

Λίστες

Η Python έχει μία δομή η οποία ονομάζεται λίστα η οποία κρατάει ένα σύνολο τιμών. Ο ακόλουθος κώδικας καταχωρεί μία λίστα η οποία ονομάζεται `a` και τις αναθέτει τις τιμές 2,5,6,8 και 1.

```
a=[2 , 5 , 6 , 8 , 1]
```

Για να επιστρέψουμε την τιμή της λίστας σε κάποια θέση πρέπει να χρησιμοποιήσουμε τη θέση μέσα σε `[]`.

```
a=[2 , 5 , 6 , 8 , 1]
print a[1]
```

Προσοχή: Η αρίθμηση των στοιχείων της λίστας ξεκινάει από το 0!

Μπορούμε να καλέσουμε στοιχεία, μετρώντας τη σειρά τους από το τέλος της λίστας χρησιμοποιώντας αρνητικούς αριθμούς. Ο ακόλουθος κώδικας θα εκτυπώσει την τελευταία και προτελευταία τιμή:

```
a=[2 , 5 , 6 , 8 , 1]
print a[-1] , a[-2]
```

Προσοχή: Αν ζητήσουμε στοιχείο σε θέση μεγαλύτερη από το μήκος της λίστας η Python θα επιστρέψει πρόβλημα!

Για να αλλάξουμε την τιμή ενός στοιχείου, απλά ζητάμε το στοιχείο στη θέση που επιθυμούμε να κάνουμε την αλλαγή και κάνουμε την αντίστοιχη ανάθεση.

```
a=[2,5,6,8,1]
a[3]=11
print a
```

Για να προσθέσουμε ένα στοιχείο στο τέλος της λίστας χρησιμοποιούμε την `append`:

```
a=['a','b','c','d','e']
a.append('f')
print a
```

Αν θέλουμε να προσθέσουμε ένα στοιχείο σε συγκεκριμένη θέση της λίστας χρησιμοποιούμε την `insert`, όπου στη σύνταξή της δηλώνουμε τη θέση και την τιμή.

```
a=[2,5,6,8,1]
a.insert(1,100)
print a
```

Με την `remove` αφαιρούμε το πρώτο στοιχείο της λίστας που περιέχει την τιμή που του δηλώσαμε. Έτσι ο κώδικας:

```
a=[2,5,1,6,8,1]
a.remove(1)
print a
```

θα επιστρέψει τη λίστα `[2,5,6,8,1]`.

Με την `count` μετράμε πόσα στοιχεία έχουν την τιμή που ζητήσαμε. Έτσι ο κώδικας:

```
a=[1,2,5,1,6,8,1]
print a.count(1)
```

θα επιστρέψει την τιμή 3.

Για να αντιστρέψουμε τα στοιχεία μίας λίστας χρησιμοποιούμε τη `reverse`:

```
a=[1,2,3,4,5]
a.reverse()
print a
```

Μπορούμε να φτιάξουμε λίστες οι οποίες είναι μέρος μίας άλλης λίστας. Πιο συγκεκριμένα αν `a` είναι η λίστα `[1,2,3,4,5,6,7,8,9,10]`, τότε:

Κλήση	Αποτέλεσμα
<code>a[3:8]</code>	<code>[4, 5, 6, 7, 8]</code>
<code>a[:8]</code>	<code>[1, 2, 3, 4, 5, 6, 7, 8]</code>
<code>a[:-1]</code>	<code>[1, 2, 3, 4, 5, 6, 7, 8, 9]</code>
<code>a[4:]</code>	<code>[5, 6, 7, 8, 9, 10]</code>
<code>a[::2]</code>	<code>[1, 3, 5, 7, 9]</code>
<code>a[1::2]</code>	<code>[2, 4, 6, 8, 10]</code>

Προσέξτε το πως ζητάμε τα ``μέρη" της λίστας, η σύνταξη είναι [αρχή, τέλος, βήμα]

Για να διαγράψουμε στοιχεία μίας λίστας μπορούμε να χρησιμοποιήσουμε τις συναρτήσεις `del` και `pop`

```
a = [1, 2, 3, 4, 5]
del a[1]
print a
```

Ο παραπάνω κώδικας διαγράφει το στοιχείο της λίστας `a` στη θέση 1 (δεύτερη θέση).

Η `pop` αφαιρεί το τελευταίο στοιχείο της λίστας και το αφαιρεί.

```
a = [1, 2, 3, 4, 5]
b = a.pop()
print b, a
```

Ο κώδικας παραπάνω θα θέσει στο `b` την τιμή 5, ενώ η λίστα `a` θα είναι πλέον `[1,2,3,4]`. Η `pop` επιπλέον μπορεί να διαγράψει και ένα στοιχείο το οποίο βρίσκεται σε άλλη θέση.

```
a = [1, 2, 3, 4, 5]
b = a.pop(2)
print b, a
```

Ο κώδικας παραπάνω θα θέσει στο `b` την τιμή 3, ενώ η λίστα `a` θα είναι πλέον `[1,2,4,5]`.

Στις λίστες μπορώ να εφαρμόσω διάφορες συναρτήσεις. Πιο συγκεκριμένα, η `len` επιστρέφει το μήκος της λίστας, η `max` το μέγιστο στοιχείο, η `min` το ελάχιστο στοιχείο, ενώ τέλος η `sum` το άθροισμα των στοιχείων.

```
a = [11, 62, 30, 45, 15]
print len(a), max(a), min(a), sum(a)
```

Ο παραπάνω κώδικας θα επιστρέψει τις τιμές: 5 62 11 163.

Μπορούμε να ενώσουμε δύο λίστες:

```
a=[1,2,3,4,5]
b=[6,7,8]
c=a+b
print c
```

Η απλά να ενώσουμε προσθέσουμε σε μία λίστα στοιχεία μέσω της extend:

```
a=[1,2,3,4,5]
a.extend([6,7,8])
print a
```

Για να αντιγράψουμε τα στοιχεία μίας λίστας σε μία άλλη χρησιμοποιούμε τη list.

```
a=[1,2,3,4,5]
b=list(a)
c=a
a.append(6)
print a,b,c
```

Προσοχή: Με το να θέσουμε μία λίστα ίση με μία άλλη δεν αντιγράφονται τα στοιχεία, αλλά οι δύο λίστες δείχνουν στην ίδια λίστα για αυτό και τον προηγούμενο κώδικα η c περιέχει και το στοιχείο 6, ενώ η b δεν το περιέχει.

Για να ελέγξουμε αν δύο λίστες περιέχουν τα ίδια στοιχεία χρησιμοποιούμε την cmp.

```
a=[1,2,3,4,5]
b=[1,2,3,4,5]
c=[1,2,3]
print cmp(a,b)
print cmp(a,c)
print cmp(c,a)
```

Προσοχή: Η cmp δεν επιστρέφει True/False, αλλά ένα αριθμό. Το 0 σημαίνει ότι οι λίστες έχουν τα ίδια στοιχεία, -1 ή 1 ότι τα στοιχεία δεν είναι ίσα.

Η Python διαθέτει τη συνάρτηση sort και την sorted για να βάζει στη σειρά τα στοιχεία μίας λίστας. Προσέξτε τη διαφορά στη χρήση τους:

```
a=[27,35,14,95,11]
b=[45,8,99,11,3,0]
a.sort()
print a
c=sorted(b)
print b
print c
```

Θα μπορούσαμε επιπλέον να αντιστρέψουμε τη σειρά:

```
a=[27,35,14,95,11]
b=[45,8,99,11,3,0]
a.sort(reverse=True)
print a
c=sorted(b,reverse=True)
print b
print c
```

```
fin=open("a.txt","r")
lst=[]
for line in fin:
    line=line.strip()
    lst.append( line)
headers= lst[0].split(",")
values= lst[-1].split(",")
tmp="<html><title>Meteo</title><body><h1>Meteo</h1>"
for i in range(len(headers)):
    tmp+= "%s : %s<br>" %(str(headers[i]),str(values[i]))
tmp+="</body></html>"
fout=open("out.html","w")
fout.write(tmp)
fout.close()
```
