



ΠΕΡΙΕΧΟΜΕΝΑ:

1. Αρχικοποίηση Πίνακα
2. Συμπεριφορά ως λίστα και αναζήτηση
3. Αποθήκευση σε αρχείο και Μετατροπές

ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ:

1. Python Advanced 2 - Iterators

Ανδρέας Γ.

Σμαραγδένιος Χορηγός Μαθήματος

Δημήτρης Δ.

Σμαραγδένιος Χορηγός Μαθήματος

To module array περιέχει:

- Υλοποιήσεις αριθμητικών πινάκων, οι οποίοι περιέχουν ομογενή δεδομένα (ίδιος τύπος δεδομένων)
- Θεωρούνται καλύτεροι (λιγότερη μνήμη - πιο γρήγορη επεξεργασία) από μία απλή λίστα, όταν έχουμε πραγματικά μεγάλους πίνακες (π.χ. δεκάδες χιλιάδες στοιχεία), γιατί αποθηκεύουν μόνο τα bytes του τύπου δεδομένων (και όχι αντικείμενα).
- Προσφέρεται διεπαφή αντίστοιχη της λίστας με αρκετές έξτρα μεθόδους.

- Κατασκευάζουμε έναν πίνακα με τη συνάρτηση:

Συνάρτηση	Επεξήγηση
<code>array(typecode[, initializer])</code>	<code>typecode</code> = κωδικός τύπου δεδομένων <code>initializer</code> = iterable με δεδομένα

- Εναλλακτικά, αφού έχουμε κατασκευάσει ένα (κενό) πίνακα με κάποιο τύπο δεδομένων, μπορούμε να γεμίσουμε τον πίνακα με τις μεθόδους:

Συνάρτηση	Επεξήγηση
<code>fromlist(list)</code>	Προσαρτά τα περιεχόμενα της λίστας
<code>fromunicode(str)</code>	Προσαρτά τους χαρακτήρες της συμβ/ρας
<code>frombytes(s)</code>	s είναι συμβολοσειρά που διαβάζεται ως bytes

- Είναι δική μας προγραμματιστική ευθύνη να δουλεύουμε με δεδομένα ίδιου τύπου.

Τύποι Δεδομένων:

Κωδικός	C	Python	bytes
'b'	signed char	int	1
'B'	unsigned char	int	1
'h'	signed short	int	2
'H'	unsigned short	int	2
'i'	signed int	int	2
'I'	unsigned int	int	2
'L'	unsigned long	int	4
'q'	signed long long	int	8
'Q'	unsigned long long	int	8
'f'	float	float	4
'd'	double	float	8

Παράδειγμα 1: array init

```
from array import array
a = array('i', [1, 2, 3])
a.fromlist([4,5,6])
print(a)

a = array('d', [1, 2, 3])
a.fromlist([4,5,6])
print(a)

a = array('u', "abc") # deprecated
a.fromunicode("αβγ")
print(a)
```

Σημείωση:

- Το numPy είναι de facto standard για αριθμητικούς πίνακες

- Οι παρακάτω μέθοδοι δίνουν συμπεριφορά στους πίνακες αντίστοιχη με της λίστας:

Μέθοδος	Επεξήγηση
append(item)	Προσθέτει το item στο τέλος του πίνακα
insert(pos, item)	Εισάγει στη θέση pos το item (σπρώχνοντας τα υπόλοιπα μία θέση δεξιά)
extend(iter)	Επεκτείνει τη λίστα με τα στοιχεία του iter(able)
pop([pos])	Αφαιρεί και επιστρέφει το στοιχείο στη θέση pos (default: το τελευταίο)
remove(item)	Αφαιρεί και επιστρέφει την πρώτη εμφάνιση του item στον πίνακα

Παράδειγμα 2: as list.py

```
arr = array('i', (randrange(100) for _ in range(10)))  
print(arr)  
arr.append(12)  
print(arr)  
arr.remove(12)  
print(arr)  
elem = arr.pop()  
print(elem, arr)  
# arr.clear() doesn't work  
print(arr)
```

- μέθοδοι για αναζήτηση στοιχείων στον πίνακα:

Μέθοδος	Επεξήγηση
index(elem [, start[, stop]])	Επιστρέφει τη θέση της πρώτης εμφάνισης του elem στον πίνακα. Αν δεν υπάρχει το στοιχείο, προκαλεί ValueError (προαιρετικά ορίσματα: αναζήτηση σε συγκεκριμένο εύρος θέσεων του πίνακα)
count(elem)	Επιστρέφει το πλήθος των εμφανίσεων του elem στον πίνακα.

Παράδειγμα 3: searching.py

```
arr = array('i', (randrange(100) for _ in range(1000)))  
  
print(arr)  
  
element = 12  
occurrences = arr.count(element)  
print("occurrences=" + str(occurrences))  
pos = 0  
for _ in range(occurrences):  
    pos = arr.index(element, pos)  
    print("pos = " + str(pos))  
    pos = pos + 1
```

- Οι παρακάτω μέθοδοι δίνουν έναν εύκολο τρόπο αποθήκευσης και διαβάσματος σε/από αρχείο.

Μέθοδος	Επεξήγηση
fromfile(fob, n)	Διαβάζει η στοιχεία από το αρχείο που έχει ανοιχτεί και περιγράφεται από το χειριστή fob. Αν το αρχείο περιέχει λιγότερα από η στοιχεία, προκαλείται εξαίρεση EOFError, αλλά τα στοιχεία που διαβάστηκαν εισάγονται κανονικά στον πίνακα.
tofile(fob)	Γράφει όλο τον πίνακα στο αρχείο

Παράδειγμα 4: array_to_file.py

```
arr = array('i', (randrange(100) for _ in range(10**6)))
some_data = arr[:100]
print(some_data)

with open("data.bin", "wb") as f:
    arr.tofile(f)

with open("data.bin", "rb") as f:
    arr.fromfile(f, 10**6)
some_data = arr[:100]
print(some_data)
```

- Μετατροπή από/σε bytes:

Μέθοδος	Επεξήγηση
frombytes(b)	b είναι τύπου bytes
tobytes()	Μετατρέπει σε bytes το περιεχόμενο του πίνακα

- Μετατροπή από/σε λίστα:

Μέθοδος	Επεξήγηση
fromlist(list)	Μετατροπή λίστας σε array
tolist()	Μετατροπή array σε λίστα

Παράδειγμα 5: conversions.py

```
from array import array
from random import randrange

arr = array('i', (randrange(100) for _ in range(50)))

print("initial array: " + str(arr))

l = arr.tolist()
print("as list: " + str(l))

bytes = arr.tobytes()
print("as bytes: " + str(bytes))

reconstructed = array('i')
arr.frombytes(bytes)
print("reconstructed: " + str(arr))
```