για προγραμματισμό γραφικών, αριθμητική επεξεργασία και web συνδεσιμότητα.
- Επιπρόσθετα υπάρχει μία τεράστια συλλογή από βιβλιοθήκες που καλύπτουν κάθε πιθανή χρήση μιας γλώσσας προγραμματισμού, από γραφικές διεπαφές χρήστη (GUI-graphical user interfaces) μέχρι βάσεις δεδομένων.

PYTHON

13

- Τα προγράμματα αποτελούνται από modules
- Υπάρχει μία βιβλιοθήκη με standard modules, αλλά μπορεί να δημιουργηθούν και από το χρήστη ως αρχεία .py
- Τα modules περιέχουν ορισμούς (definitions) και εντολές-δηλώσεις (statements)
- Οι εντολές-δηλώσεις (Statements) περιέχουν εκφράσεις (expressions)
- Οι εκφράσεις δημιουργούν και επεξεργάζονται αντικείμενα (objects)

ΔΙΕΡΜΗΝΕΥΤΗΣ ΤΗΣ PYTHON

14

- Διερμηνευόμενη γλώσσα
- ο διερμηνευτής προσφέρει ένα περιβάλλον διεπαφής
- Το αποτέλεσμα των εκφράσεων εμφανίζεται στην οθόνη

```
>>> 3 + 7
10
>>> 3 < 15
True
>>> 'print me'
'print me'
>>> print 'print me'
print me
>>>
```

ΠΡΑΞΕΙΣ ΜΕ ΜΕΤΑΒΛΗΤΕΣ

15

□ ΑΡΙΘΜΗΤΙΚΕΣ

Βασικοί Τελεστές Πράξεων

+, -, *, /, **, //, %, <<, >>

Τελεστές Σύγκρισης

<, >, ==, <=, >=, !=

□ ΑΛΦΑΡΙΘΜΗΤΙΚΕΣ

+

Τελεστές Σύγκρισης: <, >, ==, <=, >=, !=

□ ΛΟΓΙΚΕΣ

Τελεστές λογικών πράξεων: not, and, or

Για περισσότερα:

http://www.tutorialspoint.com/python/python_basic_operators.htm

ΜΕΤΑΒΛΗΤΕΣ

16

Η γλώσσα Python παρέχει εντυπωσιακές εναλλακτικές δυνατότητες έκφρασης για τη διαχείριση μεταβλητών, που διευκολύνουν τον προγραμματιστή.

- Δε δηλώνονται, απλά εκχωρούνται.
- Η μεταβλητή δημιουργείται την πρώτη φορά που της εκχωρούμε τιμή.
- Μπορούμε να εκχωρήσουμε διαφορετικούς τύπους τιμών (ακέραιες, κινητής υποδιαστολής, συμβολοσειρές, κ.ά.).
- Είναι αναφορές (δείκτες) σε αντικείμενα (τιμές).
- Ο τύπος χαρακτηρίζει το αντικείμενο και όχι τη μεταβλητή.

ΜΕΤΑΒΛΗΤΕΣ

17

- ΤΑ ΟΝΟΜΑΤΑ ΤΩΝ ΜΕΤΑΒΛΗΤΩΝ ΠΕΡΙΕΧΟΥΝ
 - ▣ ΤΑ ΓΡΑΜΜΑΤΑ ΤΟΥ **ΑΓΓΛΙΚΟΥ** ΑΛΦΑΒΗΤΟΥ (ΠΕΖΑ - ΚΕΦΑΛΑΙΑ)
 - ▣ ΤΟ ΧΑΡΑΚΤΗΡΑ “ _ “
 - ▣ ΤΑ ΨΗΦΙΑ 0,1,2,3,4,5,6,7,8,9

- ▣ ΤΟ ΟΝΟΜΑ ΠΡΕΠΕΙ ΝΑ ΑΡΧΙΖΕΙ ΜΕ ΓΡΑΜΜΑ Η’ “ _ “
- ▣ Θα πρέπει να είμαστε βέβαιοι ότι το όνομα δε είναι όμοιο με κάποια όνομα ενσωματωμένης συνάρτησης ή εντολής.
- ▣ Οι χαρακτήρες στα ονόματα μεταβλητών είναι **case sensitive**

ΜΕΤΑΒΛΗΤΕΣ

18

Όλα είναι αντικείμενα

- Ακόμα και οι συναρτήσεις και οι κλάσεις
- Ο τύπος δεδομένων είναι ιδιότητα του αντικειμένου και όχι της μεταβλητής

```
>>> x = 7
>>> x
7
>>> x = 'hello'
>>> x
'hello'
>>>
```

Μεταβλητές και τελεστές (λογικοί-αριθμητικοί)

19

□ Δραστηριότητα

Ακολουθείστε τα παρακάτω στο περιβάλλον της Python:

- Να γράψετε μια απλή γραμμή σχολίων της αρεσκείας σας (προσοχή να ξεκινάει με #).
- Να οριστεί η μεταβλητή `arithmos`, που να περιέχει την τιμή `1.234` (αριθμός κινητής υποδιαστολής).
- Να οριστεί η μεταβλητή `arithmos_tetragwno`, που να περιέχει το τετράγωνο της `arithmos`.
- Να εμφανιστούν στην οθόνη οι τιμές των μεταβλητών `arithmos`, `arithmos_tetragwno`.
- Να οριστεί η μεταβλητή `logikh`, που να περιέχει τη τιμή `True`.
- Να οριστεί η μεταβλητή `paixnidia`, που να περιέχει τη τιμή `10`.
- Να γίνουν λογικοί έλεγχοι για το περιεχόμενο της μεταβλητής `paixnidia`, με χρήση των λογικών τελεστών σύγκρισης σε συνδυασμό με τους τελεστές λογικών πράξεων (`paixnidia == 8`, `paixnidia > 8`, `paixnidia == 8 or paixnidia == 10`, `paixnidia == 9 and paixnidia == 10`).

Μεταβλητές και τελεστές (λογικοί-αριθμητικοί)

20

Επαληθεύστε τις προσπάθειες σας με τις παρακάτω απαντήσεις.

```
>>> # ύψωσε σε δύναμη
```

```
>>> arithmos = 1.234
```

```
>>> arithmos_tetragwno = arithmos ** 2
```

```
>>> print arithmos, arithmos_tetragwno 
```

Αποτέλεσμα στην οθόνη: 1.234 1.522756

```
>>> logikh = True
```

```
>>> print logikh
```

Αποτέλεσμα στην οθόνη: True

Μεταβλητές και τελεστές (λογικοί-αριθμητικοί)

21

```
>>> παιχνidia = 10
```

```
>>> παιχνidia == 8
```

```
False
```

```
>>> παιχνidia == 8 or παιχνidia == 10
```

```
True
```

```
>>> not παιχνidia == 8
```

```
True
```

```
>>> παιχνidia >= 8
```

```
True
```

```
>>> παιχνidia != 8
```

```
True
```

```
>>> παιχνidia <= 8
```

```
False
```

```
>>>
```

ΜΟΡΦΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

22

□ ΕΝΤΟΛΕΣ

- ▣ ΑΝ ΒΡΙΣΚΟΝΤΑΙ ΣΤΗΝ ΙΔΙΑ ΓΡΑΜΜΗ ΔΙΑΧΩΡΙΖΟΝΤΑΙ ΜΕ “;”.
- ▣ ΑΝ ΒΡΙΣΚΟΝΤΑΙ ΣΕ ΔΙΑΦΟΡΕΤΙΚΗ ΓΡΑΜΜΗ ΤΟ “;” ΜΠΟΡΕΙ ΝΑ ΠΑΡΑΛΕΙΦΘΕΙ.

ΜΟΡΦΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

23

- ❑ ΟΙ ΛΕΥΚΟΙ ΧΑΡΑΚΤΗΡΕΣ (ΚΕΝΑ ΚΑΙ TAB) ΣΤΗΝ ΑΡΧΗ ΤΩΝ ΓΡΑΜΜΩΝ ΟΜΑΔΟΠΟΙΟΥΝ ΤΙΣ ΕΝΤΟΛΕΣ ΣΕ ΟΜΑΔΕΣ ΕΝΤΟΛΩΝ (ΜΠΛΟΚ).
- ❑ ΤΟ ΣΥΜΒΟΛΟ # ΧΡΗΣΙΜΟΠΟΙΕΙΤΑΙ ΓΙΑ ΤΗΝ ΕΙΣΑΓΩΓΗ ΣΧΟΛΙΩΝ.

ΑΡΙΘΜΟΙ: ΑΚΕΡΑΙΟΙ

24

- Ακέραιος– ισοδύναμο με τον long της C
- Long Integer – an unbounded integer value.

```
>>> 132224
132224
>>> 132323 ** 2
17509376329L
>>>
```

ΑΡΙΘΜΟΙ: ΠΡΑΓΜΑΤΙΚΟΙ

25

- `int(x)` μετατρέπει το `x` σε ακέραιο
- `float(x)` μετατρέπει το `x` σε floating point
- `Type(x)` μας επιστρέφει το τύπο δεδομένων

```
>>> print 1.23232
1.23232
>>> 1.3E7
13000000.0
>>> int(2.0)
2
>>> float(2)
2.0
>>> x = 23
>>> type(x)
<type 'int'>
```

ΑΡΙΘΜΟΙ: ΜΙΓΑΔΙΚΟΙ

26

- Ενσωματωμένοι στην Python
- Ίδιες πράξεις με ακεραίους και πραγματικούς

```
>>> x = 3 + 2j
>>> y = -1j
>>> x + y
(3+1j)
>>> x * y
(2-3j)
```

Παραδείγματα μεταβλητές

27

```
>>> x = 12
>>> y = 23
>>> type (y)
>>> y = x / 10
>>> print y
```

<type 'int'>

1

```
>>> x = 12
>>> y = 12 / 10.0
>>> print y
>>> type (y)
```

1.2

<type 'float'>

Περισσότερα για τις μεταβλητές

Δραστηριότητα: Αντιμετάθεση

28

Δραστηριότητα Αντιμετάθεσης στην Python

```
x = 30  
y = 45  
print x, y  
z = y  
y = x  
x = z  
print x ,y
```

```
>>>
```

```
30 45
```

```
45 30
```

```
x = 30  
y = 45  
print x , y  
x , y = y , x  
print x , y
```

```
30 45
```

```
45 30
```

Περισσότερα για τις μεταβλητές

Δραστηριότητα: Ταυτότητα αντικειμένου

29

#Διαχείριση μεταβλητών

x = 12

print x

print id(x)

y = x

print y

print id(y)

y = 24

print y

print id(y)

print x

print id(x)

z=x=y

print x, y, z

print id(x), id(y), id(z)

print id(24)

>>> id (10)

27285668

>>> id (10.0)

40132376

>>>

>>>

12

20512132

12

20512132

24

20511988

12

20512132

24 24 24

20511988 20511988 20511988

20511988

>>>

ΑΡΙΘΜΟΙ: ΕΙΝΑΙ ΑΜΕΤΑΒΛΗΤΟΙ

30

```
>>> x = 4.5
```

```
>>> y = x
```

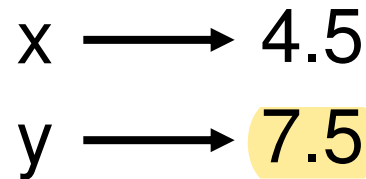
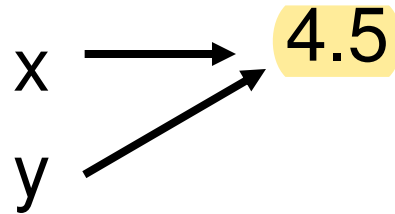
```
>>> y += 3
```

```
>>> x
```

```
4.5
```

```
>>> y
```

```
7.5
```



ΕΝΤΟΛΕΣ-Δηλώσεις (Statements)

31

- = π.χ. `x = 5`
 - ΑΝΑΘΕΣΗ ΤΙΜΗΣ (το `x` δείχνει στη τιμή-αντικείμενο 5)
 - $x \rightarrow 5$

- **Print**
 - ΕΜΦΑΝΙΖΕΙ ΣΤΗΝ ΟΘΟΝΗ

```
>>> y = 5
>>> y += 3
>>> print y
8
```

```
>>>  print "hello"
hello
>>> print "hello"*2
hellohello
```

```
print “αρχή”,
print “τέλος”
```

print evaluates each expression in turn and writes the resulting object to standard output. If an object is not a string, it is first converted to a string using the rules for string conversions.

ΕΝΤΟΛΕΣ-Δηλώσεις (Statements)

32

- Input/ Raw_input
 - ▣ ΔΙΑΒΑΖΕΙ ΑΠΟ ΤΟ ΠΛΗΚΤΡΟΛΟΓΙΟ

```
>>> x=input ('Δώσε μου το όνομα : ')\nΔώσε μου το όνομα : αρης
```

```
Traceback (most recent call last):\n  File "<pyshell#62>", line 1, in <module>\n    x= input ('Δώσε μου το όνομα : ')\n  File "<string>", line 1\n    αρης\n    | ^\nSyntaxError: invalid syntax\n>>>
```

- ▣ >>> x = raw_input ('Δώσε μου το όνομα σου : ')\nΔώσε μου το όνομα σου : Αρης
- ▣ >>> print x
- ▣ Αρης

ΣΥΜΒΟΛΟΣΕΙΡΕΣ

33

- Οι συμβολοσειρές είναι αμετάβλητες
- Δεν υπάρχει τύπος “char” όπως στη C++ ή Java
- υπερφόρτωση τελεστή + : κάνει συνένωση

```
>>> s="today"
```

```
>>> s[0] = "s"
```

Traceback (most recent call last):

File "<pyshell#17>", line 1, in <module>
s[0] = "s "

TypeError: 'str' object does not support
item assignment

```
>>> x = 'hello'
```

```
>>> x = x + ' there'
```

```
>>> x
```

```
'hello there'
```

ΣΥΜΒΟΛΟΣΕΙΡΕΣ

34

- Μπορούμε να χρησιμοποιούμε μονά, διπλά ή τρία διπλά εισαγωγικά

```
>>> 'Είμαι μια συμβολοσειρά'  
'Είμαι μια συμβολοσειρά'
```

```
>>> " Το ίδιο κι εγώ!"  
' Το ίδιο κι εγώ'
```

```
>>> s = """ Κι εγώ επίσης!  
Παρόλο που είμαι αρκετά μεγαλύτερη  
Από τις άλλες:) """  
'Κι εγώ επίσης!\nπαρότι είμαι αρκετά  
μεγαλύτερη\nαπό τις άλλες:).'
```

```
>>> print s  
Κι εγώ επίσης!  
Παρότι είμαι αρκετά μεγαλύτερη  
Από τις άλλες:)
```

ΣΥΜΒΟΛΟΣΕΙΡΕΣ

35

```
>>> s = '012345'
>>> s[3]
'3'
>>> s[1:4]
'123'
>>> s[2:]
'2345'
>>> s[:4]
'0123'
>>> s[-2]
'4'
>>> 'η' in 'πηγή'
True
```

- **len**(String) – επιστρέφει το πλήθος των χαρακτήρων της συμβολοσειράς
- **str**(Object) – επιστρέφει μια αναπαράσταση του αντικειμένου σε συμβολοσειρά

```
>>> len(s)
6
>>> str(10.3)
'10.3'
```

ΣΥΜΒΟΛΟΣΕΙΡΕΣ - formatting

36

- Παρόμοιο με της C
- `<formatted string> % <elements to insert>`
- Μπορούμε να χρησιμοποιούμε μόνο το `%s`, θα μετατρέπει το αντικείμενο σε συμβολοσειρά.

```
>>> "One, %d, three" % 2
'One, 2, three'
>>> "%d, two, %s" % (1,3)
'1, two, 3'
>>> "%s two %s" % (1, 'three')
'1 two three'
>>>
```

Μάθημα 2ο: Δομή Επιλογής

37

If <συνθήκη ελέγχου>:

#γράψε τις εντολές της if εδώ

εντολές

Παράδειγμα

#πρόγραμμα εμφάνισης της απόλυτης τιμής ενός ακεραίου αριθμού

```
a=int(input('Δώσε ένα ακέραιο αριθμό '))
```

```
if a<=0:
```

```
    a=(-1)*a
```

```
print a
```

ΕΣΟΧΕΣ (INDENTATION)

38

- Στην Python χρησιμοποιούνται εσοχές για να ορίσουμε την **εμβέλεια** των εντολών
- Όλες οι γραμμές πρέπει να έχουν την ίδια εσοχή για να έχουν την ίδια εμβέλεια ή μεγαλύτερη εσοχή αν είναι **εμφωλευμένες**
- Αυτό παρακινεί τον προγραμματιστή να στοιχίζει σωστά τον κώδικά του αφού η **στοίχιση** είναι μέρος του προγράμματος!

Μάθημα 2ο: Δομή Επιλογής

39

Η εντολή ελέγχου **ΑΝ_ΑΛΛΙΩΣ** συντάσσεται ως:

```
if <συνθήκη ελέγχου>:  
    #γράψε τις εντολές εδώ  
else:  
    #γράψε τις εντολές εδώ
```

Σημείωση: Οι ομάδες εντολών που θα εκτελεστούν, αν ισχύει μια συνθήκη, ορίζονται ως ένα μπλοκ με εσοχή (κενά διαστήματα) βάζοντας τη μία εντολή κάτω από την άλλη. Δεν πρέπει να διαγράψουμε τα κενά αυτά διαστήματα.

Η Python προσφέρει τη δυνατότητα για σύνταξη σύνθετων δομών επιλογής με τη χρήση της εντολής **elif**. Η σύνταξη είναι ως εξής:

```
if <συνθήκη>:  
    <εντολές>  
elif <συνθήκη2>:  
    <εντολές_2>  
else:  
    <εντολές_3>
```

ΕΝΤΟΛΕΣ – if ... elif

40

```
x = 30
if x <= 15 :
    y = x + 15
elif x <= 30 :
    y = x + 30
else :
    y = x
print 'y = ',y
```

Μάθημα 2ο: Δομή Επιλογής

41

Δραστηριότητα 2.2.4

Να γραφεί πρόγραμμα σε Python που να διαβάσει τη θερμοκρασία ενός δωματίου και αν η θερμοκρασία είναι ίση ή μεγαλύτερη από 32 βαθμούς να εμφανίζει μήνυμα για να ανοίξει η ψύξη του κλιματιστικού, αν είναι μικρότερη ή ίση από 16 να εμφανίζει μήνυμα για να ανοίξει η θέρμανση, διαφορετικά να εμφανίζει μήνυμα ότι δεν χρειάζεται να ανοίξει το κλιματιστικό.

```
temperature = int(input("Ποια είναι η θερμοκρασία του δωματίου σε βαθμούς Κελσίου; "))
if temperature >= 32:
    print("Άνοιξε τη ψύξη του κλιματιστικού και ρύθμισε σωστά τη θερμοκρασία")
elif temperature <= 16:
    print("Άνοιξε τη θέρμανση του κλιματιστικού και ρύθμισε για οικονομία τη θερμοκρασία")
else:
    print("Η θερμοκρασία είναι καλή δε χρειάζεται ακόμη να ανοίξεις το κλιματιστικό")
print("τέλος")
```

Μάθημα 2ο: Δομή Επανάληψης

42

Οι **βρόχοι for** εκτελούνται για συγκεκριμένο πλήθος φορές. Για τη δημιουργία της χρησιμοποιείται η συνάρτηση **range()**.

- `for onoma_metavlitis in range (αρχή, μέχρι, βήμα):`
- Εντολή1
- Εντολή2
-
- Εντολήν

Δραστηριότητα: Εξοικείωση με τη Δομή Επανάληψης For

Να γραφεί πρόγραμμα που να εκτυπώνει πέντε (5) φορές το μήνυμα «Θέλει αρετή και τόλμη η ελευθερία!»

```
for metritis in range (5):  
    print "Θέλει αρετή και τόλμη η ελευθερία!"
```

Μάθημα 2ο: Δομή Επανάληψης

Προσεγγίζοντας τη συνάρτηση range

43

- Η range() είναι μια ενσωματωμένη συνάρτηση της γλώσσας Python, η οποία, ανάμεσα σε άλλα, χρησιμοποιείται για την υπόδειξη του αριθμού των επαναλήψεων που θα εκτελεστούν σε ένα βρόχο.
- Η δομή της συνάρτησης range είναι της μορφής (αρχή, μέχρι, βήμα), όπου αρχή, μέχρι, βήμα ακέραιοι αριθμοί. Οι ενδείξεις της «αρχής» και του «βήματος» δεν είναι υποχρεωτικές, αλλά μπορεί να προστεθούν, ώστε η επανάληψη να μοιάζει με τη σύνταξη του ψευδοκώδικα. Αντίθετα η ένδειξη «μέχρι» πρέπει πάντα να αναφέρεται.

Μάθημα 2ο: Δομή Επανάληψης

Προσεγγίζοντας τη συνάρτηση `range`

44

□ Δραστηριότητα

Πειραματιστείτε με τη χρήση της συνάρτησης `range` στο προγραμματιστικό περιβάλλον της Python για να δείτε τι παράγει η συνάρτηση.

Για παράδειγμα:

- `range(10)` παράγει τη λίστα: `[0,1,2,3,4,5,6,7,8,9]`.
- `range(1, 8)` παράγει τη λίστα: `[1,2,3,4,5,7]`
- `range(0, 35, 5)` παράγει τη λίστα: `[0,5,10,15,20,25,30]`
- `range(8, -1, -1)` παράγει τη λίστα `[8, 7, 6, 5, 4, 3, 2, 1, 0]`

ΕΝΤΟΛΕΣ – for

45

```
for i in range (0, 10, 2):  
    print i
```

0
2
4
6
8

ΕΝΤΟΛΕΣ – for

46

```
for x in range(5) :  
    print x
```

```
0  
1  
2  
3  
4
```

```
for x in range(5):  
    print x  
    break  
else :  
    print 'i got here'
```

```
0
```

Η range(n) δημιουργεί μια λίστα με τους αριθμούς από 0 μέχρι και n-1

ΕΝΤΟΛΕΣ

47

□ break

- ▣ Βγαίνει από το πιο κοντινό for ή while στο οποίο περιέχεται.

□ continue

- ▣ Πηγαίνει πάλι στην αρχή του πιο κοντινού for ή while στο οποίο περιέχεται.

□ pass

- ▣ Δεν κάνει τίποτα, είναι μια κενή εντολή (empty statement)

Μάθημα 2ο: Δομή Επανάληψης

Ο βρόχος **while**

48

Αρχική τιμή μεταβλητής

While ονομα_μεταβλητής <συνθήκη>:

Εντολή1

Εντολή2

....

Εντολή_n

ΕΝΤΟΛΕΣ –while

49

```
x = 1
while x < 3 :
    print x
    x = x + 1
else:
    print 'hello'
```

```
1
2
hello
```

```
x = 1
while x < 5 :
    print x
    x = x + 1
    break
else :
    print 'i got here'
```

```
1
```

Μάθημα 2ο: Δομή Επιλογής και Επανάληψης-Δραστηριότητες

50

Δραστηριότητα 2.2.2: Στοιχίστε σωστά τις παρακάτω εντολές έτσι ώστε:

- το μήνυμα ""αριθμός διαφορετικός από το μηδέν"" να εμφανίζεται κάθε φορά που δίνεται αριθμός διαφορετικός από το μηδέν
- το μήνυμα "θετικός" να εμφανίζεται κάθε φορά που δίνεται θετικός αριθμός
- το μήνυμα "Δώσε ακέραιο:" να εμφανίζεται κάθε φορά που θέλουμε να διαβάσουμε έναν ακέραιο
- το μήνυμα "Τέλος" να εμφανίζεται όταν δοθεί το 0.

```
x=input("Δώσε ακέραιο:")
while x != 0:
    print "αριθμός διαφορετικός από το μηδέν"
    if x>0:
        print "θετικός"
    x=input("Δώσε ακέραιο:")
    print "Τέλος"
```

```
x=input("Δώσε ακέραιο:")
while x != 0:
    print "αριθμός διαφορετικός από το μηδέν"
    if x>0:
        print "θετικός"
    x=input("Δώσε ακέραιο:")
    print "Τέλος"
```

Μάθημα 2ο: Δομή Επιλογής και Επανάληψης

51

Κώδικας (δραστηριότητα 2.2.3)	Τι θα εμφανιστεί
<pre>num = 10 while num > 3: print num num = num-1</pre>	
<pre>divisor=2 for i in range (0, 10, 2): print i/divisor</pre>	
<pre>a=1 while a < 3: if a % 2 == 0: b=1 while b < 3: print "X" b=b+1 print "O" a=a+1</pre>	
<pre>a = 2 b = 3 for i in range(a): for j in range(b): print "πολύ"*i + "νόστιμο "*j</pre>	

Μάθημα 2ο: Δομή Επιλογής και Επανάληψης

52

```
num = 10
while num > 3:
    print num
    num = num-1
```

```
>>>
10
9
8
7
6
5
4
>>>
```

```
divisor=2
for i in range (0, 10, 2):
    print i/divisor
```

```
>>>
0
1
2
3
4
>>>
```

Μάθημα 2ο: Δομή Επιλογής και Επανάληψης-Δραστηριότητες

53

```
a=1
while a < 3:
    if a % 2 == 0:
        b=1
        while b < 3:
            print "X"
            b=b+1
        print "O"
    a=a+1
```

```
>>>
O
X
X
O
>>>
```

```
a = 2
b = 3
for i in range(a):
    for j in range(b):
        print "πολύ"*i + "νόστιμο "*j
```

```
>>>
νόστιμο
νόστιμο νόστιμο
πολύ
πολύνόστιμο
πολύνόστιμο νόστιμο
>>>
```

```
count=0
for letter in 'Χιόνι!':
    print 'Το γράμμα #',count,'είναι',letter
    count+=1
```

```
Το γράμμα # 0 είναι Χ
Το γράμμα # 1 είναι ι
Το γράμμα # 2 είναι ό
Το γράμμα # 3 είναι ν
Το γράμμα # 4 είναι ι
Το γράμμα # 5 είναι !
```

ΣΥΝΑΡΤΗΣΕΙΣ

54

- `def` όνομα_συνάρτησης (παράμετροι): Μπλοκ εντολών
- ΟΙ ΠΑΡΑΜΕΤΡΟΙ ΜΠΟΡΟΥΝ ΚΑΙ ΝΑ ΜΗΝ ΥΠΑΡΧΟΥΝ
- ΑΝ ΤΟ Μπλοκ εντολών ΔΕΝ ΥΠΑΡΧΕΙ ΠΡΕΠΕΙ ΝΑ ΒΑΛΟΥΜΕ ΤΗΝ ΛΕΞΗ `pass`
- ΥΠΟΣΤΗΡΙΖΕΤΑΙ ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ
- ΓΙΑ ΝΑ ΕΠΙΣΤΡΕΨΟΥΜΕ ΜΙΑ ΤΙΜΗ ΑΠΟ ΣΥΝΑΡΤΗΣΗ ΧΡΗΣΙΜΟΠΟΙΟΥΜΕ ΤΗΝ ΕΝΤΟΛΗ `return.`

ΣΥΝΑΡΤΗΣΕΙΣ- παράδειγμα 1

55

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

```
>>> max(3,5)  
5  
>>> max('hello', 'there')  
'there'  
>>> max(3, 'hello')  
'hello'
```

ΣΥΝΑΡΤΗΣΕΙΣ- παράδειγμα 2

56

Οι συναρτήσεις σαν παράμετροι

```
def foo(f, a) :  
    return f(a)
```

```
def bar(x) :  
    return x * x
```

```
>>> foo(bar, 3)  
9
```

```
def emphasize (phrase) :  
    return 'πολύ ' + phrase
```

```
statement = 'κρύο'  
print emphasize (statement )
```

```
print 'Στην Αλάσκα κάνει ' +  
    emphasize (emphasize (statement ) )
```

πολύ κρύο

Στην Αλάσκα κάνει πολύ πολύ κρύο

ΣΥΝΑΡΤΗΣΕΙΣ- Παράμετροι

57

- Στις παραμέτρους μπορώ να εκχωρήσω προεπιλεγμένη τιμή
- Αυτές οι τιμές παρακάμπτονται αν δοθεί άλλη τιμή
- Ο τύπος της προεπιλεγμένης τιμής δεν μας περιορίζει

```
>>> def foo(x = 3) :  
...     print x  
...  
>>> foo()  
3  
>>> foo(10)  
10  
>>> foo('hello')  
hello
```

ΣΥΝΑΡΤΗΣΕΙΣ- Παράμετροι

58

- Κλήση με όνομα ή με θέση
- Όλες η παράμετροι με θέση προηγούνται των ονομάτων

```
>>> def foo (a,b,c) :  
...     print a, b, c  
...  
>>> foo(c = 10, a = 2, b =  
14)  
2 14 10  
>>> foo(3, c = 2, b = 19)  
3 19 2
```

ΣΥΝΑΡΤΗΣΕΙΣ- άγνωστος αριθμός παραμέτρων

59

```
>>> def concat(sep, *args):  
    return sep.join(args)  
  
...  
>>> concat('/', "Δευτέρα", "Τρίτη", "Τετάρτη")  
'Δευτέρα/Τρίτη/Τετάρτη'  
>>>  
>>> concat('.', " Δευτέρα ", " Τρίτη ", " Τετάρτη ")  
' Δευτέρα.Τρίτη.Τετάρτη '
```

ΣΥΝΑΡΤΗΣΕΙΣ- Παράμετροι

60

- Συμβολοσειρά ως παράμετρος
- Παραμένει αμετάβλητη

```
def f(s):  
    s=s+"άνθρωπος"  
    return s  
  
b="υπερ"  
f(b)  
print b, ",",  
  
b = f(b)  
print b
```

υπερ , υπεράνθρωπος

ΕΡΩΤΗΣΕΙΣ

61

