ΠΕΡΙΕΧΟΜΕΝΑ:

1. Ορισμός Λίστας
2. Αρίθμηση Στοιχείων και Μέρος Λίστας
3. Τροποποίηση Λίστας
4. Ύπαρξη Στοιχείου, Μήκος Λίστας και άλλα…

Πασχάλης Μ.

Χρυσός Χορηγός Μαθήματος

Ιωάννης Φιλιππίδης

Χάλκινος Χορηγός Μαθήματος

ΜΑΘΗΜΑ 4: Εισαγωγή στις λίστες

1. Ορισμός Λίστας

python 3 psounis

- **Λίστα** είναι μία συλλογή μεταβλητών με μία συγκεκριμένη σειρά
  - Οι μεταβλητές μπορεί να είναι οποιουδήποτε τύπου δεδομένων.
  - Προσοχή, ότι υπάρχει **διάταξη** (εκτός από το ποια αντικείμενα αποθηκεύονται, μας ενδιαφέρει και η σειρά με την οποία αυτά αποθηκεύονται.

- **Δήλωση λίστας**
  - Ορίζουμε μια λίστα χρησιμοποιώντας κάποιο όνομα μεταβλητής και ενθέτοντας τις τιμές σε αγκύλες.

Παράδειγμα 1: lists.py

Στο παράδειγμα βλέπουμε τον ορισμό 4 λιστών:

```
list_int = [1, 3, 4]
list_float = [3.12, 5.11, 1.1]
list_string = ["My", "name", "is"]
list_collection = [1, "big", [1, 2]]

print(list_collection)
print(type(list_int))
```

Παρατηρείστε ότι η λίστα είναι ακόμη ένας τύπος δεδομένων της Python.

- **Στοιχείο λίστας**
  - Οι μεταβλητές στη λίστα ονομάζονται στοιχεία της λίστας.
  - Η αρίθμηση των στοιχείων ξεκινά από το 0
  - Έχουμε πρόσβαση σε κάποιο στοιχείο βάζοντας το όνομα της λίστας, ακολουθούμενο απο το δείκτη (αριθμό, index) του στοιχείου σε αγκύλες.
  - Η συμπεριφορά ενός στοιχείου της λίστας είναι ακριβώς ίδια με αυτή μιας μεταβλητής.

Παράδειγμα 2: list.element.py

```
my_grades = [5, 8, 6, 7]
print(my_grades)
my_grades[1] = 9
print(my_grades)
print(my_grades[0])
print(my_grades[3])
```

Σημείωση:

- Οι λίστες είναι το ισοδύναμο του πίνακα σε άλλες γλώσσες προγραμματισμού

Άσκηση 1: Οι φιλίες δεν κρατάνε για πάντα

- Κατασκευάστε μία λίστα με τους 3 καλύτερους σας φίλους.
- Τυπώστε την.
- Αντικαταστήστε τον φίλο που έχετε αποθηκεύσει στην 1η θέση του πίνακα, με κάποιον άλλο.
- Τυπώστε εκ νέου τη λίστα
- Τυπώστε ένα-ένα τα στοιχεία της λίστας.

Άσκηση 2: Ο μέγιστος (ξανά)

Κατασκευάστε ένα πρόγραμμα το οποίο:

- Θα ορίζει μία λίστα τριών ακεραίων
- Θα υπολογίζει το μέγιστο στοιχείο της λίστας και θα το τυπώνει

2. Αρίθμηση Στοιχείων και Μέρος Λίστας

- Έχουμε πρόσβαση στα στοιχεία μιας λίστας N στοιχείων
  - είτε με την **"ευθεία" αρίθμηση** (από αριστερά προς τα δεξιά): 0,1,2,…, N-1
  - είτε με την **"αντίστροφη" αρίθμηση** (από δεξιά προς τα αριστερά): -1, -2, …, -N

Παράδειγμα 3: reverse.indexing.py

Βλέπουμε στο παράδειγμα την πρόσβαση στα στοιχεία με την αντίστροφη αρίθμηση:

```
my_list = [1, 3, 5]
print(list[-1])

my_list[-2] = 6
print(my_list)

my_list[-3] = 4
print(my_list)
```

Προσοχή:

- Είναι ευθύνη του προγραμματιστή να μη βάλει δείκτη εκτός των ορίων της λίστας.
- Αλλιώς η Python θα "πετάξει" σφάλμα, ότι προσπαθούμε να έχουμε πρόσβαση εκτός των ορίων της λίστας.

- **Μέρος Λίστας**
- Μας δίνεται η δυνατότητα να κατασκευάσουμε μια νέα λίστα, από μία ήδη υπάρχουσα με τους εξής τρόπους:
  - `list_name[S:F]`
    Επιστρέφει τη λίστα από το στοιχείο με δείκτη S έως το F-1
  - `list_name[S:]`
    Ομοίως από το S έως το τέλος της λίστας
  - `list_name[:F]`
    Ομοίως από την αρχή έως το στοιχείο με δείκτη F-1
  - `list_name[:]`
    Επιστρέφει όλη τη λίστα
- Αντίστοιχα εργαζόμαστε και με την αντίστροφη αρίθμηση, π.χ.:
  `list_name[-4:-2]`

Παράδειγμα 4: list.slicing.py

```
my_list = ["a", "b", "c", "d", "e"]

my_new_list = my_list[1:4]
print(my_new_list)

my_newest_list = my_list[:]
print(my_newest_list)
```

Άσκηση 3: Μπρός - πίσω

- Κατασκευάστε μια λίστα με 10 στοιχεία της αρεσκείας σας.
- Τυπώστε τα στοιχεία που βρίσκονται στις θέσεις από 3 έως 6 με τρεις τρόπους:
  - Ένα – ένα χρησιμοποιώντας την ευθεία αρίθμηση.
  - Ένα – ένα χρησιμοποιώντας την αντίστροφη αρίθμηση.
  - Όλα μαζί, αποκόπτωντας το κατάλληλο κομμάτι της αρχικής λίστας (τυπώστε απευθείας το μέρος, χωρίς τη χρήση κάποιας βοηθητικής λίστας)

Άσκηση 4: Μέσος όρος

Στις εκλογές του 1844 (πρώτες κοινοβουλευτικές εκλογές), οι έδρες των κομμάτων ήταν:

- Ρωσικό Κόμμα: 55 έδρες
- Αγγλικό Κόμμα: 28 έδρες
- Γαλλικό Κόμμα: 20 έδρες

Υπολογίστε το μέσο όρο των εδρών των δύο πρώτων κομμάτων και τυπώστε τον:

- Κάνοντας αποκοπή του μέρους της λίστας που μας ενδιαφέρει.
- Έχοντας πρόσβαση στα δεδομένα μέσω της λίστας.

Υπενθύμιση: Ο μέσος όρος N αριθμών υπολογίζεται διαιρώντας το άθροισμα των αριθμών με το N.

3. Τροποποίηση Λίστας

- Μπορούμε να χρησιμοποιήσουμε ειδικές **μεθόδους** για να προσθέσουμε στοιχεία σε μια λίστα (με τον τελεστή .)
  - list_name.**append**(element): Προσθέτει το στοιχείο element στο τέλος της λίστας list_name
  - list_name.**insert**(index, element): Προσθέτει το στοιχείο element στη θέση index της λίστας list_name
  - list_name.**extend**(another_list): Προσθέτει τη λίστα another_list στο τέλος της λίστας list_name

Παράδειγμα 5: append.py

```
my_list = [1, 2, 3]
my_list.append(4)
print(my_list)
```

Παράδειγμα 6: insert.py

```
my_list = [1, 2, 3]
my_list.insert(1, 4)
print(my_list)
```

Παράδειγμα 7: extend.py

```
my_list = [1, 2, 3]
my_list.extend([5, 8])
print(my_list)
```

- και αντίστοιχα για να αφαιρέσουμε στοιχεία από τη λίστα:
  - list_name.**pop**(index): Διαγράφει και Επιστρέφει το στοιχείο element στη θέση index της λίστας list_name (ή το τελευταίο αν δεν καθοριστεί δείκτης).
  - list_name.**remove**(value): Διαγράφει το πρώτο στοιχείο με τιμή value της λίστας list_name
  - list_name.**clear**(): Διαγράφει όλα τα στοιχεία της λίστας list_name

Παράδειγμα 8: pop.py

```
my_list = [1, 2, 3]
last = my_list.pop()
print(last)
print(my_list)
my_list.pop(0)
print(my_list)
```

Παράδειγμα 9: remove.py

```
my_list = [1, 2, 3, 2, 4]
my_list.remove(2)
print(my_list)
```

Παράδειγμα 10: clear.py

```
my_list = [1, 2, 3]
my_list.extend([5, 8])
print(my_list)
```

Σημείωση: Η λίστα είναι μία κλάση (class) που έχει μεθόδους (methods) στις οποίες έχουμε πρόσβαση με την τελεία. Θα μάθουμε πολλά περισσότερα, όταν μελετήσουμε τον αντικειμενοστρεφή προγραμματισμό στην Python (Object Oriented Programming – OOP)

Άσκηση 5: Μια ουρά σε ένα ταμείο

- Κατασκευάστε μία κενή λίστα με όνομα cash_desk
- Προσθέστε στην ουρά τον Tom
- Προσθέστε στην ουρά τον Bob
- Τυπώστε την ουρά
- Αφαιρέστε το 1ο άτομο της ουράς, και τυπώστε ότι εξυπηρετήθηκε.
- Προσθέστε στην ουρά την Pam
- Προσθέστε στην ουρά τον Jim
- Τυπώστε την ουρά
- Αφαιρέστε το 1ο άτομο της ουράς, και τυπώστε ότι εξυπηρετήθηκε.
- Τυπώστε την ουρά

Ελαφρώς Προχωρημένη Άσκηση: Συμβουλευθείτε το βίντεο.

Άσκηση 6: Μία στοίβα εγγράφων

Σε ένα μη αποδοτικό γραφείο εξυπηρέτησης πολιτών, ο υπάλληλος έχει μία στοίβα από αιτήσεις και:

- Όποιο αίτημα του έρχεται το βάζει πάνω πάνω στη στοίβα
- Όταν αποφασίσει να διεκπεραιώσει κάποιο αίτημα, αρπάζει αυτό που βρίσκεται πάνω πάνω στη στοίβα

Κατασκευάστε πρόγραμμα που μοντελοποιεί το μόχθο του:

- Η στοίβα αρχικά είναι κενή.
- Έρχεται το αίτημα του Bob.
- Έρχεται το αίτημα του John.
- Τυπώστε τη στοίβα
- Εξυπηρετεί το τελευταίο αίτημα, τυπώνοντας ποιον αφορά
- Έρχεται το αίτημα της Pat
- Τυπώστε τη στοίβα

Ελαφρώς Προχωρημένη Άσκηση: Συμβουλευθείτε το βίντεο.

4. Ύπαρξη Στοιχείου, Μήκος Λίστας και άλλα

- **Ύπαρξη Στοιχείου**
  - Ελέγχουμε αν ένα στοιχείο υπάρχει σε μια λίστα με τον τελεστή in:

    ```
    ELEMENT in LIST
    ```

    Επιστρέφει True / False ανάλογα με το αν το στοιχείο υπάρχει στη λίστα
  - Και αντίστοιχα ότι ένα στοιχείο δεν υπάρχει σε μια λίστα με τη σύντμηση not in:

    ```
    ELEMENT not in LIST
    ```

Παράδειγμα 11: in.py

Οι δύο αυτοί τελεστές συνήθως βρίσκονται μέσα σε κάποια δομή ελέγχου, π.χ.:

```
my_list = [1, 2, 3]

if 4 in my_list:
    print("exists")
else:
    print("not exists")
```

Κι άλλη λειτουργικότητα επί της λίστας:

- Η συνάρτηση len επιστρέφει πόσα στοιχεία έχει η λίστα (μήκος λίστας):

  ```
  len(LIST)
  ```

- Η μέθοδος sort ταξινομεί σε αύξουσα σειρά τα στοιχεία της λίστας

  ```
  LIST.sort()
  ```

- Η μέθοδος reverse αντιστρέφει τη σειρά των στοιχείων της λίστας:

  ```
  LIST.reverse()
  ```

Παράδειγμα 12: more.methods.py

```
my_list = [1, 2, 5]
print(my_list)
my_list.reverse()
print(my_list)
my_list.sort()
print(my_list)
print(len(my_list))
```

Άσκηση 7: Ταξινόμηση πραγματικών

Κατασκευάστε πρόγραμμα το οποίο:

- Δέχεται από την είσοδο τρεις πραγματικούς και τους αποθηκεύει σε μία λίστα
- Ταξινομεί τη λίστα
- Τυπώνει τη λίστα

Άσκηση 8: Λίστα Αγαπημένων Ταινιών

Κατασκευάστε ένα πρόγραμμα το οποίο:

- Θα αρχικοποιεί μία λίστα με 4 αγαπημένες σας ταινίες

Έπειτα:

- Θα ζητάει από το χρήστη να εισάγει μία νέα αγαπημένη του ταινία.
- Αν η ταινία υπάρχει ήδη στη λίστα, θα ενημερώνει το χρήστη ότι δεν έγινε η αποθήκευση
- Αν η ταινία δεν υπάρχει στη λίστα, θα την προσθέτει, θα ταξινομεί τη λίστα και έπειτα θα τυπώνει τη λιστά, καθώς και το πλήθος των αγαπημένων ταινιών του χρήστη.

"Now is better than never."

Zen of Python #15