

String Methods

Note: You can explore the [associated workbook](#) for this chapter in the cloud.

A *string* is a Python data type that is treated like text, even if it contains a number. Strings are always enclosed by either single quotation marks `'this is a string'` or double quotation marks `"this is a string"`. In this lesson, we're going to explore some special things that you can do with strings.

Practice with Strings

We're going to practice by using Franz Kafka's 1915 novella, *The Metamorphosis*.

```
sample_text = open("../texts/literature/Kafka-The-Metamorphosis.txt", encoding="utf-8").read()
```

```
print(sample_text)
```

☰ Contents

Print to PDF

[Extract Parts of Strings](#)

[Index](#)

[Slice](#)

[String Methods](#)

[Replace Words](#)

[Transform Strings to](#)

[Lowercase/Uppercase](#)

[Split Strings By a Delimiter](#)

[Join Strings By a Delimiter](#)

Extract Parts of Strings

Index

By using square brackets `[]`, you can grab or “index” part of a string based on its character number.

Here’s what happens if we index the first few characters of the *The Metamorphosis*. Remember that its opening line reads: “One morning, when Gregor Samsa woke from troubled dreams.”

Python Review

A Python index begins with `0`. So the `0`th character in a Python string is actually the `1`st character.

```
sample_text[0]
```

```
'O'
```

```
sample_text[1]
```

```
'n'
```

```
sample_text[2]
```

```
'e'
```

Slice

You can slice a string up between certain characters or up to certain characters.

```
string[start:stop:step]
```

By putting specific index numbers between these colons, you can slice the string at certain starting and stopping points, and you can also “step” by different amounts—that is, you can jump by a certain number through the string and take every nth item in the string (e.g. every 3rd item).

Let’s index our Kafka sample text from the beginning to the 121st character.

```
sample_text[:121]
```

```
'One morning, when Gregor Samsa woke from troubled dreams, he found\nhimself transformed in his bed into a horrible vermin.'
```

```
sample_text[0:121]
```

Let’s index our Kafka sample text from the 121st character to the 250th character.

```
sample_text[121:250]
```

```
' He lay on\nhis armour-like back, and if he lifted his head a little he could\nsee his brown belly, slightly domed and divided by '
```

Let's create a variable `first_line` and assign it the first sentence of *The Metamorphosis*.

```
first_line = sample_text[:121]
print(first_line)
```

One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin.

String Methods

String Method	Explanation
<code>string.lower()</code>	makes the string lowercase
<code>string.upper()</code>	makes the string uppercase
<code>string.title()</code>	makes the string titlecase
<code>string.strip()</code>	removes lead and trailing white spaces
<code>string.replace('old string', 'new string')</code>	replaces <code>old string</code> with <code>new string</code>
<code>string.split('delim')</code>	returns a list of substrings separated by the given delimiter
<code>string.join(list)</code>	opposite of <code>split()</code> , joins the elements in the given list together using the string
<code>string.startswith('some string')</code>	tests whether string begins with <code>some string</code>
<code>string.endswith('some string')</code>	tests whether string ends with <code>some string</code>
<code>string.isspace()</code>	tests whether string is a space

Replace Words

String Method	Explanation
<code>string.replace('old string', 'new string')</code>	replaces <code>old string</code> with <code>new string</code>

To replace a certain string within a string, you can use the `replace` method.

```
print(first_line.replace("morning", "evening"))
```

One evening, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin.

```
print(first_line.replace("vermin", "grilled cheese"))
```

```
One morning, when Gregor Samsa woke from troubled dreams, he found  
himself transformed in his bed into a horrible grilled cheese.
```

Your turn! Replace the word "vermin" with a word of your choosing:

```
first_line.replace("vermin", #your code here )
```

Transform Strings to Lowercase/Uppercase

String Method	Explanation
<code>string.lower()</code>	makes the string lowercase
<code>string.upper()</code>	makes the string uppercase
<code>string.title()</code>	makes the string titlecase

```
("I am really very quiet").lower()
```

```
("I am really very quiet").upper()
```

```
'I AM REALLY VERY QUIET'
```

Your turn! Transform the first line of Kafka's *The Metamorphosis* to upper case:

```
# your code here
```

Split Strings By a Delimiter

String Method	Explanation
<code>string.split('delim')</code>	returns a list of substrings separated by the given delimiter

With the `.split()` method, you can split up a strings into a a list of parts. By default, it splits on spaces, but you can put in a different delimiter and split on something else.

```
first_line.split()
```

```
[ 'One',
  'morning,',
  'when',
  'Gregor',
  'Samsa',
  'woke',
  'from',
  'troubled',
  'dreams,',
  'he',
  'found',
  'himself',
  'transformed',
  'in',
  'his',
  'bed',
  'into',
  'a',
  'horrible',
  'vermin. ']
```

Your turn! Split on the words “dreams” and see what happens.

```
# your code here
```

Join Strings By a Delimiter

String Method	Explanation
<code>string.join(list)</code>	opposite of <code>split()</code> , joins the elements in the given list together using the string

You can also put something back together again with the `join()` method!

```
kafka_split_words = first_line.split()
kafka_split_words
```

```
[ 'One',
  'morning,',
  'when',
  'Gregor',
  'Samsa',
  'woke',
  'from',
  'troubled',
  'dreams,',
  'he',
  'found',
  'himself',
  'transformed',
  'in',
  'his',
  'bed',
  'into',
  'a',
  'horrible',
  'vermin. ']
```

```
"SPACE".join(kafka_split_words)
```

```
'OneSPACEmorning, SPACEwhenSPACEGregorSPACESamsaSPACEwokeSPACEfromSPACEtroubledSPACEdr
eams,SPACEheSPACEfoundSPACEhimselfSPACEtransformedSPACEinSPACEhisSPACEbedSPACEintoSPA
CEaSPACEhorribleSPACEvermin. '
```

```
" ".join(kafka_split_words)
```

```
'One morning, when Gregor Samsa woke from troubled dreams, he found himself
transformed in his bed into a horrible vermin.'
```

Your turn! Join `kafka_split_words` with a delimiter of your choosing.

```
# your code here
```

By [Melanie Walsh](#)

© Copyright 2021.



This book is licensed under a [Creative Commons BY-NC-SA 4.0 License](#).