STRINGS
Retake of test

- Two possibilities: April 4 at 14:15 and April 4 at 16:15
- Room Liivi 2-206
- Format is the same as in the main test
- Best attempt counts
Project

• Register your team
• Submit project description

Schedule
• Form a team and select topic: **March 28**
• Alpha version: **April 25**
• Beta version: **May 23**
String data type

```python
>>> type("mg5#X4")
<class 'str'>
```

```python
>>> type('mg5#X4')
<class 'str'>
```

Single or double quotes
Conversion to numeric type and back

\[
\begin{align*}
\text{int('12')} & \rightarrow 12 \\
\text{float('34.5')} & \rightarrow 34.5 \\
\text{str(12)} & \rightarrow '12' \\
\text{str(34.5)} & \rightarrow '34.5'
\end{align*}
\]
Indices

characters → C e d a r
indices → 0 1 2 3 4

word = "Cedar"
letter = word[1] → e
print(letter)

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

```
cedar
```

```
word = "Cedar"
length = len(word)  # 5
print(length)
```
Looping through strings

- Loop over indices

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

- Loop over characters

```python
word = 'Flick'
for letter in word:
    print(letter)
```

Example:

<table>
<thead>
<tr>
<th>index</th>
<th>letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>F</td>
</tr>
<tr>
<td>1</td>
<td>l</td>
</tr>
<tr>
<td>2</td>
<td>i</td>
</tr>
<tr>
<td>3</td>
<td>c</td>
</tr>
<tr>
<td>4</td>
<td>k</td>
</tr>
</tbody>
</table>

Output:

```
0 F
1 l
2 i
3 c
4 k
```
Slicing

characters → C l e a r c a c h e
indices → 0 1 2 3 4 5 6 7 8 9 10

>>> text = 'Clear cache'

>>> text[3:8]
'ar ca'

>>> text[:9]
'Clear cac'

>>> text[:]
'Clear cache'

>>> text[4:]
'r cache'

>>> text[::]
'Clear cache'

>>> text[:::-2]
'Clear cac'
Operations with strings

• Concatenation +

    >>> space = " "
    >>> "Talk" + space + "show"
    'Talk show'

• Multiplication *

    >>> ("Talk" + space) * 3
    'Talk Talk Talk '
Logical expressions using strings

- **Substring**

  ```
  >>> "lo" in "Hello"
  True
  >>> "a" in "Hello"
  False
  ```

- **Equality**

  ```
  >>> "hello" != "Hello"
  True
  >>> "hello" == "Hello"
  False
  ```

- **Inequality**

  ```
  >>> "world" > "Hello"
  True
  >>> "world" < "Hello"
  False
  ```
Functions

capitalize  isalnum  join  rsplit
casefold  isalpha  ljust  rstrip
center  isdecimal  lower  split
count  isdigit  lstrip  splitlines
code  isidentifier  maketrans  startswith
codepoint  islower  partition  strip
endcode  isnumeric  replace  swapcase
endswith  isprintable  rfind  title
expandtabs  isspace  rindex  translate
find  istitle  rjust  upper
format  isupper  rpartition  zfill
format_map
index

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