Συμβολοσειρές / String

Κεφάλαιο 6

Python για Όλους

www.py4e.com

Τύπος Δεδομένων String

- String είναι μια ακολουθία χαρακτήρων
- Μια τιμή string χρησιμοποιεί εισαγωγικά 'Hello’ ή "Hello"
- Για strings το + σημαίνει «συνένωση»
- Όταν ένα string περιέχει αριθμούς, εξακολουθεί να είναι string
- Μπορούμε να μετατρέψουμε αριθμούς που περιέχονται σε ένα string σε αριθμούς χρησιμοποιώντας την int()

```
>>> str1 = "Hello"
>>> str2 = 'there'
>>> bob = str1 + str2
>>> print(bob)
Hellothere
>>> str3 = '123'
>>> str3 = str3 + 1
Traceback (most recent call
last):  File "<stdin>", line 1,
in <module>
TypeError: cannot concatenate
'str' and 'int' objects
>>> x = int(str3) + 1
>>> print(x)
124
>>>
```

Ανάγνωση και Μετατροπή

- Προτιμούμε να διαβάζουμε δεδομένα χρησιμοποιώντας συμβολοσειρές και στη συνέχεια να αναλύουμε και να μετατρέπουμε τα δεδομένα όπου χρειάζεται
- Αυτό μας επιτρέπει περισσότερο έλεγχο σε περιπτώσεις σφαλμάτων και/ή κακή εισαγωγή χρηστών
- Οι αριθμοί εισαγωγής πρέπει να μετατραπούν από συμβολοσειρές

```
>>> όνομα = input('Εισάγετε:')
Εισάγετε:Chuck
>>> print(όνομα)
Chuck
>>> apple = input('Εισάγετε:')
Enter:100
>>> x = apple - 10
Traceback (most recent call
last):  File "<stdin>", line 1,
in <module>
TypeError: unsupported operand
type(s) for -: 'str' and 'int'
>>> x = int(apple) - 10
>>> print(x)
90
```

Ψάχνοντας μέσα σε String

- Μπορούμε να πάρουμε οποιοδήποτε χαρακτήρα σε μια συμβολοσειρά χρησιμοποιώντας ένα δείκτη που καθορίζεται μέσα σε αγκύλες
- Η τιμή του δείκτη πρέπει να είναι ένας ακέραιος και ξεκινά από 0
- Η τιμή του δείκτη μπορεί να είναι μια έκφραση που θα υπολογιστεί

b	a	n	a	n	a
0	1	2	3	4	5

```
>>> φρούτο = 'banana'
>>> γράμμα = φρούτο[1]
>>> print(γράμμα)
a
>>> x = 3
>>> w = φρούτο[x - 1]
>>> print(w)
n
```

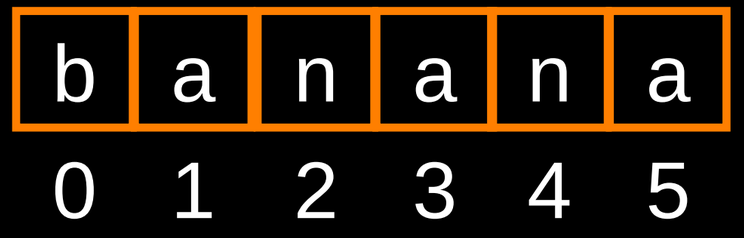
A Character Too Far

- Θα προκύψει λάθος στην python εάν προσπαθήσετε να χρησιμοποιήσετε τιμή δείκτη μεγαλύτερη από το μήκος της συμβολοσειράς μειωμένο κατά ένα
- Προσοχή λοιπόν κατά την κατασκευή τιμών δεικτών και τμημάτων

```
>>> zot = 'abc'
>>> print(zot[5])
Traceback (most recent call last):  File "<stdin>", line 1, in <module>
IndexError: string index out of range
>>>
```

Οι Συμβολοσειρές έχουν Μήκος (Length)

Η ενσωματωμένη συνάρτηση len επιστέφει το μήκος μιας συμβολοσειράς




```
>>> φρούτο = 'banana'
>>> print(len(φρούτο))
6
```

Η συνάρτηση len

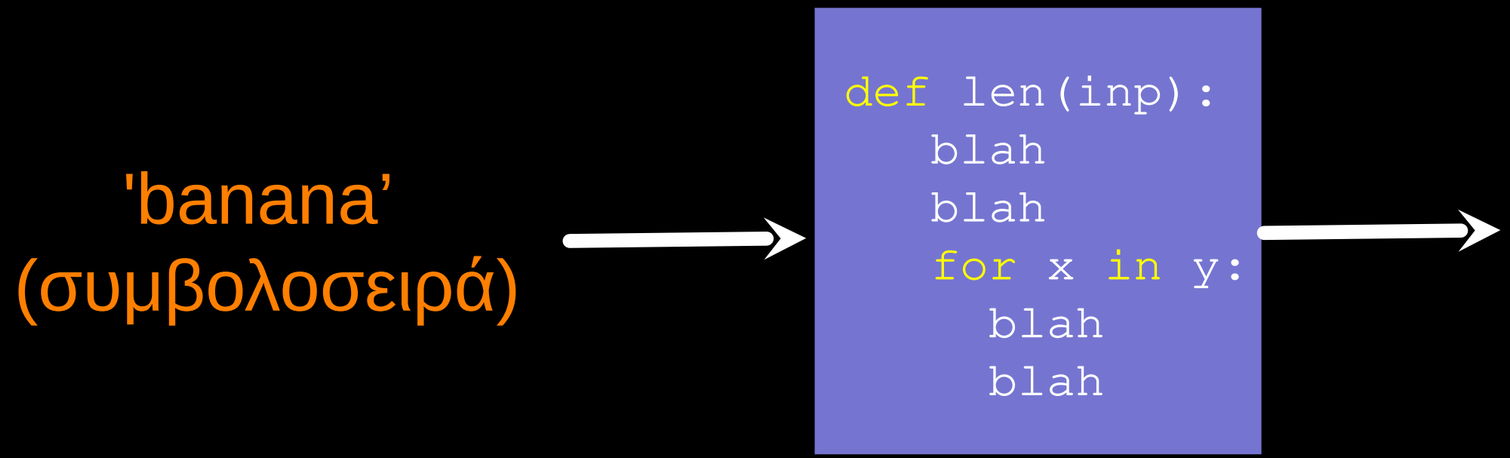
```
>>> φρούτο = 'banana'
>>> x = len(φρούτο)
>>> print(x)
6
```

Μια συνάρτηση είναι κάποιος αποθηκευμένος κώδικας που χρησιμοποιούμε. Μια συνάρτηση δέχεται κάποια είσοδο και παράγει κάποια έξοδο.

Η συνάρτηση len

```
>>> φρούτο = 'banana'
>>> x = len(φρούτο)
>>> print(x)
6
```

Μια συνάρτηση είναι κάποιος αποθηκευμένος κώδικας που χρησιμοποιούμε. Μια συνάρτηση δέχεται κάποια είσοδο και παράγει κάποια έξοδο.



Βρόχοι σε Συμβολοσειρές

Χρησιμοποιώντας μια εντολή while, μια μεταβλητή επανάληψης και τη συνάρτηση len, μπορούμε να κατασκευάσουμε ένα βρόχο που προσπελαύνει κάθε ένα από τα γράμματα της συμβολοσειράς ξεχωριστά

```
φρούτο = 'banana'
δείκτης = 0
while δείκτης < len(φρούτο):
    γράμμα = φρούτο[δείκτης]
    print(δείκτης, γράμμα)
    δείκτης = δείκτης + 1
```

```
0 b
1 a
2 n
3 a
4 n
5 a
```

Βρόχοι σε Συμβολοσειρές

- Ένας καθορισμένος βρόχος με χρήση της εντολής for είναι πολύ πιο κομψός
- Ο βρόχος for διαχειρίζεται πλήρως την μεταβλητή επανάληψης

```
φρούτο = 'banana'
for γράμμα in φρούτο:
    print(γράμμα)
```

```
b
a
n
a
n
a
```

Βρόχοι σε Συμβολοσειρές

- Ένας καθορισμένος βρόχος με χρήση της εντολής for είναι πολύ πιο κομψός
- Ο βρόχος for διαχειρίζεται πλήρως την μεταβλητή επανάληψης

```
φρούτο = 'banana'
for γράμμα in φρούτο :
    print(γράμμα)


δείκτης = 0
while δείκτης < len(φρούτο):
    γράμμα = φρούτο[δείκτης]
    print(γράμμα)
    δείκτης = δείκτης + 1
```

```
b
a
n
a
n
a
```

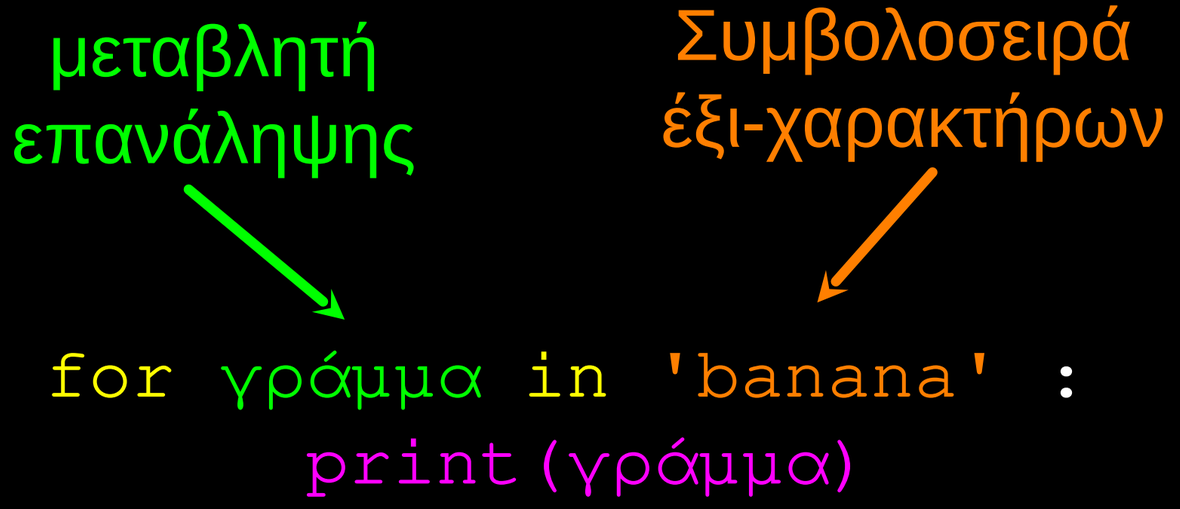
Βρόχος και Μέτρηση

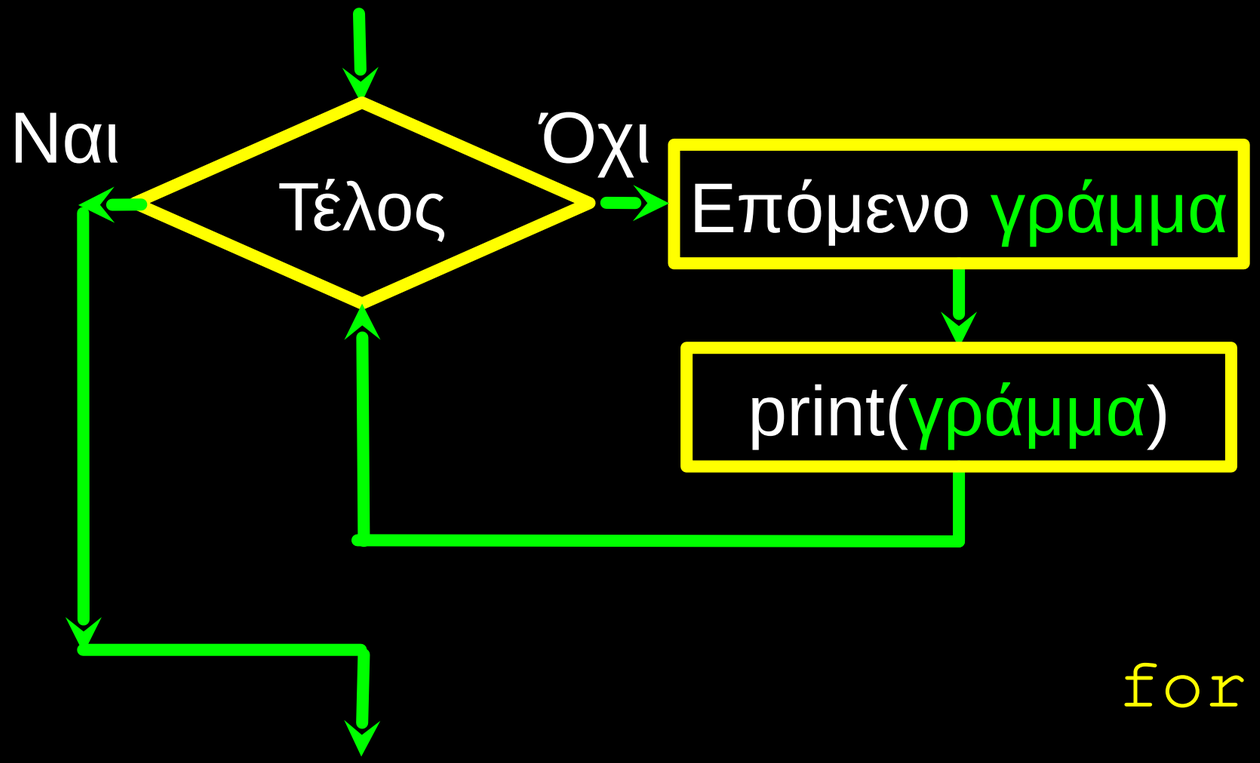
Αυτός είναι ένας απλός βρόχος που διατρέχει κάθε γράμμα μιας συμβολοσειράς και μετράει πόσες φορές ο βρόχος συναντά τον χαρακτήρα «a»

```
λέξη = 'banana’
πλήθος = 0
for γράμμα in λέξη :
    if γράμμα == 'a' :
        πλήθος = πλήθος + 1
print(πλήθος)
```

Μελετώντας Βαθύτερα την in

- Η μεταβλητή επανάληψης «διατρέχει» την ακολουθία (επιθυμητό σύνολο)
- Το μπλοκ (σώμα) του κώδικα εκτελείτε μία φορά για κάθε τιμή (εντός) της ακολουθίας
- Η μεταβλητή επανάληψης «διατρέχει» όλες τις τιμές μέσα στην ακολουθία





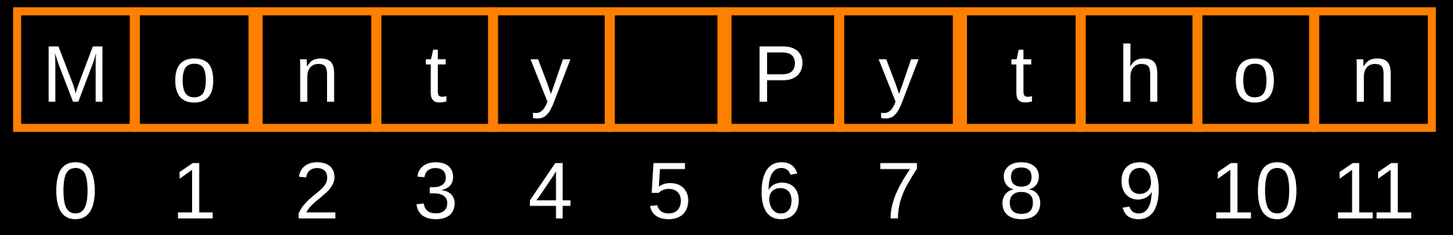
b	a	n	a	n	a

```
for γράμμα in 'banana' :
    print(γράμμα)
```

Η μεταβλητή επανάληψης διατρέχει» την συμβολοσειρά και το μπλοκ (σώμα) του κώδικα εκτελείτε μία φορά για κάθε τιμή (εντός) της ακολουθίας

Περισσότερες Λειτουργίες Συμβολοσειρών

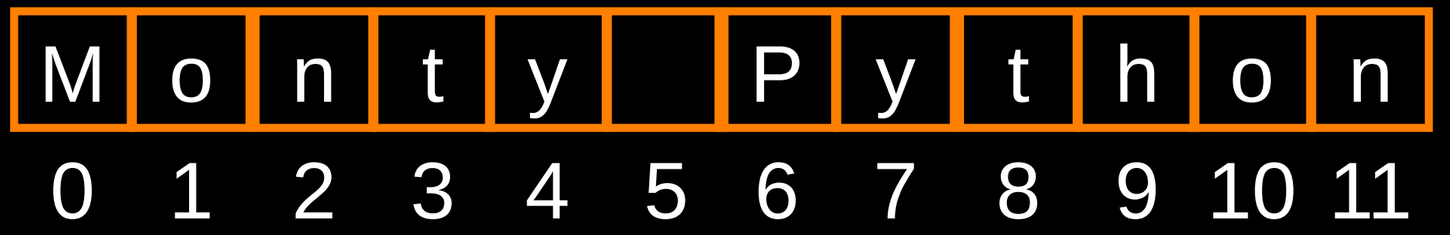
Τεμαχισμός Συμβολοσειράς



- Μπορούμε επίσης να εξετάσουμε οποιοδήποτε συνεχές τμήμα μιας συμβολοσειράς χρησιμοποιώντας τελεστή :
- Ο δεύτερος αριθμός είναι ένας πέρα από το τέλος του τμήματος - "έως το χωρίς να συμπεριλαμβάνεται»
- Εάν ο δεύτερος αριθμός είναι πέρα από το τέλος της συμβολοσειράς, σταματά στο τέλος

```
>>> s = 'Monty Python'
>>> print(s[0:4])
Mont
>>> print(s[6:7])
P
>>> print(s[6:20])
Python
```

Τεμαχισμός Συμβολοσειράς



Εάν παραλείψουμε τον πρώτο αριθμό ή τον τελευταίο αριθμό του τμήματος, εννοείται η αρχή ή το τέλος της συμβολοσειράς αντίστοιχα

```
>>> s = 'Monty Python'
>>> print(s[:2])
Mo
>>> print(s[8:])
thon
>>> print(s[:])
Monty Python
```

Συνένωση Συμβολοσειρών

Όταν εφαρμόζεται ο τελεστής + σε συμβολοσειρές, σημαίνει “συνένωση”

```
>>> a = 'Hello'
>>> b = a + 'There'
>>> print(b)
HelloThere
>>> c = a + ' ' + 'There'
>>> print(c)
Hello There
>>>
```

Χρησιμοποιώντας το in ως Λογικό Τελεστή

- Η λέξη-κλειδί in μπορεί επίσης να χρησιμοποιηθεί για να ελέγξει εάν μια συμβολοσειρά είναι «μέσα» μια άλλη συμβολοσειρά
- Η έκφραση in είναι μια λογική έκφραση που επιστρέφει True ή False και μπορεί να χρησιμοποιηθεί σε μια πρόταση if

```
>>> φρούτο = 'banana'
>>> 'n' in φρούτο
True
>>> 'm' in φρούτο
False
>>> 'nan' in φρούτο
True
>>> if 'a' in φρούτο :
...     print('Βρέθηκε!')
...
Βρέθηκε!
>>>
```

Σύγκριση Συμβολοσειρών

```
if λέξη == 'banana':
    print ('Όλο τα ίδια, bananas.')

if λέξη < 'banana':
    print('Η λέξη σου,' + λέξη + ', προηγείται της banana.')
elif λέξη > 'banana':
    print('Η λέξη σου,' + λέξη + ', έπεται της banana.')
else:
    print('Όλο τα ίδια, bananas.')
```

Βιβλιοθήκη String

- Η Python διαθέτει ένα πλήθος συναρτήσεων συμβολοσειρών οι οποίες περιέχονται στη βιβλιοθήκη string
- Αυτές οι συναρτήσεις είναι ενσωματωμένες σε κάθε string - τα επικαλούμαστε προσθέτοντας τη συνάρτηση στο τέλος της μεταβλητής συμβολοσειράς
- Αυτές οι συναρτήσεις τροποποιούν την αρχική συμβολοσειρά, αλλά επιστέφουν μια νέα, τροποποιημένη συμβολοσειρά

```
>>> greet = 'Hello Bob'
>>> zap = greet.lower()
>>> print(zap)
hello bob
>>> print(greet)
Hello Bob
>>> print('Hi There'.lower())
hi there
>>>
```

```
>>> stuff = 'Hello world'
>>> type(stuff)
<class 'str'>
>>> dir(stuff)
['capitalize', 'casefold', 'center', 'count', 'encode',
'endswith', 'expandtabs', 'find', 'format', 'format_map',
'index', 'isalnum', 'isalpha', 'isdecimal', 'isdigit',
'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace',
'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip',
'maketrans', 'partition', 'replace', 'rfind', 'rindex', 'rjust',
'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines',
'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper',
'zfill']
```

https://docs.python.org/3/library/stdtypes.html#string-methods

str.**replace** (*παλιό*, *νέο*[, *πλήθος*])
Επιστρέφει ένα αντίγραφο της συμβολοσειράς με όλες τις εμφανίσεις της συμβολοσειράς «παλαιό» να αντικαθίστανται από την «νέο». Εάν δίνεται ο προαιρετικός αριθμός πλήθος, αντικαθίστανται μόνο οι πρώτες «πλήθος» εμφανίσεις.

str.**rfind**(*sub[, αρχή[, τέλος]*)
Επιστρέφει τον μεγαλύτερο δείκτη στη συμβολοσειρά όπου βρίσκεται το sub string, έτσι ώστε το sub να περιέχεται στο s[αρχή: τέλος]. Τα προαιρετικά ορίσματα αρχή και τελος ερμηνεύονται όπως στ slice. Επιστρέφει -1 σε περίπτωση αποτυχίας.

str.**rindex**(*sub[, αρχή[, τέλος]]*)
Όπως η rfind() αλλά εγείρει ValueError όταν το τμήμα της συμβολοσειράς sub δεν βρεθεί

str.**rjust**(*εύρος[, χαρακτήρας_αναπλήρωσης]*)
Επιστρέψτε τη συμβολοσειρά ίδια με την αρχική συμβολοσειρά επιστρέφεται εάν το εύρος είναι μικρότερο ή ίσο με το len(s) ή μια νέα με δεξιά στοιχισμένη την αρχική συμβολοσειρά και συμπληρωμένη αριστερά με τον καθορισμένο χαρακτήρα συμπλήρωσης (η προεπιλογή είναι ένα κενό ASCII).

str.**rpartition**(*sep*)
Διαχωρίζει τη συμβολοσειρά στην τελευταία εμφάνιση του διαχωριστικού(sep) και επιστρέψτε μια τριάδα που περιέχει το τμήμα πριν από το διαχωριστικό, το ίδιο το διαχωριστικό και το τμήμα μετά το διαχωριστικό. Εάν το διαχωριστής δεν βρεθεί, επιστρέφει μια τριάδα που περιέχει δύο κενές συμβολοσειρές, ακολουθούμενη από την ίδια τη συμβολοσειρά.

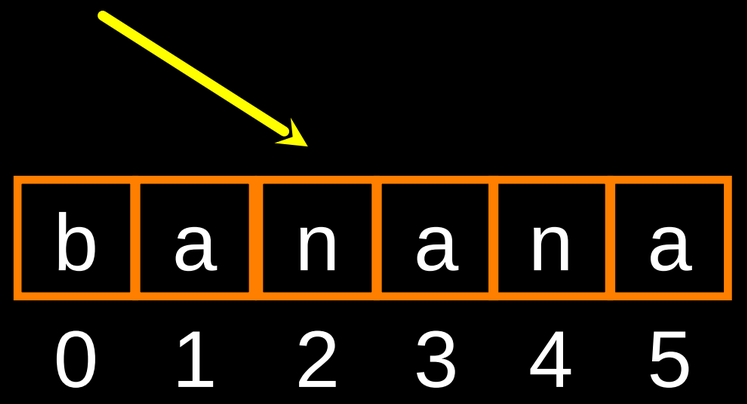
str.**rsplit**(*sep=None, maxsplit=-1*)
Επιστρέφει μια λίστα με τις λέξεις στη συμβολοσειρά, χρησιμοποιώντας το sep ως διαχωριστικό. Εάν δίνεται το maxsplit, γίνονται το πολύ maxsplit τμήματα ξεκινώντας από τα δεξιά. Εάν το sep δεν είναι καθορισμένο ή None, οποιαδήποτε συμβολοσειρά λευκού χώρου είναι διαχωριστικό. Εκτός από τη φορά του διαχωρισμού, από τα δεξιά, το rsplit() συμπεριφέρεται σαν split() το οποίο περιγράφεται λεπτομερώς παρακάτω.

Βιβλιοθήκη String

```
str.capitalize()
str.center(width[, fillchar])
str.endswith(suffix[, start[, end]])
str.find(sub[, start[, end]])
str.lstrip([chars])
```

```
str.replace(old, new[, count])
str.lower()
str.rstrip([chars])
str.strip([chars])
str.upper()
```

Αναζήτηση ενός String



- Χρησιμοποιούμε τη συνάρτηση find() για να αναζητήσουμε μια υποσυμβολοσειρά μέσα σε μια άλλη συμβολοσειρά

- find() βρίσκει την πρώτη εμφάνιση της υποσυμβολοσειράς

- Εάν η υποσυμβολοσειρά δεν βρεθεί, η find() επιστέφει -1

- Θυμηθείτε ότι η θέση συμβολοσειράς ξεκινά από το μηδέν

```
>>> φρούτο = 'banana'
>>> θέση = φρούτο.find('na')
>>> print(θέση)
2
>>> aa = φρούτο.find('z')
>>> print(aa)
-1
```

Μετατροπή όλων σε ΚΕΦΑΛΑΙΑ

- Μπορείτε να δημιουργήσετε ένα αντίγραφο μιας συμβολοσειράς σε πεζά ή κεφαλαία γράμματα
- Συχνά όταν ψάχνουμε για μια συμβολοσειρά χρησιμοποιώντας το find() μετατρέπουμε πρώτα τη συμβολοσειρά σε πεζά για να μπορέσουμε να αναζητήσουμε μια συμβολοσειρά ανεξαρτήτως πεζών - κεφαλαίων

```
>>> greet = 'Hello Bob'
>>> nnn = greet.upper()
>>> print(nnn)
HELLO BOB
>>> www = greet.lower()
>>> print(www)
hello bob
>>>
```

Αναζήτηση και Αντικατάσταση

- Η συνάρτηση replace() είναι σαν μια λειτουργία "αναζήτησης και αντικατάστασης" σε έναν επεξεργαστή κειμένου
- Αντικαθιστά όλες τις εμφανίσεις της αναζητούμενης συμβολοσειράς με τη συμβολοσειρά αντικατάστασης

```
>>> greet = 'Hello Bob'
>>> nstr = greet.replace('Bob','Jane')
>>> print(nstr)
Hello Jane
>>> nstr = greet.replace('o','X')
>>> print(nstr)
HellX BXb
>>>
```

Απαλοιφή Λευκών-Χαρακτήρων

- Μερικές φορές θέλουμε σε μια συμβολοσειρά να αφαιρέσουμε τους κενούς χαρακτήρες στην αρχή ή/και στο τέλος
- lstrip() και rstrip() αφαιρούν τους κενούς χαρακτήρες στα αριστερά ή δεξιά αντίστοιχα
- strip() αφαιρεί τους κενούς χαρακτήρες από την αρχή και το τέλος

```
>>> greet = '   Hello Bob  '
>>> greet.lstrip()
'Hello Bob  '
>>> greet.rstrip()
'   Hello Bob'
>>> greet.strip()
'Hello Bob'
>>>
```

Προθέματα

```
>>> γραμμή = 'Καλή σας μέρα'
>>> γραμμή.startswith('Καλή')
True
>>> γραμμή.startswith('ρ')
False
```

Ανάλυση και Εξαγωγή

21 31

```
From stephen.marquard@uct.ac.za Sat Jan  5 09:14:16 2008
```

```
>>> data = 'From stephen.marquard@uct.ac.za Sat Jan  5 09:14:16 2008'
>>> atpos = data.find('@')
>>> print(atpos)
21
>>> sppos = data.find(' ',atpos)
>>> print(sppos)
31
>>> host = data[atpos+1 : sppos]
>>> print(host)
uct.ac.za
```

Δύο Είδη Συμβολοσειρών

```
Python 2.7.10
>>> x = '이광춘'
>>> type(x)
<type 'str'>
>>> x = u'이광춘'
>>> type(x)
<type 'unicode'>
>>>
```

```
Python 3.5.1
>>> x = '이광춘'
>>> type(x)
<class 'str'>
>>> x = u'이광춘'
>>> type(x)
<class 'str'>
>>>
```

Στην Python 3, όλες οι συμβολοσειρές είναι Unicode

Σύνοψη

- Τύπος Συμβολοσειράς
- Ανάγνωση/Μετατροπή
- Δείκτης συμβολοσειράς []
- Τεμαχισμός συμβολοσειράς [2:4]
- Επαναλήψεις σε συμβολοσειρές με for και while
- Συνένωση συμβολοσειρών με +
- Λειτουργίες συμβολοσειράς
- Βιβλιοθήκη String
- Σύγκριση συμβολοσειρών
- Αναζήτηση σε συμβολοσειρά
- Αντικατάσταση κειμένου
- Απαλοιφή λευκών-χαρακτήρων

Ευχαριστίες / Συνεισφορές

Αρχική ανάπτυξη : Charles Severance, University of Michigan School of Information

Απόδοση στα Ελληνικά: Κιουρτίδου Δ. Κωνσταντία

… Εισαγάγετε νέους Μεταφραστές και άτομα που έχουν συνεισφέρει εδώ

Συνέχεια…