

Λίστες

Σε αυτό το σημείο θα μάθουμε τι είναι οι λίστες, πώς ορίζονται, πως χρησιμοποιούνται και διάφορα tips για να μην δημιουργηθούν σφάλματα!

Τι είναι οι λίστες;

- Η λίστα είναι μια δομή δεδομένων η οποία περιέχει μια σειρά από δεδομένα του ίδιου τύπου.

Σύνταξη

Ο τρόπος σύνταξης και δημιουργίας μια νέας λίστας είναι:

όνομα_λίστας = [] //πρόκειται για μια κενή λίστα

Μπορούμε να αρχικοποιήσουμε μια λίστα και να της δώσουμε ορίσματα:

όνομα_λίστας = [1, 3, 5, 6] //λίστα ακεραίων

όνομα_λίστας = ["Giannis", "Dimitris"]

Προσπέλαση Στοιχείων - 1

- Αρχικά πρέπει να πούμε πως η αρίθμηση των στοιχείων σε μια λίστα ξεκινά από το 0 και όχι το 1!
- Η σύνταξη που χρησιμοποιούμε για να αναφερθούμε σε ένα συγκεκριμένο στοιχείο στην λίστα είναι η εξής:

Προσπέλαση Στοιχείων - 1

```
//Αρχικοποιούμε την λίστα μας  
  
my_list = [10, 20, 50, 60]  
  
// αναφερόμαστε στο πρώτο στοιχείο  
// της λίστας, δηλαδή τον αριθμό 10  
  
print(my_list[0])  
  
//αναφερόμαστε στο 3ο στοιχείο της  
//λίστας, δηλαδή τον αριθμό 50  
  
print(my_list[2])
```

Απεικόνιση Λιστών

<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>
10	20	50	60

Προσπέλαση Στοιχείων - 2

- **Προσοχή:** Ένα συχνό λάθος που γίνεται είναι ότι όταν θέλουμε να αναφερθούμε στο τελευταίο στοιχείο της λίστας βάζουμε το πλήθος των στοιχείων του πίνακα
- Ακουλουθεί παράδειγμα με την λίστα `my_list`

Προσπέλαση Στοιχείων - 2

- Θέλουμε να αναφερθούμε στο τελευταίο στοιχείο της λίστας δηλαδή στον αριθμό 60!

```
my_list = [10, 20, 50, 60]
```

```
//Λάθος Αναφορά
```

```
my_list[4]
```

```
//Σωστή Αναφορά
```

```
my_list[3]
```


Προσπέλαση Στοιχείων - 2

- Η πρώτη αναφορά είναι λάθος γιατί ναι μεν θέλουμε να αναφερθούμε στο 4^ο στοιχείο της λίστας αλλά χρησιμοποιούμε τον αριθμό 3 καθώς η αρίθμηση ξεκινά από το 0.

Προσπέλαση Στοιχείων - 3

- Μπορούμε να διατρέξουμε όλα τα στοιχεία της λίστας χρησιμοποιώντας μια επαναληπτική δομή. Ως επαναληπτική δομή μπορούμε να χρησιμοποιήσουμε την `for loop`
- Ο κώδικας δίνεται στην επόμενη διαφάνεια...

Προσπέλαση Στοιχείων - 3

```
my_list = [10, 30, 50, 60]  
for i in my_list:  
    print(i)
```

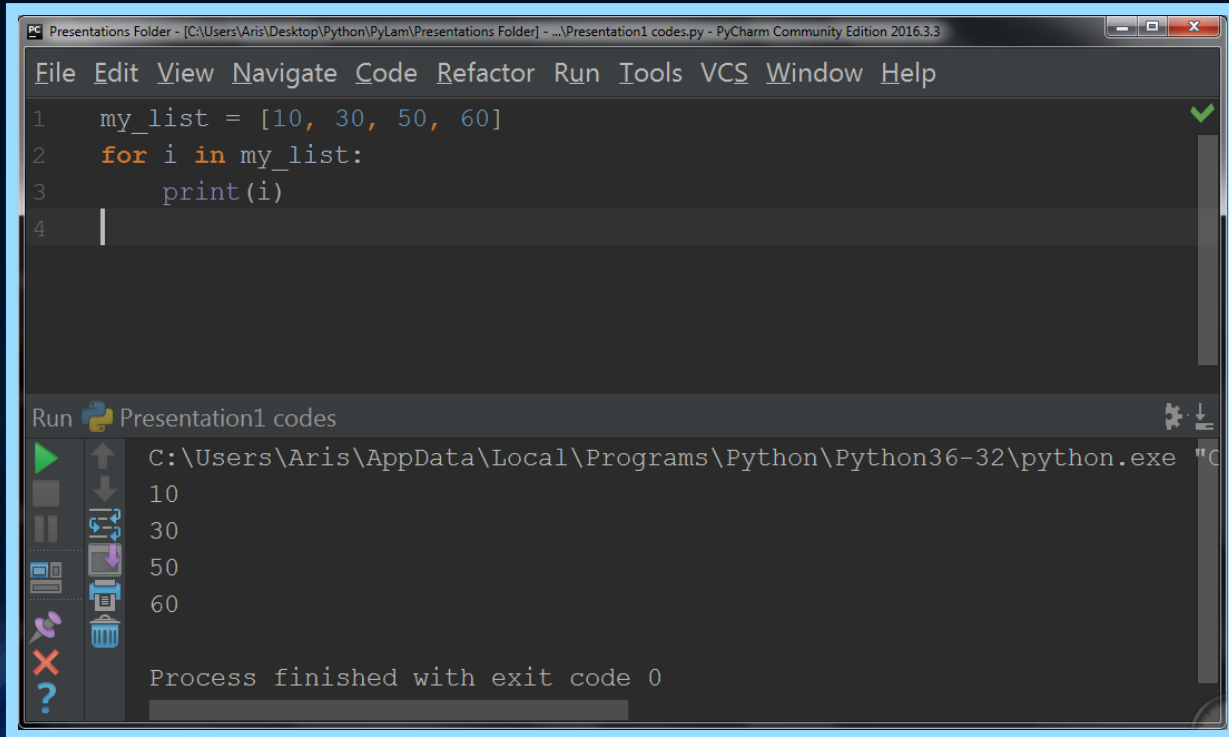
Προσπέλαση Στοιχείων - 3

- Γιατί συντάσσεται έτσι η for;

Ο λόγος είναι ότι χρειαζόμαστε έναν μετρητή για να μπορούμε να ανφερθούμε κάθε φορά σε ένα μόνο στοιχείο στην λίστα.

Έτσι, ο προηγούμενος αλγόριθμος θα εμφανίσει το εξής:

Προσέλαση Στοιχείων - 3



The image shows a PyCharm IDE window titled "Presentations Folder - [C:\Users\Aris\Desktop\Python\PyLam\Presentations Folder] - ...\Presentation1 codes.py - PyCharm Community Edition 2016.3.3". The main editor displays a Python script with four lines of code. A green checkmark is visible in the right margin next to the first line. Below the editor is a "Run" toolbar with a green play button icon. The "Run" panel shows the command "C:\Users\Aris\AppData\Local\Programs\Python\Python36-32\python.exe" and the output of the script, which lists the elements of the list: 10, 30, 50, and 60. At the bottom, it states "Process finished with exit code 0".

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
1 my_list = [10, 30, 50, 60]
2 for i in my_list:
3     print(i)
4
```

Run Presentation1 codes

C:\Users\Aris\AppData\Local\Programs\Python\Python36-32\python.exe "C:\Users\Aris\Desktop\Python\PyLam\Presentations Folder\Presentation1 codes.py"

10
30
50
60

Process finished with exit code 0

Εισαγωγή Στοιχείων - 1

- Θα χρειαστούμε την συνάρτηση **append()**.

Παράδειγμα:

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

```
my_list.append(10)
```

Προσθέτει τον αριθμό 10 στο τέλος της λίστας

Εισαγωγή Στοιχείων - 2

- Η λίστα μετά την κλήση της συνάρτησης `append()` θα είναι:
 - `my_list` → `[2, 4, 6, 8, 10]`
- **Σημείωση:** Μπορούμε να εισάγουμε ένα στοιχείο σε μια λίστα ακόμα και αν είναι τύπου `string`, `boolean`, `float`...
- Δεν περιορίζεται σε λίστες που περιέχουν μόνο ακραίους αριθμούς.

Αφαίρεση Στοιχείων

- Αντίστοιχα υπάρχει η συνάρτηση **remove()**
- Παράδειγμα:
 - `my_list = [4, 7, 17, 25]`
 - `my_list.remove(17)`

Οπότε το 17 έχει αφαιρεθεί από την λίστα

- **Προσοχή:** Δεν αναφερόμαστε στο στοιχείο που βρίσκεται στην 17^η θέση του πίνακα αλλά στο στοιχείο ίσο με 17