Προγραμματισμός Η/Υ / python

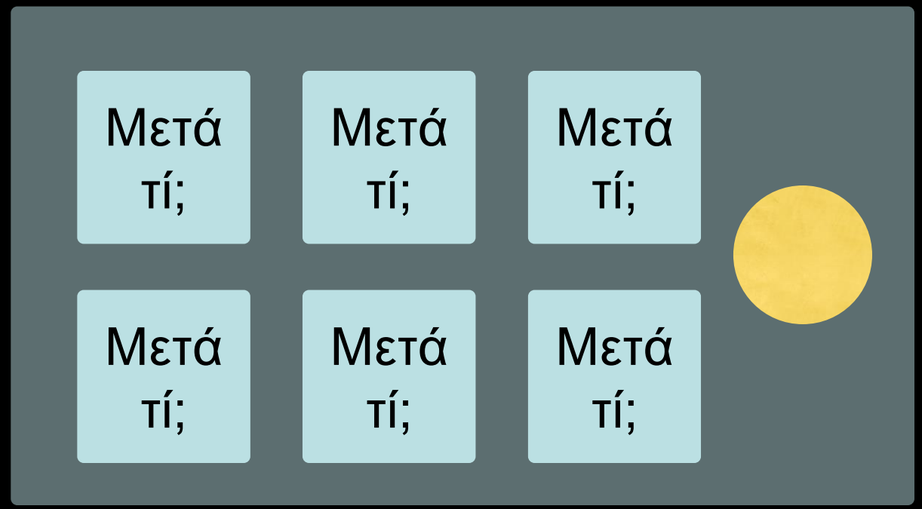
1

Εισαγωγή

Οι Υπολογιστές είναι χρήσιμοι...

- Οι υπολογιστές κατασκευάζονται για έναν σκοπό – για να κάνουν πράγματα (**υπολογισμούς**) για εμάς.
- Αλλά πρέπει να τους μιλήσουμε στη γλώσσα τους προκειμένου να τους περιγράψουμε τι θέλουμε να συμβεί.
- Για τους απλούς χρήστες είναι πολύ εύκολο – κάποιος άλλος έγραψε τα προγράμματα (οδηγίες) στον υπολογιστή και οι χρήστες επιλέγουν απλά αυτό που χρειάζονται.

Μετά τι;



Οι Προγραμματιστές Η/Υ είναι σε υψηλή ζήτηση

- Οι εφαρμογές iPhone είναι μια αγορά εκατοντάδων εκατομμυρίων $$$
- Οι εφαρμογές iPhone έχουν πάνω από 3 δισεκατομμύρια λήψεις
- Αρκετοί παραδοσιακοί προγραμματιστές έχουν εγκαταλείψει τις δουλειές τους για να γίνουν προγραμματιστές iPhone πλήρους απασχόλησης
- Οι προγραμματιστές γνωρίζουν τρόπους προγραμματισμού

Χρήστες ή Προγραμματιστές

- Οι χρήστες βλέπουν τους υπολογιστές ως ένα σύνολο εργαλείων - επεξεργαστές κειμένου, υπολογιστικά φύλλα, χάρτες, λίστες υποχρεώσεων κ.λπ. Χρησιμοποιούν τα προγράμματα/εφαρμογές λογισμικού

- Οι προγραμματιστές μαθαίνουν τις «ιδιοτροπίες» του υπολογιστή και τη γλώσσα του υπολογιστή. Γράφουν προγράμματα/εφαρμογές λογισιμικού

- Οι προγραμματιστές έχουν κάποια εργαλεία που τους επιτρέπουν να δημιουργήσουν νέα εργαλεία. Editors, IDE (Integrated Development Environment)

- Οι προγραμματιστές συνήθως γράφουν εργαλεία για πολλούς χρήστες και κάποιες φορές κατασκευάζουν μικρούς «βοηθούς» για τον εαυτό τους, για να αυτοματοποιήσουν μια εργασία

Γιατί να γίνεις Προγραμματιστής?

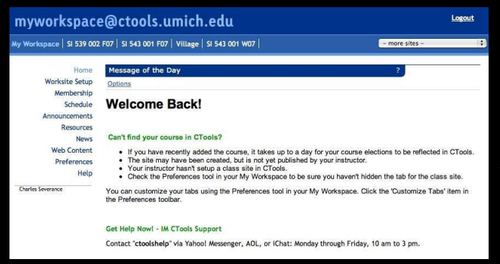
- Για να ολοκληρώσουμε κάποια εργασία με δεδομένα π.χ. στην οικονομετρία/στατιστική κλπ- είμαστε και ο χρήστης και ο προγραμματιστής

  - Οργάνωση των δεδομένων μιας έρευνας. Επεξεργασία και εκτέλεση μοντέλων (απαιτεί γνώση και χρήση γλώσσας, πχ R, Python, Visual Basic, SQL κλπ.

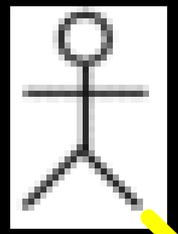
- Να δημιουργήσουμε κάτι που θα χρησιμοποιήσουν οι άλλοι – ένα πρόγραμμα / εφαρμογή λογισμικού

  - πχ λογισμικό για πελατολόγιο

  - πχ λογισμικό για έλεγχο υπολοίπου σε λογαριασμό τράπεζας



Χρήστης



Υπολογιστής
Υλικό + Λογισμικό

Προγραμματιστής

Δεδομένα

Πληροφορίες

....

Δίκτυα

Με την οπτική ενός δημιουργού λογισμικού, δημιουργούμε το λογισμικό. Οι τελικοί χρήστες (τα ενδιαφερόμενα μέρη / φορείς) είναι ο στόχος μας – αυτοί που θέλουμε να ευχαριστήσουμε - συχνά μας πληρώνουν χρήματα όταν είναι ευχαριστημένοι. Αλλά τα δεδομένα, οι πληροφορίες και τα δίκτυα είναι το πρόβλημά μας, που πρέπει να επιλύσουμε για λογαριασμό τους. Το υλικό και το λογισμικό είναι φίλοι και σύμμαχοί μας σε αυτήν την αναζήτηση.

Τι είναι ο Κώδικας; Λογισμικό; Πρόγραμμα;

- Μια ακολουθία οδηγιών, αποθηκευμένων στον Η/Υ

  - Είναι ένα μικρό κομμάτι της νοημοσύνης μας μέσα στον υπολογιστή

  - Καταλαβαίνουμε το πρόβλημα -> βρίσκουμε τη λύση -> το κωδικοποιούμε σε αλγόριθμο -> το κωδικοποιούμε σε γλώσσα προγραμματισμού -> δημιουργούμε το λογισμικό -> το δίνουμε στον τελικό χρήστη

- Ένα κομμάτι δημιουργικής τέχνης - ιδιαίτερα όταν κάνουμε καλή δουλειά σχετικά με την «εμπειρία του χρήστη»

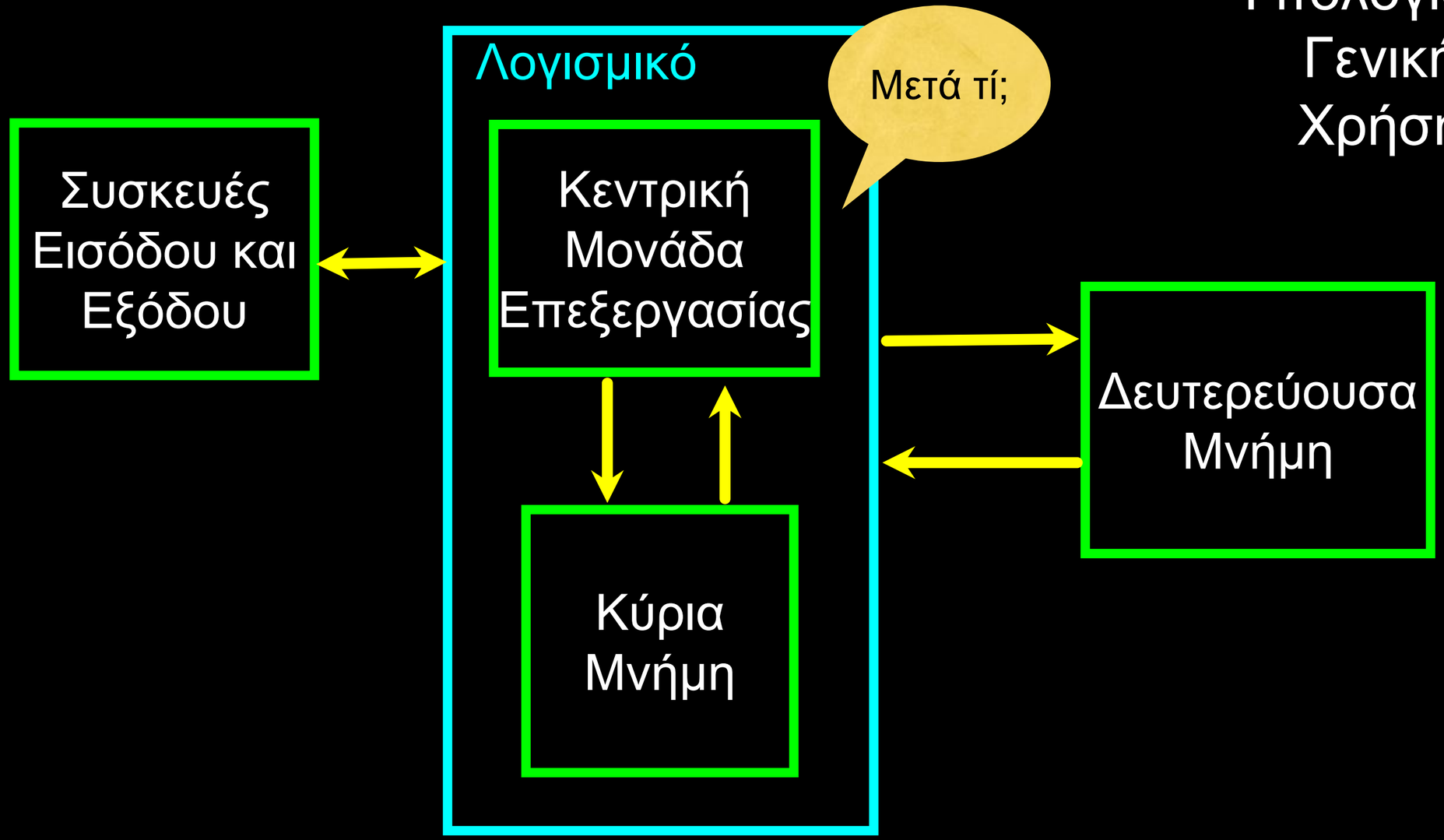
Πως είναι ο Κώδικας; Python

```
όνομα = input('Εισάγετε αρχείο:')
handle = open(όνομα)

πλήθη = dict()
for γραμμή in handle:
    λέξεις = γραμμή.split()
    for λέξη in λέξεις:
        πλήθη[λέξη] = πλήθη.get(λέξη,0) + 1

μαχπλήθος = None
μαχλέξη = None
for λέξη,πλήθος in πλήθη.items():
    if μαχπλήθος is None or πλήθος > μαχπλήθος :
        μαχλέξη = λέξη
        μαχπλήθος = πλήθος

print(μαχλέξη, μαχπλήθος)
```

Αρχιτεκτονική Υλικού / Hardware

http://upload.wikimedia.org/wikipedia/commons/3/3d/RaspberryPi.jpg

Υπολογιστής Γενικής Χρήσης

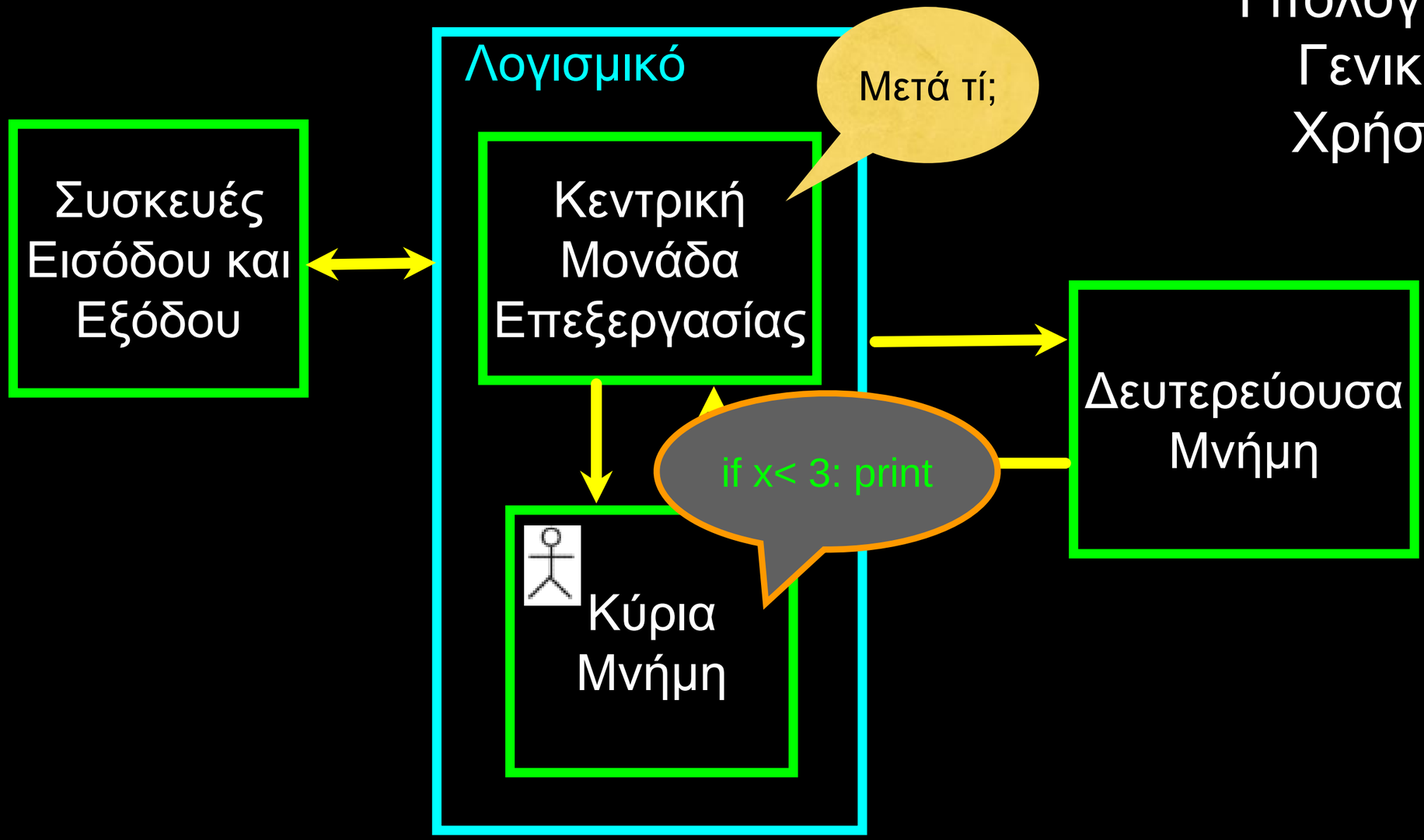


Ορισμοί

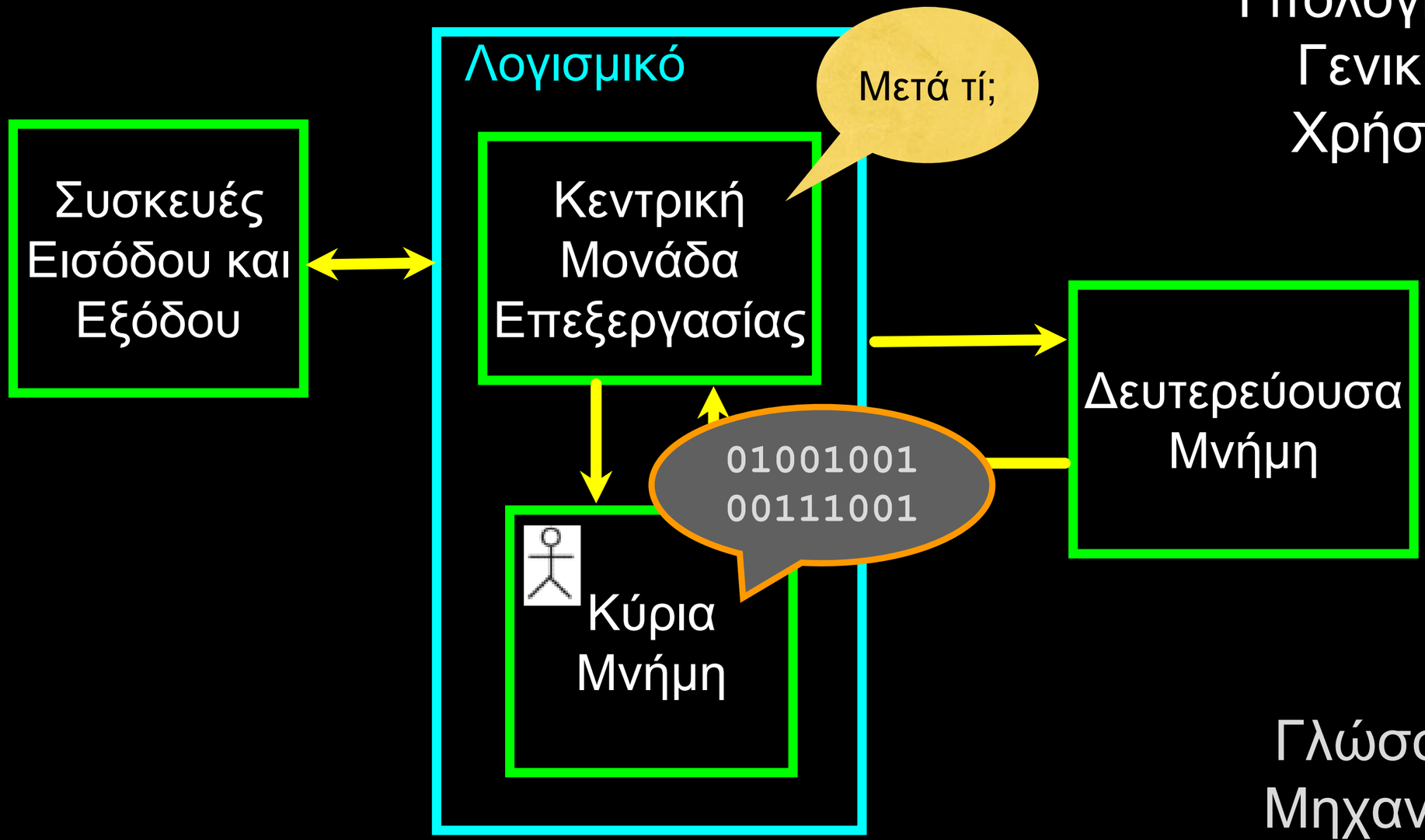
- Κεντρική Μονάδα Επεξεργασίας - CPU(Central Processing Unit): Εκτελεί το πρόγραμμα - Η CPU ρωτά πάντα «ποια είναι η επόμενη εντολή ΜΕΤΑ ΤΙ; " Όχι ακριβώς ο εγκέφαλος - πολύ “χαζή” αλλά πολύ γρήγορη.

- Συσκευές Εισόδου: Πληκτρολόγιο, Ποντίκι, Επιφάνεια Αφής.
- Συσκευές Εξόδου: Οθόνη, Ηχεία, Εκτυπωτής, DVD εγγραφής.
- Κύρια Μνήμη: Γρήγορη, μικρή, προσωρινή αποθήκευση- αδειάζει κατά την επανεκκίνηση – γνωστή και ως RAM.
- Δευτερεύουσα Μνήμη: Πιο αργή, μεγάλη, μόνιμη αποθήκευση - διατηρείται μέχρι να διαγραφεί – σκληρός δίσκος / «στικάκι» μνήμης

Υπολογιστής Γενικής Χρήσης



Υπολογιστής
Γενικής
Χρήσης



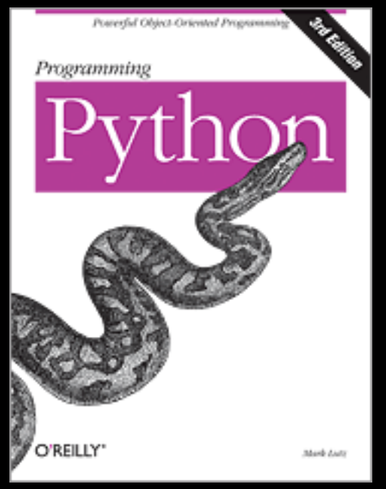
Γλώσσα
Μηχανής

Η Python ως Γλώσσα

Η Python είναι γλώσσα προγραμματισμού γενικής χρήσης.

Διερμηνευόμενη

Αναπτύχθηκε αρχικά από Guido van Rossum.



Μιλώντας στην Python

```
csev$ python3
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  5 2015, 21:12:44)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
"help", "copyright", "credits" or "license" for more information.
>>>
```

Μετά τι;

```
csev$ python3
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  5 2015, 21:12:44)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
"help", "copyright", "credits" or "license" for more information.
>>> x = 1
>>> print(x)
1
>>> x = x + 1
>>> print(x)
2
>>> exit()
```

Αυτή είναι μια καλή δοκιμή για να βεβαιωθείτε ότι έχετε εγκαταστήσει σωστά την Python. Να σημειωθεί ότι και το quit() επίσης τερματίζει την εκτέλεση της Python.

Βασικά στοιχεία

Στοιχεία της Python

- Λεξιλόγιο / Λέξεις – Μεταβλητές και Δεσμευμένες λέξεις
- Δομή πρότασης - έγκυρα πρότυπα σύνταξης
- Δομή σεναρίου/προγράμματος (script) - κατασκευή προγράμματος για έναν σκοπό

Ένα σύντομο σενάριο/script σχετικά με το πώς να μετρήσετε τις λέξεις ενός αρχείου με την Python

```
όνομα = input('Εισάγετε αρχείο:')
handle = open(όνομα)

πλήθη = dict()
for γραμμή in handle:
    λέξεις = γραμμή.split()
    for λέξη in λέξεις:
        πλήθη[λέξη] = πλήθη.get(λέξη,0) + 1

μαχπλήθος = None
μαχλέξη = None
for λέξη,πλήθος in πλήθη.items():
    if μαχπλήθος is None or πλήθος > μαχπλήθος :
        μαχλέξη = λέξη
        μαχπλήθος = πλήθος

print(μαχλέξη, μαχπλήθος )
```

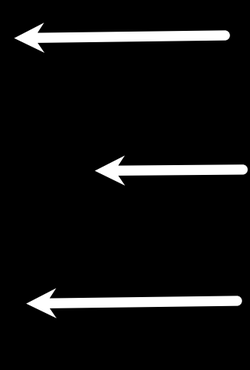
Δεσμευμένες Λέξεις

Δεν μπορείτε να χρησιμοποιήσετε δεσμευμένες λέξεις ως ονόματα μεταβλητών / αναγνωριστικών

```
False   class   return  is      finally
None    if      for     lambda  continue
True    def     from    while   nonlocal
and     del     global  not     with
as      elif    try     or      yield
assert  else    import  pass
break   except  in      raise
```

Προτάσεις ή Γραμμές

```
x = 2
x = x + 2
print(x)
```



Εντολή Εκχώρησης Τιμής
Εκχώρηση με έκφραση
Εντολή Εκτύπωσης

Μεταβλητή Τελεστής Σταθερά Συνάρτηση

Παράγραφοι Προγράμματος

Python Scripts (Σενάρια)

- Η διαδραστική Python είναι καλή για πειραματισμούς και προγράμματα μήκους 3-4 γραμμών.
- Τα περισσότερα προγράμματα είναι πολύ μεγαλύτερα, οπότε τα πληκτρολογούμε σε ένα αρχείο και λέμε στην Python να εκτελέσει τις εντολές του αρχείου.
- Κατά μία έννοια, «δίνουμε ένα σενάριο στην Python».
- Ως σύμβαση, προσθέτουμε ".py" ως επίθημα στο τέλος αυτών των αρχείων για να υποδείξουμε ότι περιέχουν Python.

Διαδραστικά έναντι Σεναρίου

- Διαδραστικά

  - Πληκτρολογείτε απευθείας στην Python μία γραμμή κάθε φορά και ανταποκρίνεται.

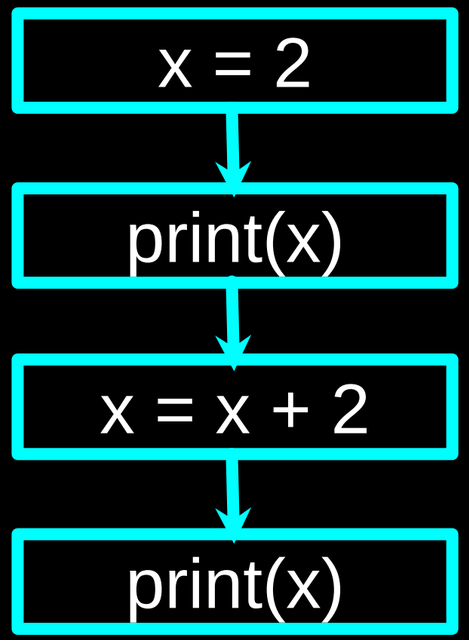
- Σενάριο

  - Εισάγετε μια ακολουθία δηλώσεων (γραμμών) σε ένα αρχείο χρησιμοποιώντας έναν επεξεργαστή κειμένου και λέτε στην Python να εκτελέσει τις δηλώσεις στο αρχείο.

Βήματα ή Ροή Προγράμματος

- Όπως μια συνταγή ή κάποιες οδηγίες εγκατάστασης, ένα πρόγραμμα είναι μια ακολουθία βημάτων που πρέπει να γίνουν με τη σειρά.
- Ορισμένα βήματα είναι υπό όρους - ενδέχεται να παραλειφθούν.
- Μερικές φορές ένα βήμα ή μια ομάδα βημάτων πρέπει να επαναληφθεί.
- Μερικές φορές αποθηκεύουμε ένα σύνολο βημάτων που θα χρησιμοποιηθούν ξανά και ξανά, ανάλογα με τις ανάγκες σε διάφορα σημεία σε όλο το πρόγραμμα

Διαδοχικά Βήματα - Δομή Ακολουθίας



Πρόγραμμα:

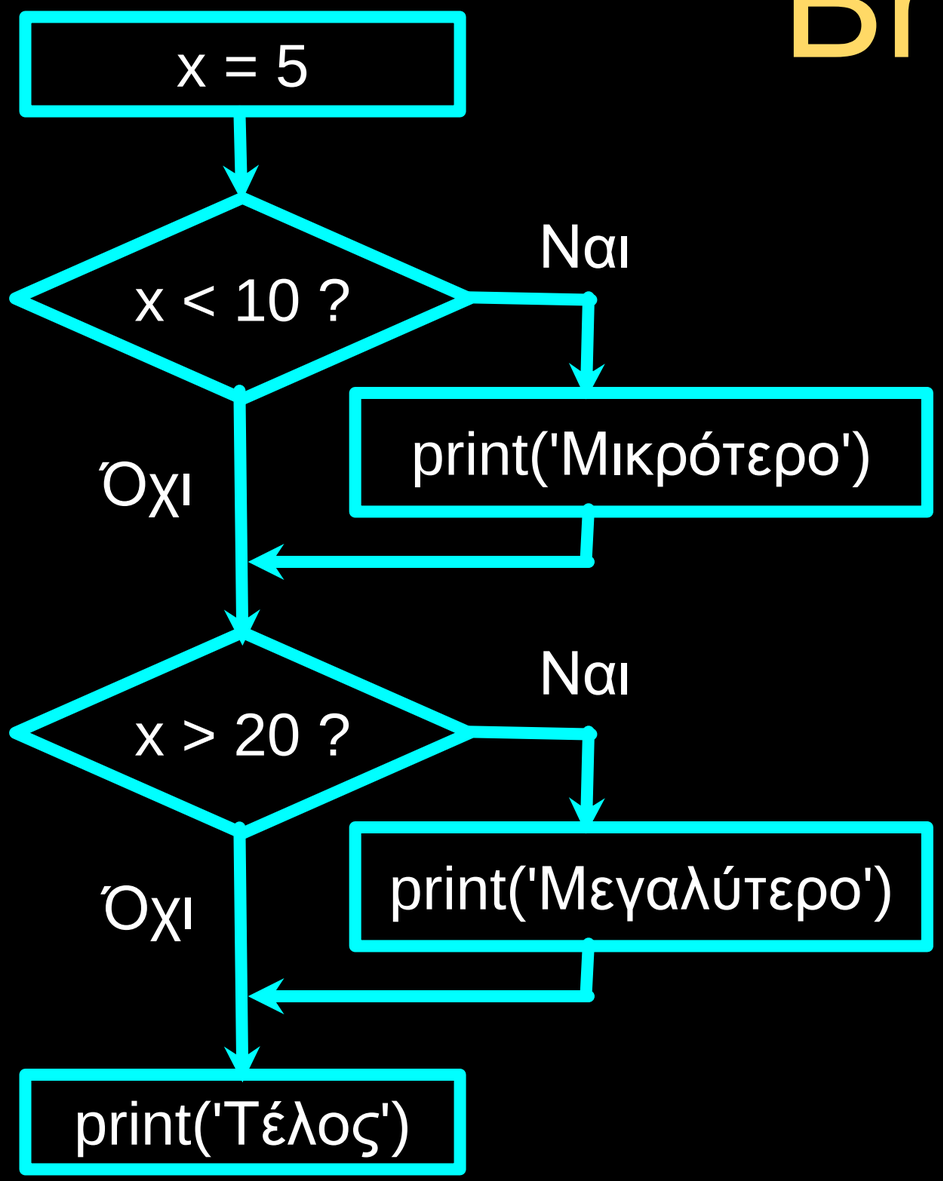
```
x = 2
print(x)
x = x + 2
print(x)
```

Έξοδος:

```
2
4
```

Όταν ένα πρόγραμμα εκτελείται, ρέει από το ένα βήμα στο επόμενο. Ως προγραμματιστές, δημιουργούμε «μονοπάτια» για να τα ακολουθήσει το πρόγραμμα.

Βήματα Υπό Όρους – Δομή Επιλογής

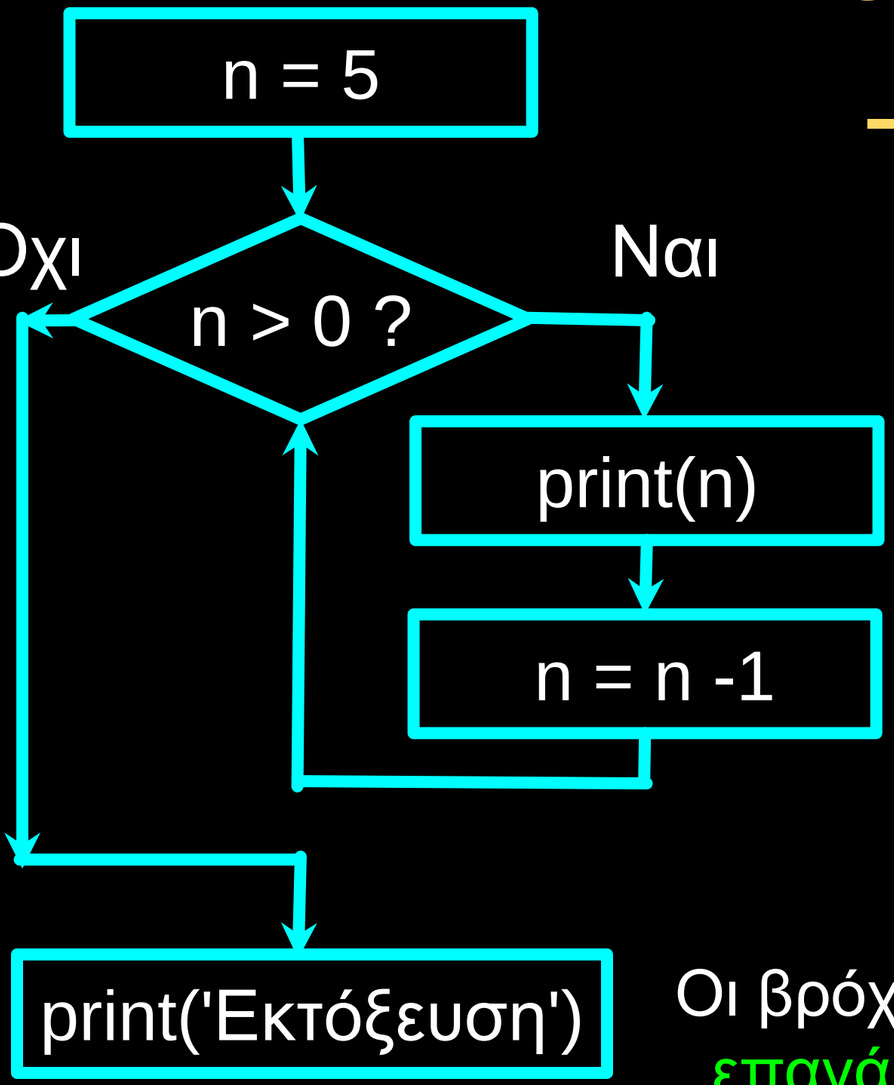


Πρόγραμμα:

```
x = 5
if x < 10:
    print('Μικρότερο')
if x > 20:
    print('Μεγαλύτερο')
print('Τέλος')
```

Έξοδος:
Μικρότερο
Τέλος

Επαναλαμβανόμενα βήματα – Δομή Επανάληψης



Πρόγραμμα:

```
n = 5
while n > 0 :
    print(n)
    n = n - 1
print('Εκτόξευση!')
```

Έξοδος:

```
5
4
3
2
1
Εκτόξευση!
```

Οι βρόχοι (επαναλαμβανόμενα βήματα) έχουν μεταβλητές επανάληψης που αλλάζουν κάθε φορά που εκτελείται ο βρόχος.

```
όνομα = input('Εισάγετε αρχείο:')
handle = open(όνομα, 'r')

πλήθη = dict()
for γραμμή in handle:
    λέξεις = γραμμή.split()
    for λέξη in λέξεις:
        πλήθη[λέξη] = πλήθη.get(λέξη,0) + 1

μαχπλήθος = None
μαχλέξη = None
for λέξη,count in πλήθη.items():
    if μαχπλήθος is None or πλήθος > μαχπλήθος:
        μαχλέξη = λέξη
        μαχπλήθος = πλήθος

print(μαχλέξη, μαχπλήθος)
```

Ακολουθία

Επανάληψη

Επιλογή

```
όνομα = input('Εισάγετε αρχείο:')
handle = open(όνομα, 'r')

πλήθη = dict()
for γραμμή in handle:
    λέξεις = γραμμή.split()
    for λέξη in λέξεις:
        πλήθη[λέξη] = πλήθη.get(λέξη,0) + 1

μαχπλήθος = None
μαχλέξη = None
for λέξη,count in πλήθη.items():
    if μαχπλήθος is None or πλήθος > μαχπλήθος:
        μαχλέξη = λέξη
        μαχπλήθος = πλήθος

print(μαχλέξη, μαχπλήθος)
```

Ένα σύντομο πρόγραμμα σε Python για να μετρήσετε τις λέξεις ενός αρχείου

Μια λέξη που χρησιμοποιείται για την ανάγνωση δεδομένων από έναν χρήστη

Μια πρόταση για τον υπολογισμό μιας από τις πολλές μετρήσεις

Μια παράγραφος σχετικά με τον τρόπο εύρεσης του μεγαλύτερου στοιχείου σε μια λίστα

Σύνοψη

- Αυτή είναι μια γρήγορη επισκόπηση
- Θα ασχολούμαστε με αυτές τις έννοιες σε όλη τη έκταση του μαθήματος
- Επικεντρωθείτε στη συνολική εικόνα