

Δομές Δεδομένων - Συμβολοσειρές

PYTHON





Δομές Δεδομένων - 1

Οι δομές δεδομένων, γενικά, χωρίζονται σε δύο βασικές κατηγορίες τις **στατικές** και τις **δυναμικές**, ανάλογα με τη φάση της ανάπτυξης του προγράμματος κατά την οποία καθορίζεται το μέγεθος της δομής.

Το μέγεθος μιας στατικής δομής καθορίζεται κατά τη συγγραφή του προγράμματος από τον προγραμματιστή, ώστε να είναι γνωστό κατά τη φάση της μεταγλώττισης. Όταν το πρόγραμμα ξεκινήσει να εκτελείται δεσμεύεται χώρος στη μνήμη για τις στατικές δομές.

Από την άλλη οι **δυναμικές** δομές δεδομένων στηρίζονται σε μια λειτουργία που λέγεται **δυναμική εκχώρηση μνήμης**. Κατά τη δυναμική εκχώρηση μνήμης, το πρόγραμμα μπορεί να ζητήσει από το λειτουργικό σύστημα όση μνήμη απαιτείται για τη δημιουργία της δομής δεδομένων, κατά την εκτέλεση του προγράμματος.

Δομές Δεδομένων - 2

Οι δυναμικές δομές δεδομένων μπορούν να μεταβάλλουν το μέγεθός τους, προσθέτοντας ή αφαιρώντας αντικείμενα. Η πιο γνωστή δυναμική δομή δεδομένων είναι η λίστα, η οποία είναι και η βασική δομή δεδομένων της Python.

Ενδιαφέρον παρουσιάζει ο τύπος της συμβολοσειράς (str) ο οποίος επιτρέπει τη δυναμική δέσμευση μνήμης, αλλά όχι τη μεταβολή του μεγέθους της

Συμβολοσειρές

Η **συμβολοσειρά** είναι μια ακολουθία από χαρακτήρες και μπορεί να αποτελείται από περισσότερες από μία λέξεις, με τις λέξεις να μπορούν να είναι στην Ελληνική Γλώσσα, στην Αγγλική ή σε κάθε γλώσσα που υποστηρίζεται από το πρότυπο Unicode.

Μια συμβολοσειρά την ορίζουμε με εισαγωγικά αμφίπλευρα, μονά ή διπλά.

```
a = "Καλημέρα "
```

```
b = "ηλιόλουστη μέρα"
```

```
b = 'Have a nice day!'
```

Τι θα εμφανίσουν οι παρακάτω εντολές;

```
spam = "We want A SHRUBBERY!"
```

```
print spam
```

We want A SHRUBBERY!

Help! Help! I'm being repressed!

Μήνυμα λάθους.

```
spam = " I'm not quite dead yet!"
```

```
print spam
```

I'm not quite dead yet!

Μήνυμα λάθους

```
spam = 'I'm not quite dead yet!'
```

```
print spam
```

I'm not quite dead yet!

Μήνυμα λάθους

```
spam = ' blow my nose at you, so-called "Arthur King," '
```

```
print spam
```

```
I blow my nose at you, so-called "Arthur King,"
```

```
Μήνυμα λάθους
```

```
spam = "We are the knights who say 'Ni!'"
```

```
print spam
```

We are the knights who say 'Ni!'

Μήνυμα λάθους

```
spam = ' I'm not quite dead yet! '
```

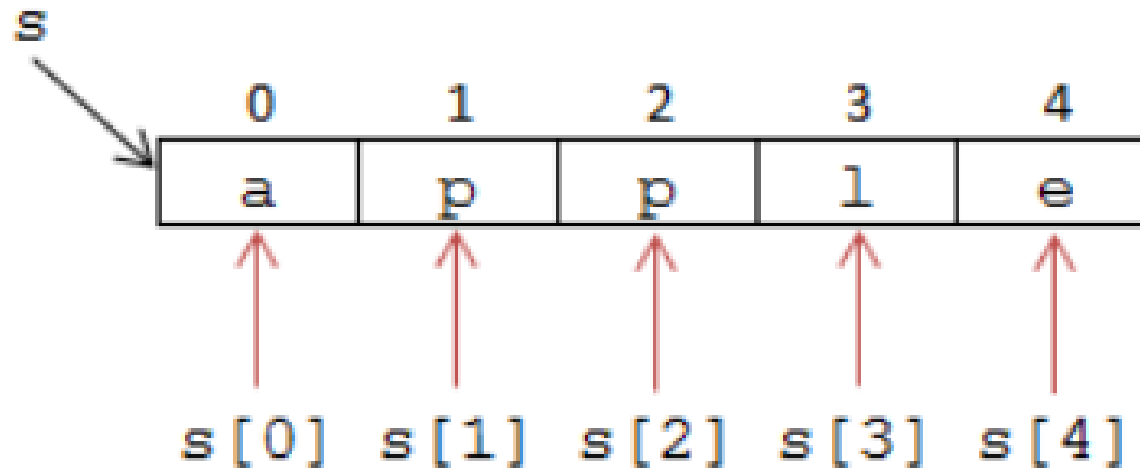
```
print spam
```

Μήνυμα λάθους

I'm not quite dead yet!

Συμβολοσειρές

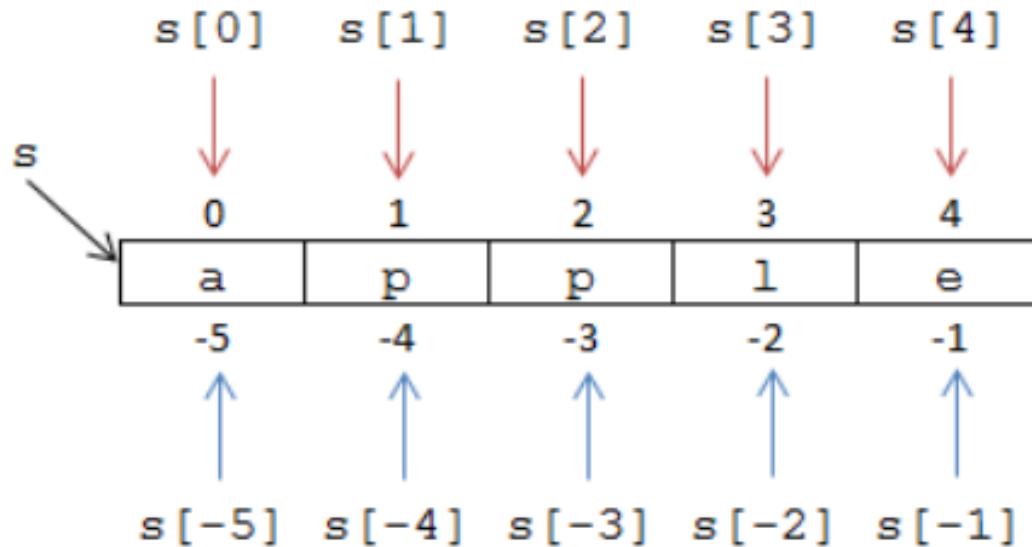
Η προσπέλαση σε κάθε χαρακτήρα της συμβολοσειράς γίνεται με τον αριθμό της θέσης που βρίσκεται, μέσα σε αγκύλες. Η αρίθμηση στην `python` ξεκινάει από το 0.



Συμβολοσειρές

Εναλλακτικά, μπορούμε να χρησιμοποιήσουμε και αρνητικούς δείκτες, οι οποίοι μετρώνε από το τέλος της συμβολοσειράς προς την αρχή.

Ο δείκτης -1 δίνει τον τελευταίο χαρακτήρα μιας συμβολοσειράς



```
word = "SPAM"
```

```
print word[1]
```

A

M

S

P

```
word = "SPAM"
```

```
print word[-1]
```

M

P

A

S

```
word = "SPAM"
```

```
print word[2]
```

P

M

A

S

```
word = "SPAM"
```

```
print word[10]
```

A

Μήνυμα λάθους

```
print "SPAM"[1]
```

P

M

A

S

```
print "SPAM"[-1]
```

M

S

P

A

```
print "SPAM"[2]
```

M

A

S

P

Συμβολοσειρές

Οι συμβολοσειρές είναι ακολουθιακές δομές δεδομένων και οποιαδήποτε από τις παρακάτω δύο μεθόδους θα εμφανίσει με τη σειρά όλους τους χαρακτήρες της συμβολοσειράς.

```
word = "ΠΛΗΡΟΦΟΡΙΚΗ"  
for ltr in word:  
    print ltr
```

```
word = "ΠΛΗΡΟΦΟΡΙΚΗ"  
for i in range(11):  
    print word[i]
```

```
word = "ΠΛΗΡΟΦΟΡΙΚΗ"  
for i in range(-11, 0, 1):  
    print word[i]
```

Π
Λ
Η
Ρ
Ο
Φ
Ο
Ρ
Ι
Κ
Η

Τι θα εμφανίσουν οι παρακάτω εντολές;

```
word = "SPAM"
```

```
for i in range(2):
```

```
    print word[i],
```

SPAM

A

SP

A

```
word = "SPAM"
```

```
for i in range(1,3):
```

```
    print word[i],
```

SPA

SP

PA

```
word = "SPAM"
```

```
for i in range(0,4,2):
```

```
    print word[i],
```

```
SPAM
```

```
SPM
```

```
SA
```

```
word = "SPAM"
```

```
for i in range(-1,-4,-1):
```

```
    print word[i],
```

```
SPAM
```

```
MAP
```

```
MAPS
```

```
word = "SPAM"
```

```
word = "STAR"
```

```
print word
```

Μήνυμα λάθους

STAR

SPAM

```
word = "SPAM"
```

```
word[1] = "T"
```

```
print word
```

SPAM

STAM

Μήνυμα λάθους

```
word = "SPAM"
```

```
word[0] = "E"
```

```
print word
```

Μήνυμα λάθους

EPAM

SPAM

Τελεστές Συμβολοσειράς

Ο τελεστής + όταν εφαρμόζεται σε αντικείμενα τύπου string, έχει σαν αποτέλεσμα τη συνένωσή τους σε μια συμβολοσειρά. (Πολυμορφισμός τελεστή +)

Ο τελεστής * όταν εφαρμόζεται σε αντικείμενα τύπου string με ακέραιο N έχει σαν αποτέλεσμα μια συμβολοσειρά που προκύπτει από N επαναλήψεις του string.

Οι υπαρξιακοί τελεστές in , not in οι οποίοι ελέγχουν, αν ένα αντικείμενο ανήκει σε ένα σύνολο αντικειμένων. Το αποτέλεσμα είναι True ή False.

Οι γνωστοί συγκριτικοί τελεστές (<, <=, >, >=, ==, !=) ισχύουν και στις συμβολοσειρές, η λειτουργία των οποίων βασίζεται στη λεξικογραφική διάταξη των χαρακτήρων, δηλαδή στη θέση που θα είχαν οι λέξεις σε ένα λεξικό.

Μικρότερες θεωρούνται οι λέξεις που βρίσκονται πιο μπροστά σε ένα λεξικό.

```
print "Monty" +'Python'
```

MontyPython

Μήνυμα λάθους

```
print "Monty" +'Python'+3
```

MontyPython3

Μήνυμα λάθους

```
print "Monty" +'Python'+str(3)
```

Μήνυμα λάθους

MontyPython3

```
print "1"+"2"
```

12

Μήνυμα λάθους

3

```
print 3 * "Spam"
```

Μήνυμα λάθους

3Spam

SpamSpamSpam

```
print ("Spam" +'Spam' )*2
```

Μήνυμα λάθους

SpamSpam

SpamSpamSpamSpam

```
print "Monty" +'Python'+ "Spam"*0
```

MontyPython

MontyPythonSpam

Μήνυμα λάθους

```
print "Spam" * "3"
```

Spam3

Μήνυμα λάθους

SpamSpamSpam

```
print "Monty" in 'Monty Python'
```

True

Μήνυμα λάθους

False

```
w= "Brian"
```

```
print w in "The life of Brian"
```

False

True

Μήνυμα λάθους

```
print "Monty" not in "Python"
```

MontyPython

Μήνυμα λάθους

True

False

```
letter = "m"
```

```
print letter in "1234567890"
```

True

Μήνυμα λάθους

False

```
letter = "m"
```

```
print letter in "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

False

True

```
letter = "Θ"
```

```
vowels = 'iIoOeEuU'
```

```
if letter in vowels:
```

```
    print 'Φωνήεν'
```

```
else:
```

```
    print "Σύμφωνο"
```

Σύμφωνο

Φωνήεν

```
print "Monty" == 'Monty'
```

Μήνυμα λάθους

True

False

```
w= "Brian"
```

```
print w != "Brian"
```

Μήνυμα λάθους

True

False

```
print "Monty" <= "Python"
```

False

True

Μήνυμα λάθους

```
num="10"
```

```
print num > "2"
```

False

True

Μήνυμα λάθους

```
print "Monty" <= "Monty Python"
```

Μήνυμα λάθους

True

False

```
print "Python" != "python"
```

False

True

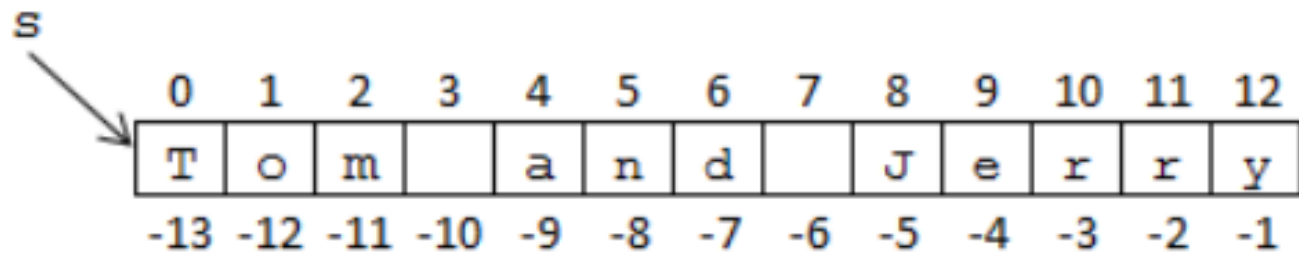
Διαμέριση Συμβολοσειράς (Slicing)

Ο τελεστής διαμέρισης (:) μπορεί να μας επιστρέψει ένα τμήμα μιας συμβολοσειράς. Η έκφραση `word[a : b]` μας επιστρέφει το τμήμα της συμβολοσειράς ή της λίστας από το στοιχείο `word[a]` μέχρι και το στοιχείο `word[b-1]`.

Μπορούμε να παραλείψουμε την αρχή ή το τέλος αν θέλουμε ένα τμήμα από την αρχή ή το τέλος της συμβολοσειράς αντίστοιχα. **`word[start : end]`**

word[start:end] # από start μέχρι end-1
word[start:] # από start μέχρι τέλος
word[:end] # από αρχή μέχρι end-1
word[:] # ολόκληρη η λέξη

word[start:end:step] # από start μέχρι end-1 με βήμα step
word[start::step] # από start μέχρι τέλος με βήμα step
word[:end:step] # από αρχή μέχρι end-1 με βήμα step
word[::step] # ολόκληρη η λέξη με βήμα step



```
>>> s = 'Tom and Jerry'
>>> s[0:3]
'Tom'
>>> s[4:7]
'and'
>>> s[8:13]
'Jerry'
>>> s[0:1]
'T'
>>>
```

```
>>> s[:3]
'Tom'
>>> s[4:]
'and Jerry'
>>>
```

```
>>> s[-13:-10]
'Tom'
>>>
```

```
word = "SPAM"
```

```
print word[0:2]
```

SP

A

A

SPAM

```
word = "SPAM"
```

```
print word[1:3]
```

SPA

PA

SP

```
word = "SPAM"
```

```
print word[0:4:2]
```

SPM

SA

SPAM

```
word = "SPAM"
```

```
print word[-1:-4:-1]
```

MAP

MAPS

SPAM

Μέθοδοι Συμβολοσειράς

Η συνάρτηση `len` μας επιστρέφει το μήκος της συμβολοσειράς δηλαδή πόσους χαρακτήρες περιέχει.

Ο τύπος `str` έχει και κάποιες εγγενείς μεθόδους, τις οποίες μπορούμε να καλέσουμε με τον συμβολισμό τελεία "." (dot notation). Για να δούμε όλες τις μεθόδους που υποστηρίζει ο τύπος `str`, δίνουμε στον interpreter: `dir(str)`.

Μέθοδοι Συμβολοσειράς

Παραδείγματα μεθόδων: `word.capitalize()`, `word.upper()`, `word.lower()`.

`word.upper()` Επιστρέφει ένα `string` μετατρέποντας τους πεζούς ενός `string` σε κεφαλαίους χαρακτήρες.

`word.lower()` Επιστρέφει ένα `string` μετατρέποντας τους κεφαλαίους χαρακτήρες ενός `string` σε πεζούς.

`word.capitalize()` Επιστρέφει ένα `string` μετατρέποντας το πρώτο μόνο χαρακτήρα ενός `string` σε κεφαλαίο χαρακτήρα.

Το αρχικό `string` `word` στις παραπάνω περιπτώσεις παραμένει αναλλοίωτο.

```
>>> s = 'hello'
>>> s.capitalize()
'Hello'
>>> s.lower()
'hello'
>>> s.upper()
'HELLO'
```

```
>>> word = "python"
```

```
>>> len(word)
```

```
6
```

```
>>> len("123")
```

```
3
```

```
word = "SPAM"
```

```
print len(word)
```

3

4

```
print len("SPAM")
```

4

3

```
word = "SPAM\n"
```

```
print len(word)
```

4

5

6

```
word = "SPAM"
```

```
print word.lower()
```

spam

SPAM

```
word = "Spam"  
print word.upper()
```

Spam

SPAM

```
word = "SPAM"  
print word.lower()  
print word
```

spam

SPAM

spam

spam

```
word = "the life of Brian"  
print word.capitalize()
```

The Life Of Brian

The life of Brian

The life of brian

```
word = "SPAM"  
word = word.lower()  
print word.lower()  
print word
```

spam

spam

SPAM

spam