

Δομές Δεδομένων

Τι είναι οι δομές δεδομένων

- Είναι τρόποι αποθήκευσης και οργάνωσης δεδομένων.
- Βοηθούν στην αποτελεσματική διαχείριση πληροφοριών.
- Υπάρχουν **στατικές** και **δυναμικές** δομές δεδομένων.

♦ Στατικές Δομές Δεδομένων

Έχουν σταθερό μέγεθος, δηλαδή δεν αλλάζει κατά την εκτέλεση του προγράμματος.

Παράδειγμα: **Tuple (tuple)**

♦ Δυναμικές Δομές Δεδομένων

Μπορούν να μεγαλώνουν ή να μικραίνουν καθώς το πρόγραμμα εκτελείται.

Παράδειγμα: **Λίστα (List), Σύνολο (Set), Λεξικό (Dictionary)**

Δομές Δεδομένων

Διαφορές Στατικών & Δυναμικών Δομών Δεδομένων

Χαρακτηριστικό	Στατικές Δομές Δεδομένων	Δυναμικές Δομές Δεδομένων
Μέγεθος	Σταθερό	Μεταβαλλόμενο
Τύποι στην Python	tuple	list, set, dict
Χρήση μνήμης	Σταθερή	Δυναμικά προσαρμόσιμη
Τροποποίηση	Δεν επιτρέπεται	Επιτρέπεται (προσθήκη/διαγραφή στοιχείων)

Δομές Δεδομένων

Λίστες

Τι είναι οι Λίστες στην Python;

- Λίστες είναι συλλογές από αντικείμενα, όπως αριθμοί ή συμβολοσειρές, και χρησιμοποιούνται για να αποθηκεύουμε πολλά δεδομένα.
- Μια λίστα μπορεί να έχει διαφορετικά είδη δεδομένων (π.χ. αριθμούς, λέξεις, ακόμα και άλλες λίστες!).
- Στην Python, οι λίστες γράφονται με αγκύλες: [].

Δομές Δεδομένων

Λίστες

Πως Δημιουργούμε Μια Λίστα;

Δημιουργία μιας απλής λίστας

```
my_list = [1, 2, 3, "Python", 4.5]
```

```
print(my_list) # Εκτυπώνει: [1, 2, 3, 'Python', 4.5]
```

- Τα στοιχεία μιας λίστας ξεκινούν από το δείκτη 0 (π.χ. το πρώτο στοιχείο είναι στο δείκτη 0).

Δομές Δεδομένων

Λίστες

Αλλαγή Στοιχείων Λίστας

- Μπορούμε να αλλάξουμε ένα στοιχείο της λίστας με το δείκτη του (index).

Παράδειγμα:

```
fruits = ["apple", "banana", "cherry"]  
fruits[1] = "blueberry" # Αλλάζουμε το "banana" σε "blueberry"  
print fruits # ["apple", "blueberry", "cherry"]
```

Δομές Δεδομένων

Λίστες

Μέγεθος Λίστας & έλεγχος ύπαρξης αντικειμένου

- Μπορούμε να βρούμε το μέγεθος της λίστας με τη συνάρτηση **len**!

παράδειγμα :

```
fruits = ["apple", "banana", "cherry"]  
len(fruits) # 3
```

- Επίσης μπορούμε να ελέγξουμε αν ένα στοιχείο περιέχεται σε μία λίστα χρησιμοποιώντας τον τελεστή **in**

παράδειγμα:

```
fruits = ["apple", "banana", "cherry"]  
"banana" in fruits # True
```

Δομές Δεδομένων

Λίστες

Εργασία

Γράψε ένα πρόγραμμα που:

1. Δημιουργεί μια λίστα με 5 αγαπημένα φαγητά.
2. Εκτυπώνει ολόκληρη τη λίστα.
3. Αλλάζει το δεύτερο φαγητό με ένα άλλο που θα δώσει ο χρήστης.
4. Εκτυπώνει ξανά τη λίστα με την αλλαγή.
5. Εμφανίζει το μέγεθος (πλήθος στοιχείων) της λίστας.
6. Ζητά από τον χρήστη ένα φαγητό και ελέγχει αν υπάρχει στη λίστα, εμφανίζοντας το κατάλληλο μήνυμα.

Δομές Δεδομένων

Λίστες

Προσθήκη στοιχείου

- Η συνάρτηση **append()** προσθέτει ένα νέο στοιχείο στο τέλος της λίστας.
- Η συνάρτηση **insert(index, item)** προσθέτει ένα στοιχείο σε μια συγκεκριμένη θέση (δείκτη) μέσα στη λίστα.

Παράδειγμα:

```
fruits.append("orange") # Προσθέτουμε το "orange"  
print fruits # ["apple", "blueberry", "cherry", "orange"]
```

```
fruits.insert(1, "orange") # Εισάγουμε το "orange" στη θέση 1  
print fruits # ["apple", "orange", "banana", "cherry"]
```

Δομές Δεδομένων

Λίστες

Αφαίρεση στοιχείου

- Η συνάρτηση `pop()` αφαιρεί και επιστρέφει το στοιχείο από τη λίστα στη θέση που υποδεικνύουμε (ή το τελευταίο αν δεν καθορίσουμε θέση).

Παράδειγμα:

```
fruits = ["apple", "banana", "cherry"]  
removed_fruit = fruits.pop(1) # Αφαιρούμε το στοιχείο στη θέση 1 και το  
επιστρέφουμε  
print fruits # ["apple", "cherry"]  
print removed_fruit # "banana"
```

Δομές Δεδομένων

Λίστες

Εργασία

Γράψε ένα πρόγραμμα που:

1. Δημιουργεί μια κενή λίστα ταινίες.
2. Ζητά από τον χρήστη να προσθέσει 3 αγαπημένες του ταινίες χρησιμοποιώντας `append()`.
3. Ζητά από τον χρήστη να προσθέσει μία ακόμη ταινία σε συγκεκριμένη θέση (που θα επιλέξει) χρησιμοποιώντας `insert()`.
4. Εκτυπώνει τη λίστα με όλες τις ταινίες.
5. Ζητά από τον χρήστη να αφαιρέσει μια ταινία από συγκεκριμένη θέση χρησιμοποιώντας `pop()`.
6. Εκτυπώνει τη λίστα μετά την αφαίρεση.