


Εισαγωγή στη γλώσσα προγραμματισμού  python™
4.: Δεδομένα, μεταβλητές και εκφράσεις

Μάθημα 4.

Λίστες και πλειάδες

4.1 Λίστες : ορισμός

Λίστες :

Ακολουθίες από
ετερογενή δεδομένα
που μπορούν να
μεταβληθούν

```
>>> my_list = ['Γιώργος', 3.14, 100]
>>>
>>> my_list[0]
'Γιώργος'
>>>
>>> my_list[1:]
[3.14, 100]
>>>
>>> len(my_list)
3
>>> my_list[0][0]
'Γ'
>>> |
```

4.2 Πράξεις σε λίστες

Ισχύουν οι πράξεις
που είδαμε στις
συμβολοσειρές
+ *n [], [:] len()

```
>>> li = [1,2,3]
>>> li[1:]
[2, 3]
>>> li[0]
1
>>> li + [4,5]
[1, 2, 3, 4, 5]
>>> li * 2
[1, 2, 3, 1, 2, 3]
>>> li = li*2
>>> len(li)
6
```

4.2 Μέθοδοι λιστών

`append(x)` προσθήκη στοιχείου στο τέλος $a[\text{len}(a):] = [x]$.

`extend(L)` επέκταση λίστας με τα στοιχεία της L $a[\text{len}(a):] = L$.

`insert(i, x)` εισαγωγή του x στη θέση i

`remove(x)` διαγραφή της πρώτης εμφάνισης του x στη λίστα, `error` αν δεν υπάρχει το x

`pop([I])` διαγραφή του στοιχείου στη θέση I , `pop()` διαγράφει το τελευταίο στοιχείο

`index(x)` η θέση του στοιχείου x , `error` αν δεν υπάρχει το x

`count(x)` πόσες φορές εμφανίζεται το x στη λίστα

`sort()` ταξινόμηση των στοιχείων της λίστας αλλάζοντας την

`reverse()` αντίστροφη ταξινόμηση των στοιχείων, αλλάζοντας την

Λίστα 2 διαστάσεων

```
pinax = [ [2,6,4,7], [1,0,3,2] ]
```

| | 0 | 1 | 2 | 3 |
|---|----------|----------|----------|----------|
| 0 | 2 | 6 | 4 | 7 |
| 1 | 1 | 0 | 3 | 2 |

Πώς αναφερόμαστε στο 3 ;

```
pinax[1][2]
```

Ταξινόμηση λίστας

```
>>> a = [8,5,10]
>>> b = sorted(a)
>>> b
[5, 8, 10]
>>> a
[8, 5, 10]
>>> a = [8,5,10]
>>> b = a.sort()
>>> b
>>> a
[5, 8, 10]
```

sorted(λ) επιστρέφει
τη λίστα ταξινομημένη

λ.**sort()** την ταξινομεί
την ίδια

Πλειάδες (tuples)

Παρόμοιες με τις λίστες, με μια σημαντική
διαφορά : είναι αμετάβλητες

```
>>> t = (1,2,3)
```

```
>>> t[0] = 4
```

```
Traceback (most recent call last):
```

```
File "<pyshell#8>", line 1, in <module>
```

```
    t[0] = 4
```

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