

rzfim - Python Sequence Types

List

רשימה היא אוסף של איברים, לא בהכרח מאותו סוג.

```
my_list = [12, "abc", 4.5]
```

Tuple

tuple הוא רשימה קבוצה. רשימה שלא ניתן לשנות את הערכים שלה.(immutable)

```
my_tuple = (1, "abc", 2.5)
```

String

.(Unicodes) String היא אוסף של תווים(.immutable). היא גם קבוצה(.).

```
my_string = "This is a string 123$%^"
```

תזכורת – חלק מפונקציות של מחרוזות שלמדנו

```
1 course = " Python Progammimg"  
2 print(course.upper())  
3 print(course.lower())  
4 print(course.title())  
5  
6 print(course.strip())  
7  
8 print(course.find("pro"))  
9 print(course.replace("P", "-"))
```

PYTHON PROGAMMING
python progammimg
Python Progammimg

Python Progammimg

-1
-ython -rogammimg

פונקציות נוספות וחוויות - ניתוח מחרוזת

`isalpha()`.מחרוזת או תו

מחזירה True אם התו הוא אות או אם המחרוזת מורכבת רק מאותיות.

`islower()`.מחרוזת או תו

`isupper()`.מחרוזת או תו

מחזירה True אם התו הוא אות קטנה/גדולה או אם המחרוזת מורכבת רק מאותיות קטנות/גדולות.

`isdigit()`.מחרוזת או תו

מחזירה True אם התו הוא מספר או אם המחרוזת מורכבת רק ממספרים.

Strings as sequences

SEQUENCES (String, List, Tuple)

- indexing: x[6]
- slicing: x[1:4]
- adding/concatenating: +
- multiplying: *
- checking membership: in/not in
- iterating for i in x:
- len(sequence1)
- min(sequence1)
- max(sequence1)
- sorted(list1)
- sequence1.count(item)
- sequence1.index(item)

- **indexing**

- Access any item in the sequence using its index

String

```
x = 'frog'  
print (x[3]) # prints 'g'
```

• slicing

- Slice out substrings, sublists, subtuples using indexes

[start : end+1 : step]

x = 'computer'

Code	Result	Explanation
x[1:4]	'omp'	Items 1 to 3
x[1:6:2]	'opt'	Items 1, 3, 5
x[3:]	'puter'	Items 3 to end
x[:5]	'compu'	Items 0 to 4
x[-1]	'r'	Last item
x[-3:]	'ter'	Last 3 items
x[:-2]	'comput'	All except last 2 items

קישור להעמקה בנושא slicing דין ב (stackoverflow

<https://stackoverflow.com/questions/509211/understanding-slice-notation>

- **adding / concatenating**

- Combine 2 sequences of the same type using +

String

```
x = 'horse' + 'shoe'  
print (x)                      # prints 'horseshoe'
```

- **multiplying**
 - Multiply a sequence using *

String

```
x = 'bug' * 3
print (x)          # prints 'bugbugbug'
```

- **checking membership**

- Test whether an item is **in** or **not in** a sequence

String

```
x = 'bug'  
print ('u' in x)                      # prints True  
print ('u' not in x)                   # prints False
```

• iterating

- Iterate through the items in a sequence

Item

```
some_string = "Hello"
for character in some_string:
    print(character * 2, end="")
                                # prints HHeelllloo
```

Index & Item

```
some_string = "HELLO"
for index, character in enumerate(some_string):
    print(index, character, end="")  # prints 0 H 1 E 2 L 3 L 4 O
```

- **number of items**

- Count the number of items in a sequence

String

```
x = 'bug'  
print (len(x)) # prints 3
```

- **minimum**

- Find the minimum item in a sequence lexicographically
- alpha or numeric types, but cannot mix types

String

```
x = 'HELLO'  
print (min(x)) # prints 'E'
```

- **maximum**

- Find the maximum item in a sequence
- alpha or numeric types, but cannot mix types

String

```
x = 'bug'  
print (max(x)) # prints 'u'
```

• sorting

- Returns a new list of items in **sorted** order
- Does not change the original list
- We did not see List yet. Will see it later.

String

```
x = 'bug'  
print (sorted(x))          # prints ['b', 'g', 'u']
```

- **count (item)**

- Returns count of an item

String

```
x = 'hippo'  
print (x.count('p')) # prints 2
```

- **index (item)**

- Returns the index of the first occurrence of an item

String

```
x = 'hippo'  
print (x.index('p')) # prints 2
```

אם הינו לא נמצא זורק
.ValueError
חריגת מסוג
עדיף להשתמש ב find

• unpacking

- Unpack the n items of a sequence into n variables

```
some_string = "Bye"  
a, b, c = some_string  
print(a, b, c)                      # now a is 'B'  
                                         # b is 'y',  
                                         # c is 'e'
```

Note:

The number of variables must exactly match the length of the list.

Hands On – Working with strings