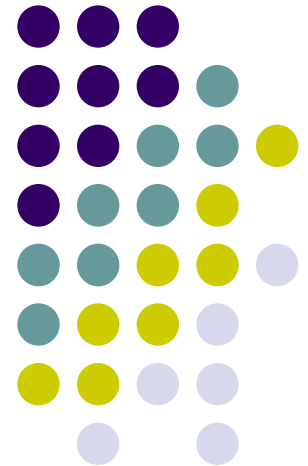
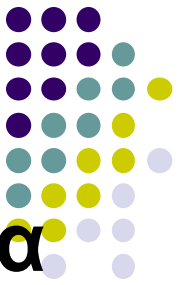


Δομές Δεδομένων Λίστα

*Γ' ΕΠΑΛ
Κ. Φλώρος*



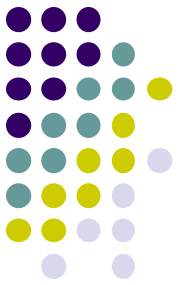
Εισαγωγή (1) - Λίστες



Η **λίστα** είναι μια διατεταγμένη ακολουθία από αντικείμενα, όχι απαραίτητα του ίδιου τύπου.

Αν και μια λίστα μπορεί να αποτελείται και από αντικείμενα διαφορετικού τύπου συνήθως όμως είναι του ίδιου τύπου.

- Η **λίστα** αποτελεί τη βασική δομή δεδομένων στην Python. Είναι δυναμική δομή δηλαδή μπορούμε να της προσθέτουμε και να της αφαιρούμε στοιχεία(`mutable`).



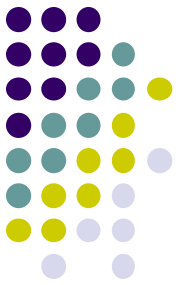
Εισαγωγή (2)- Λίστες

Τα στοιχεία λίστας περιλαμβάνονται μεταξύ των στοιχείων των αγκυλών «[.]».

Κάθε αντικείμενο της λίστας χαρακτηρίζεται από ένα μοναδικό αύξοντα αριθμό, ο οποίος ορίζει τη θέση του στη λίστα.

Η προσπέλαση στα στοιχεία της λίστας γίνεται όπως στις συμβολοσειρές, με το όνομα της λίστας και τον αύξοντα αριθμό του αντικειμένου μέσα σε αγκύλες.

Η αρίθμηση των στοιχείων ξεκινάει από το 0.



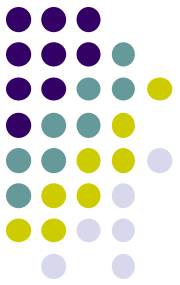
Λίστες - Παράδειγμα

- Η αρίθμηση των χαρακτήρων στη λίστα όπως και στη συμβολοσειρά ξεκινάει από το 0. Για παράδειγμα:

αν $L = [3, 5, 8, 13, 21, 34]$

0	1	2	3	4	5
3	5	8	13	21	34
L[0]	L[1]	L[2]	L[3]	L[4]	L[5]

- Το 1^ο στοιχείο της λίστας είναι το L[0], το οποίο είναι ίσο με το 3, το 2^ο το L[1], ... και τελευταίο το L[5].



Άσκηση 1- Παράδειγμα

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

```
>>> print fruits
```

```
['apple', 'orange', 'banana', 'cherry']
```

```
>>> print fruits[0]
```

```
'apple'
```

```
>>> print fruits[1]
```

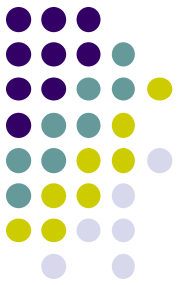
```
'orange'
```

```
>>> print fruits[2]
```

```
'banana'
```

```
>>> print fruits[3]
```

```
'cherry'
```



Άσκηση 2- Παράδειγμα

```
>>> lista=[10, 20,30,40]
```

```
>>> print lista
```

```
[10, 20,30,40]
```

```
>>> print lista[0]
```

```
10
```

```
>>> print lista[1]
```

```
20
```

```
>>> print lista[2]
```

```
30
```

```
>>> print lista[3]
```

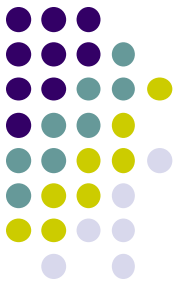
```
40
```



Άσκηση 3 - Παράδειγμα

Ποιο θα είναι το αποτέλεσμα μετά τη εκτέλεση του προγράμματος;

```
daysofweek=["Δευτέρα","Τρίτη","Τετάρτη",  
            "Πέμπτη","Παρασκευή","Σάββατο","Κυρι  
ακή"]  
print "Θα έχω ελεύθερο χρόνο την ",  
      daysofweek[6]
```



Προσθήκη σε λίστα

Αν θέλουμε να προσθέσουμε ένα στοιχείο στο τέλος της λίστας γράφουμε:

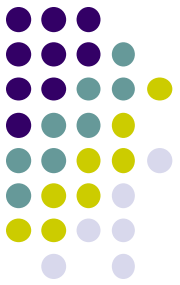
Λίστα = Λίστα + [στοιχείο]

Ενώ στην αρχή λίστας

Λίστα = [στοιχείο] + Λίστα

Σημείωση: Στην πραγματικότητα οι παραπάνω εντολές δεν προσθέτουν το στοιχείο στην ήδη υπάρχουσα λίστα αλλά δημιουργούν μια νέα λίστα κάθε φορά. Για αυτό αν θέλουμε να προσθέσουμε ένα στοιχείο στο τέλος της λίστας προτιμούμε το τελεστή += όπως παρακάτω:

Λίστα += [στοιχείο]



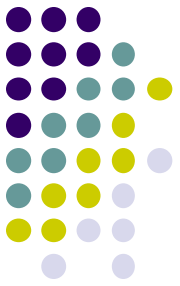
Άσκηση 4 - Παράδειγμα

```
>>>lista1=[10, 20, 30, 40]
```

```
>>>lista2=[15, 25, 45, 95]
```

```
>>>print lista1+lista2
```

```
[10, 20, 30, 40, 15, 25, 45, 95]
```



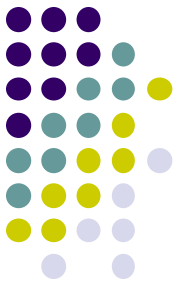
Άσκηση 5 - Παράδειγμα

```
fruits=['apple', 'orange','banana','cherry']
```

```
list1=[10, 20, 30, 40, 90]
```

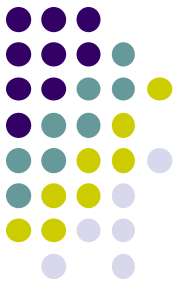
```
print fruits+list1
```

```
['apple', 'orange', 'banana', 'cherry', 10, 20, 30, 40, 90]
```



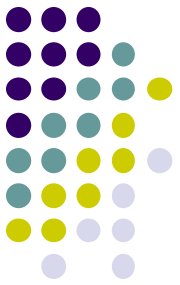
Άσκηση 6 - Παράδειγμα

```
>>> fib=[3, 5, 8, 13, 21]
>>> fib=fib + [ fib[3] + fib[4] ]
>>> print fib
[3, 5, 8, 13, 21, 34]
```



Άσκηση 7 - Παράδειγμα

```
fruits=['apple', 'orange','banana','cherry']  
print fruits*4
```



Άσκηση 8 - Παράδειγμα

```
>>> a=[1,3,5,7,11,13,17,23]
```

```
>>> print a
```

```
[1, 3, 5, 7, 11, 13, 17, 23]
```

```
>>> print a[2]
```

```
5
```

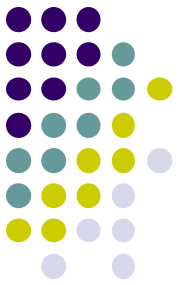
```
>>> a[2]=0
```

```
>>> print a
```

```
[1, 3, 0, 7, 11, 13, 17, 23]
```

```
>>> print a[2]
```

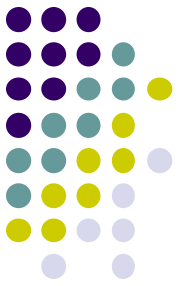
```
0
```



Τελεστές στις λίστες

- **item in List** → Επιστρέφει **True**, αν το στοιχείο *item* υπάρχει μέσα στη λίστα *List*, αλλιώς επιστρέφει **False**.
- **item not in List** → Επιστρέφει **True** αν το στοιχείο *item* δεν υπάρχει μέσα στη λίστα *List*, αλλιώς αν υπάρχει επιστρέφει **False**.

Τελεστές στις λίστες - Παραδείγματα



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

```
>>> 'apple' in fruits
```

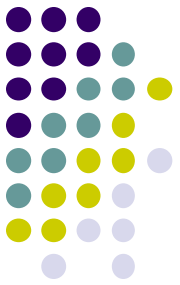
```
True
```

```
>>> 'melon' not in fruits
```

```
True
```

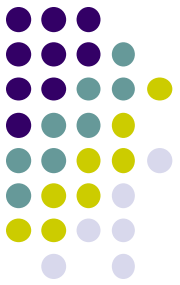
```
>>> 'banana' not in fruits
```

```
False
```



Συναρτήσεις στις λίστες

- **len(List)** → Επιστρέφει το πλήθος των στοιχείων (ή μέγεθος) της λίστας.
Π.χ. $L=[10, 20, 30, 40, 50]$ τότε
 $\text{len}(L) \rightarrow 5$
- **list(String)** → Η συνάρτηση list επιστρέφει μια λίστα με στοιχεία τους χαρακτήρες της συμβολοσειράς string. Χωρίς όρισμα η list επιστρέφει άδεια λίστα. Για παράδειγμα:



Άσκηση 9 - Παράδειγμα

- `list1=list('epal alimou')`

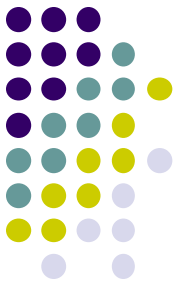
Δημιουργεί μια λίστα με στοιχεία τους χαρακτήρες της συμβολοσειράς που δίνεται ως όρισμα. Δηλαδή:

```
list1=['e', 'p', 'a', 'l', ' ', 'a', 'l', 'i', 'm', 'o', 'u'].
```

- Χωρίς όρισμα επιστρέφει άδεια λίστα. Π.χ.

```
list2=list() τότε
```

```
list2=[]
```



Μέθοδοι στις λίστες (append)

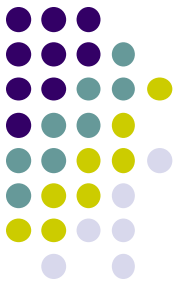
- Listname.append (object) → προσθήκη του στοιχείου object στο τέλος της λίστας. Π.χ.

```
list1=['red', 'blue', 'black']
```

```
list1.append('green')
```

Άρα list1=['red', 'blue', 'black', 'green']

ΠΡΟΣΟΧΗ!!! Με την append μπορούμε να προσθέσουμε ένα μόνο στοιχείο στη λίστα στο τέλος της.



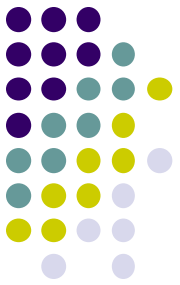
Άσκηση 10 - Παράδειγμα

Ποιο θα είναι το αποτέλεσμα μετά τη εκτέλεση του προγράμματος;

```
a=[10,15,20,25,30,40]
```

```
a.append(70)
```

```
print a
```



Μέθοδοι στις λίστες (insert)

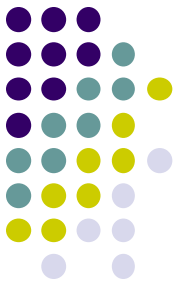
`Listname.insert(index,object)` : προσθήκη του στοιχείου `object` στη θέση `index` της λίστας, μετακινώντας όλα τα στοιχεία από τη θέση `index` και μετά, κατά μία θέση. Π.χ.

```
a=[10,15,20, 25,30,40]
```

```
a.insert(2,50)
```

```
print a
```

```
[10,15, 50, 20, 25, 30, 40]
```



Μέθοδοι στις λίστες (pop)

Listname.pop(index) → Αφαιρεί από τη λίστα το στοιχείο που βρίσκεται στη θέση index. Αν δε δοθεί θέση, τότε θα αφαιρεθεί το τελευταίο στοιχείο της λίστας.

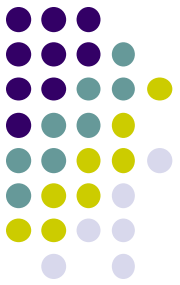
```
a=[10,5,2,25,3,40,7,29]
```

```
a.pop(3)
```

```
print a -> [10,5,2,3,40,7,29]
```

```
a.pop()
```

```
print a -> [10,5,2,3,40,7]
```

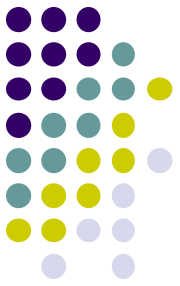


Άσκηση 11 - Παράδειγμα

Ποιο θα είναι το αποτέλεσμα μετά τη εκτέλεση του προγράμματος;

```
sqlist=[ ]  
for x in range(1,11):  
    sqlist.append(x*x)  
print sqlist
```

Προσπέλαση των στοιχείων της λίστας (Διάσχιση λίστας)

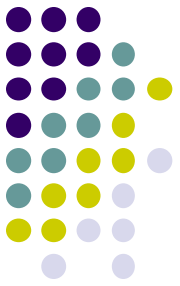


Παράδειγμα

```
lista=[10,15,20]  
for item in lista:  
    print item
```

Για να επεξεργαστούμε
τα στοιχεία της λίστας,
ένα κάθε φορά:
for item in lista:
 <Εντολές>

Προσπέλαση των στοιχείων μιας λίστας



Παράδειγμα

```
lista=[10,15,20]
```

```
n=len(lista)
```

```
for i in range(n):
```

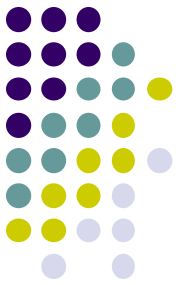
```
    print lista[i]
```



Άσκηση 14 - Παράδειγμα

Ποιο θα είναι το αποτέλεσμα μετά τη εκτέλεση του προγράμματος;

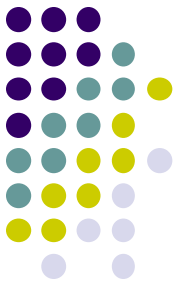
```
daysofweek=["Δευτέρα", "Τρίτη",  
            "Τετάρτη", "Πέμπτη", "Παρασκευή",  
            "Σάββατο", "Κυριακή"]  
for item in daysofweek:  
    print item
```



Άσκηση 15

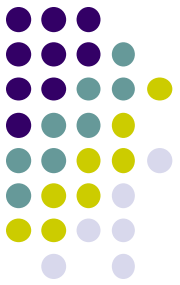
Να γράψετε πρόγραμμα σε python το οποίο:

- να δημιουργεί μία λίστα,
- να τη γεμίζει με 10 αριθμούς που θα εισάγει ο χρήστης,
- να εμφανίζει τη λίστα,
- να υπολογίζει και εμφανίζει το άθροισμα τους,
- να υπολογίζει και εμφανίζει το μέσο όρο τους,
- να βρίσκει και να εμφανίζει τον μικρότερο αριθμό και
- να βρίσκει και να εμφανίζει τον μεγαλύτερο αριθμό.



Άσκηση 15 - Λύση

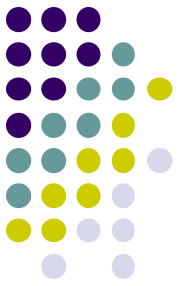
```
L=[]  
for i in range(1,11):  
    num=int(input('Δώστε αριθμό: '))  
    L.append(num)  
print "Η λίστα είναι: ", L  
s = 0.0  
for number in L :  
    s = s + number  
print 'ο αθροισμα είναι: ',s
```



Άσκηση 15 – Λύση(συνέχεια)

```
average = s / len( L )  
print 'ο mesos oros einai: ',average  
for number in L :  
    print number  
mik=L[0]  
for number in L :  
    if number<mik:  
        mik=number  
print 'ο μικρότερος αριθμος είναι: ',mik
```

Άσκηση 15 – Λύση(συνέχεια)



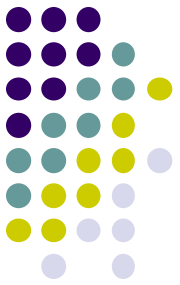
```
meg=L[0]  
for number in L :  
    if number>meg:  
        meg=number  
print 'ο μεγαλύτερος αριθμος της λίστας  
    είναι: ', meg
```



Άσκηση 16

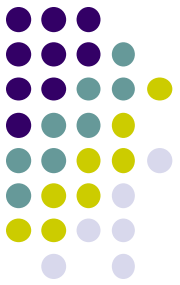
Να γράψετε πρόγραμμα σε python το οποίο:

- να δημιουργεί μία λίστα,
- να τη γεμίζει με θερμοκρασίες που θα εισάγει ο χρήστης,
- η εισαγωγή τελειώνει όταν για θερμοκρασία δώσουμε το 9999,
- να εμφανίζει τη λίστα,
- να υπολογίζει και εμφανίζει το μέσο όρο των θερμοκρασιών και
- να εμφανίζει τα στοιχεία της λίστας



Άσκηση 16 - Λύση

```
L=[]  
num=int(input('Δώστε αριθμό: '))  
while num!=9999:  
    L.append(num)  
    num=int(input('Δώστε αριθμό: '))  
  
print "Η λίστα είναι: ", L
```



Άσκηση 17 - Λύση(συνέχεια)

s = 0.0

for number in L :

s = s + number

if len(L)!=0:

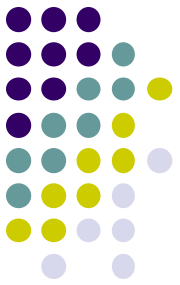
average = s / len(L) # δεν θα γίνει ακέραια
διαίρεση

print 'ο mesos oros einai: ', average

for number in L :

print number

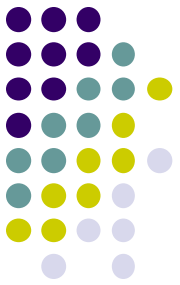
Άσκηση 17 («Παράλληλες λίστες»)



Σε ένα διαγωνισμό συμμετέχουν 100 μαθητές.

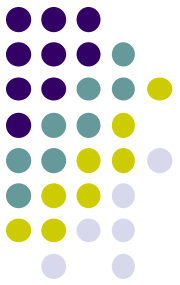
Να γράψετε πρόγραμμα σε python το οποίο:

- Να διαβάσει το ονοματεπώνυμο κάθε μαθητή και τη βαθμολογία που έλαβε.
- Τα στοιχεία αυτά καταχωρίζονται στις λίστες NAME και VATHMOS αντίστοιχα.
- Να υπολογίζει και να εμφανίζει τον μέσο όρο (MO) της βαθμολογίας όλων των μαθητών.
- Να εντοπίζει και να εμφανίζει με κατάλληλο μήνυμα την υψηλότερη βαθμολογία και
- τα ονοματεπώνυμα των μαθητών που έχουν αυτή τη βαθμολογία.



Άσκηση 17 - Λύση

```
NAME=[ ]
VATHMOS=[ ]
for i in range(100):
    onoma=raw_input('Δώστε ονοματεπώνυμο: ')
    NAME.append(onoma)
    num=int(input('Δώστε βαθμό: '))
    VATHMOS.append(num)
s = 0.0
for number in VATHMOS :
    s = s + number
print 'ο αθροισμα είναι: ',s
average = s / len( VATHMOS )
print 'ο mesos oros einai: ',average
```

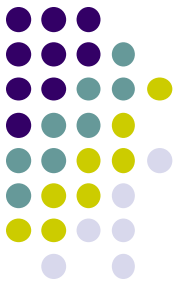


Άσκηση 17 – Λύση(συνέχεια)

```
meg=VATHMOS[0]
for number in VATHMOS :
    if number>meg:
        meg=number
print'ο μεγαλύτερος βαθμός της λίστας είναι: ', meg

print 'Οι μαθητές με τη μεγαλύτερη βαθμολογία είναι: '
for i in range(100):
    if VATHMOS[i]==meg:
        print NAME[i]
```

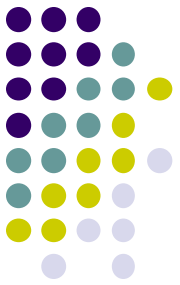
Άσκηση 18 (Διαχωρισμός λίστας)



Δίδεται η λίστα $L=[5, 7, -1, 3, -11, 6, -4, 22, -3]$ με ακέραιους αριθμούς.

Να γράψετε πρόγραμμα σε python το οποίο:

- Να διαχωρίζει τη λίστα σε δύο άλλες λίστες, η μια λίστα με όνομα POSITIVES να περιλαμβάνει τους θετικούς αριθμούς και η άλλη με όνομα NEGATIVES τους αρνητικούς αριθμούς της αρχικής λίστας αντίστοιχα.
- Τέλος να εμφανίζει τις λίστες POSITIVES και NEGATIVES



Άσκηση 18 - Λύση

```
L=[5, 7, -1, 3, -11, 6, -4, 22, -3]
```

```
POSITIVES =[ ]
```

```
NEGATIVES =[ ]
```

```
for num in L:
```

```
    if num>0:
```

```
        POSITIVES.append(num)
```

```
    else:
```

```
        NEGATIVES.append(num)
```

```
print POSITIVES
```

```
print NEGATIVES
```

Άσκηση 19 (Συγχώνευση διατεταγμένων λιστών)

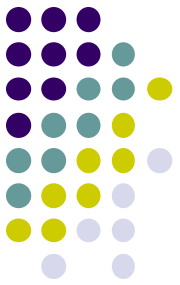


Δίδονται οι ταξινομημένες λίστες $X=[1,3,5,7,9,11]$
και $Y=[2,4,6,8]$.

Να γράψετε πρόγραμμα σε python το οποίο:

- Να συγχωνεύει τα στοιχεία των δύο ταξινομημένων λιστών, σε μια νέα ταξινομημένη λίστα με όνομα NEALISTA.
- Τέλος να εμφανίζει την νέα ταξινομημένη λίστα.

Άσκηση 19 – Λύση



X=[1,3,5,7,9,11]

Y=[2,4,6,8]

NEALISTA=[]

while X!=[] and Y!=[]:

if X[0]<Y[0]:

NEALISTA.append(X.pop(0))

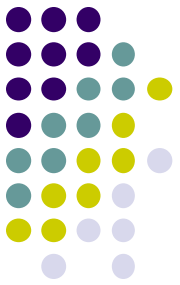
else:

NEALISTA.append(Y.pop(0))

NEALISTA=NEALISTA+X+Y

print NEALISTA

Άσκηση 19 – Λύση (Με συνάρτηση)



```
X=[1,3,5,7,9,11]
```

```
Y=[2,4,6,8]
```

```
NEALISTA=merge(X,Y)
```

```
print NEALISTA
```

```
def merge(A,B):
```

```
    L=[ ]
```

```
    while A!=[ ] and B!=[ ]:
```

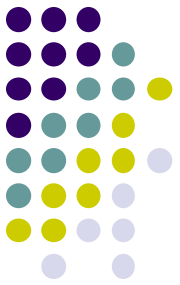
```
        if A[0]<B[0]:
```

```
            L.append(A.pop(0))
```

```
        else:
```

```
            L.append(B.pop(0))
```

```
    return L+A+B
```



Η συνάρτηση range

Η συνάρτηση `range(A, M, B)` επιστρέφει μια λίστα αριθμών ξεκινώντας με τον αριθμό `A` μέχρι το `M` με βήμα `B`. Το `M` δεν συμπεριλαμβάνεται στη λίστα.

```
>>> range(4)
```

```
[0, 1, 2, 3]
```

```
>>> range(0, 4)
```

```
[0, 1, 2, 3]
```

```
>>> range(0, 4, 1)
```

```
[0, 1, 2, 3]
```

Η συνάρτηση range - Παραδείγματα



```
>>> range(1, 1, 10)
```

```
[]
```

```
>>> range(1, 5, -1)
```

```
[]
```

```
>>> range(0)
```

```
[]
```

```
>>> range(10, 30, 5)
```

```
[10, 15, 20, 25]
```

```
>>> range(30, 10, -5)
```

```
[30, 25, 20, 15]
```

```
>>> range(1, 2, 10)
```

```
[1]
```



Άσκηση 20 - Παράδειγμα

```
>>> a=[1, 3, 5, 7, 11, 13, 17, 23]
```

```
>>> print a
```

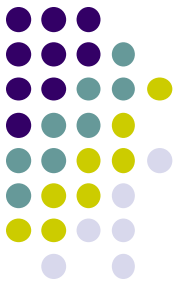
```
[1, 3, 5, 7, 11, 13, 17, 23]
```

```
>>> print len(a)
```

```
8
```

```
>>> print a[1:4]
```

```
[3, 5, 7]
```



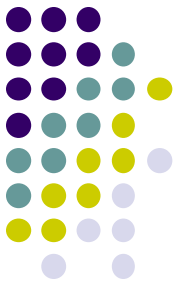
Τελεστής διαμέρισης :

Ο τελεστής διαμέρισης (slice operator) συμβολίζεται με την άνω κάτω τελεία :

Μπορεί να μας επιστρέψει ένα τμήμα μιας συμβολοσειράς ή λίστας.

Η έκφραση `x[a:b]` μας επιστρέφει το τμήμα της συμβολοσειράς ή της λίστας από το στοιχείο `x[a]` μέχρι και το στοιχείο `x[b-1]`. Μπορούμε να παραλείψουμε την αρχή ή το τέλος αν θέλουμε ένα τμήμα από την αρχή ή το τέλος της λίστας/συμβολοσειράς αντίστοιχα.

Τελεστής διαμέρισης – Παράδειγμα με συμβολοσειρά



```
>>> lexi='epal alimou'
```

```
>>> lexi[:4]
```

```
'epal'
```

```
>>> lexi[5:]
```

```
'alimou'
```

```
>>> lexi[:]
```

```
'epal alimou'
```

```
>>> lexi[3:8]
```

```
'l ali'
```

```
>>> lexi[0:4]
```

```
'epal'
```

```
>>> lexi[5:len(lexi)]
```

```
'alimou'
```

```
>>> lexi[0:len(lexi)]
```

```
'epal alimou'
```

```
>>> lexi[:len(lexi)]
```

```
'epal alimou'
```

Αλγόριθμος κρυπτογράφησης του Καίσαρα - Παράδειγμα



```
>>> alphabet='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
>>> cipherAlphabet = alphabet[3: ] + alphabet[ :3]
>>> print cipherAlphabet
DEFGHIJKLMNOPQRSTUVWXYZABC
```

Τελεστής διαμέρισης – Παράδειγμα με λίστες (1)



```
>>> L=[5, 8, 13, 21, 34]
```

```
>>> L[:3]
```

```
[5, 8, 13]
```

```
>>> L[0:3]
```

```
[5, 8, 13]
```

```
>>> L[3:]
```

```
[21, 34]
```

```
>>> L[3:5]
```

```
[21, 34]
```

```
>>> L[3:len(L)]
```

```
[21, 34]
```

```
>>> L[0:len(L)]
```

```
[5, 8, 13, 21, 34]
```

```
>>> L[:]
```

```
[5, 8, 13, 21, 34]
```

```
>>> L[2:4]
```

```
[13, 21]
```

```
>>> L[:len(L)]
```

```
[5, 8, 13, 21, 34]
```

Τελεστής διαμέρισης – Παράδειγμα με λίστες (2)



```
>>> f=[5,8,13,21,34]
```

```
>>> fib=f[:] #δημιουργεί αντίγραφο
```

```
>>> print fib
```

```
[5, 8, 13, 21, 34]
```

```
>>> a=fib #δείχνουν την ίδια λίστα
```

```
>>> a.pop()
```

```
34
```

```
>>> fib.pop()
```

```
21
```

```
>>> a[0]=a[1]=55
```

```
>>> print a
```

```
[55, 55,13]
```

```
>>> print fib
```

```
[55, 55,13]
```

```
>>> print f
```

```
[5,8,13,21,34]
```

#η αρχική λίστα δεν
έχει αλλάξει



Συναρτήσεις και Λίστες (1)

```
def changeme(mylist):  
    mylist.append([1, 2, 3, 4])  
    print "Τιμές μέσα στη συνάρτηση:", mylist  
    return  
mylist=[10, 20, 30]  
changeme(mylist)  
print " Τιμές έξω από τη συνάρτηση: ", mylist
```

Αποτέλεσμα:

Τιμές μέσα στη συνάρτηση: [10,20,30,[1,2,3,4]]

Τιμές έξω από τη συνάρτηση:[10,20,30,[1,2,3,4]]



Συναρτήσεις και Λίστες (2)

```
def changeme(mylist):  
    mylist=[1, 2, 3, 4]  
    print "Τιμές μέσα στη συνάρτηση:", mylist  
    return
```

```
mylist=[10, 20, 30]  
changeme(mylist)  
print " Τιμές έξω από τη συνάρτηση: ", mylist
```

Αποτέλεσμα:

Τιμές μέσα στη συνάρτηση: [1, 2, 3, 4]

Τιμές έξω από τη συνάρτηση: [10, 20, 30]