07 – String methods

Bálint Aradi

Scientific Programming in Python (2026)

https://atticlectures.net/scipro/python-2026/

Some string methods

split(*separator*)

- Splits a string into pieces using a given delimiter

```
"a,b,c,d".split(",")
['a', 'b', 'c', 'd']
```

- If no delimiter is specified, the string is split by any whitespace characters (space, tab, newline)

```
"One short line.\nOne more.".split()
['One', 'short', 'line.', 'One',
'more.']
```

join(*iterator*)

- Joins the elements of the iterator into a string using the string as delimiter
- All elements returned by the iterator must be strings

```
", ".join(["word1", "word2", "word3"])
'word1, word2, word3'
```

Some string methods

lower(), upper()

- Converts all characters in a string to lower/upper case

```
"Word".lower()
'word'
words = ["Apfel", "Birne"]
[word.lower() for word in words]
['apfel', 'birne']
```

lstrip(), rstrip(), strip()

- Removes whitespace characters from left, right and both sides of a string

```
" word ".lstrip()
'word '
" word ".rstrip()
' word'
" word ".strip()
'word'
```

Some string methods

replace()

- Replaces all occurances of a substring with a given replacement

```
txt = "However, the sky was dark."
txt.replace("was", "is")
'However, the sky is dark.'
txt.replace(",", "")
'However the sky was dark.'
```

- The result of all string methods is always a new string (strings are immutable)
- If the result should be manipulated further by a string method, the methods can be “chained”

```
txt2 = txt.replace("was", "is")
txt_new = txt2.replace(",", "")
```

```
txt_new = txt.replace("was", "is").replace(",", "")
```

For further sting methods, see the Python Library Docs (String methods)

For non-trivial replacements regular expressions might be more suitable