

Κεφάλαιο 8

- Γνωριμία με το νέο περιβάλλον
 - Εισαγωγή στο 8

η ύλη του 8

ΚΕΦΑΛΑΙΟ 8. Δομές Δεδομένων II

8.1 Συμβολοσειρές (strings).

8.2 Λίστες.

8.3 Στοίβα.

8.4 Ουρά.

8.8 Δραστηριότητες.

8.9 Ερωτήσεις.

(Από τις παραγράφους 8.8 και 8.9, μόνο

Strings

- Αλφαριθμητικά
- Συμβολοσειρές
- String

Παράδειγμα

```
a = "This is a string"
```

Παράδειγμα

```
a = 'This is a string'
```

Παράδειγμα

```
a = "This is a string"
```

Παράδειγμα

`a = "" This is a string ""`

Παράδειγμα

```
>>> a = """This is
... a string
... with new lines!"""
>>> print a
This is
a string
with new lines!
>>> a
'This is \na string\nwith new lines!'
>>>
```


Χαρακτηριστικά

Τα **αλφαριθμητικά** ή **συμβολοσειρές** στην Python είναι ακολουθίες από χαρακτήρες που έχουν **σταθερό μέγεθος** και **μη μεταβαλλόμενα περιεχόμενα**. Δηλαδή, δεν μπορούμε να προσθέσουμε ή να αφαιρέσουμε χαρακτήρες, ούτε να τροποποιήσουμε τα περιεχόμενα του αλφαριθμητικού. Γι' αυτό λέμε ότι η δομή αυτή ανήκει στις **μη μεταβαλλόμενες (immutable) δομές** της Python. **Η αρίθμηση των χαρακτήρων σε ένα αλφαριθμητικό ξεκινάει από το 0**. Για παράδειγμα: αν `word = "PYTHON"`, η αναπαράσταση του αλφαριθμητικού μοιάζει με το παρακάτω σχήμα:

0	1	2	3	4	5
'P'	'Y'	'T'	'H'	'O'	'N'

μη μεταβαλλόμενη δομή immutable (1/3)

```
>>> a = "this is the one"
```

```
>>> b = a
```

```
>>> id(a)
```

```
139663146042160
```

```
>>> id(b)
```

```
139663146042160
```

μη μεταβαλλόμενη δομή immutable (2/3)

```
>>> a += "!"
```

```
>>> b
```

```
'this is the one'
```

```
>>> a
```

```
'this is the one!'
```

```
>>>
```

μη μεταβαλλόμενη δομή immutable (2/3)

```
>>> a += "!"
```

```
>>> b
```

```
'this is the one'
```

```
>>> a
```

```
'this is the one!'
```

```
>>>
```

μη μεταβαλλόμενη δομή immutable (3/3)

```
>>> id(a)
139663146042096
>>> id(b)
139663146042160
>>>
```

Αρίθμηση: από το 0

```
>>> a = "Hello!"
```

```
>>> a[0]
```

```
'H'
```

```
>>> a[5]
```

```
'!'
```

String slicing (κομμάτισμα) 1/2

```
>>> a = "pizza donut beer"  
>>> b = a[6:11]  
>>> b  
'donut'
```

String slicing (κομμάτισμα) 2/2

```
>>> a = "pizza donut beer"  
>>> c = a[6:11:2]  
>>> c  
'dnt'
```


String slicing

```
c = a[6:11:2]
```

```
c = a[ΑΠΟ:ΜΕΧΡΙ:ΒΗΜΑ]
```

(όπως η `range(6, 11, 2)`)

String slicing (κομμάτισμα) 3

```
>>> a = "pizza donut beer"  
>>> c = a[_____ : _____ : ____]  
>>> c  
'reeb'
```

String slicing (κομμάτισμα) 3

```
>>> a = "pizza donut beer"  
>>> c = a[len(a)-1:len(a)-5:-1]  
>>> c  
'reeb'
```

Χαρακτηριστικά

- Μη μεταβαλλόμενα περιεχόμενα (immutable)
 - Σταθερό μέγεθος
- Αρίθμηση

Με τις λίστες τι ισχύει;

- Είναι παρόμοια η αρίθμηση;
 - Ισχύει το κομμάτιασμα στις λίστες;
- Έχουν μη μεταβαλλόμενα περιεχόμενα;

String: Χρήσιμα (1/3)

- `str`
 - Μετατροπή **σε** string: `str(2020)` -> "2020"
- `float, int`
 - Μετατροπή **από** string:
 - `float("3.14")` -> 3.14
 - `int("4")` -> 4

String: Χρήσιμα (2/3)

- `len`
 - Μήκος του string: `len("Hello!")` -> 6
- Πρόσθεση δύο string
 - `"new" + " " + "year"` -> `"new year"`
- Πολλαπλασιασμός string με αριθμό
 - `"!" * 3` -> `"!!!"`
 - `3 * "!"` -> `"!!!"`

String: Χρήσιμα (3/3)

- Έλεγχος `>` , `<` , `==` , `!=`
 - Όπως σε ένα λεξικό
 - `"a" < "b" -> True`
 - `"a" < "a1" -> True`
- Έλεγχος “εντός” (**in**) ή “εκτός” (**not in**)
 - `"new" in "new year" -> True`
 - `"a" in "abcd" -> True`
 - `"b" not in "aaaaa" -> True`

Σύνοψη Χρήσιμων

- Πως θα μετατρέψω ένα ακέραιο σε string;
- Πως θα μετατρέψω ένα string σε ακέραιο;
- Πως θα δω πόσους χαρακτήρες έχει ένα string;
- Μπορώ να προσθέσω (+) δύο string;
- Μπορώ να πολλαπλασιάσω δύο string;
- Μπορώ να ελέγξω αν μια πρόταση υπάρχει μέσα σε μία παράγραφο;

Σύνοψη Χρήσιμων

- `str(1820)`
- `int("300")`
- `len("This is a reallyyyy long string!")`
- `"Good" + "Morning"`
- Μπορώ να πολλαπλασιάσω δύο string; ΟΧΙ
 - αλλα ναι σε string με αριθμό: `3 * "!"`
- Ναι, με την `in`: `"hi!" in "hi! This is an auto message."`

Έτοιμος κώδικας

Ένας – ένας οι χαρακτήρες ενός string με for.

Έτοιμος κώδικας

Ένας – ένας οι χαρακτήρες ενός `string` με `for`.

```
for char in sentence :
```

Έτοιμος κώδικας

Τι πρέπει να είναι το sentence για να δουλέψει;

```
for char in sentence :
```

Έτοιμος κώδικας

Το **sentence** πρέπει να είναι **string**!

```
for char in sentence :
```

Αν το `sentence` ήταν...

- λίστα;
- αρχείο;

```
for char in sentence :
```

- τί θα ήταν το `char` ?

Αν το `sentence` ήταν...

- λίστα -> ένα **στοιχείο** της λίστας
- αρχείο -> μία **γραμμή** του αρχείου (`readline()`)

```
for char in sentence :
```

Έτοιμος κώδικας

Ένας – ένας οι χαρακτήρες ενός string με for.

```
def trimSpaces( sentence ):
    result = ""
    for char in sentence :
        if char != " " :
            result += char
    return result
```


Έτοιμος κώδικας

Ένας – ένας οι χαρακτήρες ενός string με for.

```
def count_vowels( word ):
    vowels = "ΑΕΙΟΥαειου"
    count = 0
    for letter in word :
        if letter in vowels:
            count += 1
    return count
```