Strings

a sequence of characters
String data type

>>> type("Hello")
<class 'str'>

>>> type('world')
<class 'str'>

Single or double quotes
Conversion to numbers and back

int('123')    ->    123
float('456.7') ->    456.7
str(123)      ->    '123'
str(456.7)    ->    '456.7'
Operations with strings

• Concatenation +

```python
>>> space = " 

>>> "Hello" + space + "world"
'Hello world'
```

• Multiplication *

```python
>>> ("Hello" + space) * 3
'Hello Hello Hello'
```
Index

<table>
<thead>
<tr>
<th>H</th>
<th>e</th>
<th>l</th>
<th>l</th>
<th>o</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

string[index]

string = "Hello"
letter = string[1]
print(letter)

print(string[5])

Traceback (most recent call last):
  print(string[5])
IndexError: string index out of range
Length

len(string)

word = "Hello"
length = len(word)
print(length)
Looping through strings

```python
word = 'Hello'
index = 0
while index < len(word):
    letter = word[index]
    print(index, letter)
    index = index + 1

word = 'Hello'
for letter in word:
    print(letter)
```
Slicing

```python
>>> string = "Hello world"
>>> string[3:8]
lo wo
```

```python
>>> string[4:]
o world
```

```python
>>> string[:9]
Hello wor
```

```python
>>> string[:]
Hello world
```

```python
>>> string[::-2]
Hello wor
```
Logical expressions with strings

```python
>>> "a" in "Hello"
False
>>> "lo" in "Hello"
True
>>> "hello" == "Hello"
False
>>> "hello" != "Hello"
True
>>> "world" < "Hello"
False
>>> "world" > "Hello"
True
```
<table>
<thead>
<tr>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>capitalize</td>
</tr>
<tr>
<td>casefold</td>
</tr>
<tr>
<td>center</td>
</tr>
<tr>
<td>count</td>
</tr>
<tr>
<td>encode</td>
</tr>
<tr>
<td>endswith</td>
</tr>
<tr>
<td>expandtabs</td>
</tr>
<tr>
<td>find</td>
</tr>
<tr>
<td>format</td>
</tr>
<tr>
<td>format_map</td>
</tr>
<tr>
<td>index</td>
</tr>
</tbody>
</table>

[https://docs.python.org/3/library/stdtypes.html#string-methods](https://docs.python.org/3/library/stdtypes.html#string-methods)
Project

• Before session 11 – idea (text) in Moodle
• During session 11 – discussion with instructor up to 10 points
• Before session 15 – program in Moodle
• During session 15 – demonstration to instructor up to 30 points
Project formulation

- The idea should be original - you should come up with it by yourself
- The topic (subject) of the project should be interesting to you
- The workload is ca 20h (trace the working hours on the project and submit it in a separate file together with the project)
- The solution (program) must contain:
  - at least two functions (def) written by you and you should invoke them in the program.
  - at least two conditional statements (if), at least one of them must contain else and/or elif part
  - at least one (nested) loop (for and/or while)
  - try-except block to make sure that the program does not crash
  - a list, a dictionary or a tuple.
  - simple graphical design

https://courses.cs.ut.ee/2018/nkp/fall/Main/Project
Test 5

Questions?
Homework (name and grades, username from url)

Questions?
Exercises

https://courses.cs.ut.ee/2018/nkp/fall/Main/During5