String Module

Table of contents

01

String Constants

Introduction

- The string module in Python provides a **collection** of **constants**, and **functions** that are useful for working with strings.

- It can be **used** in:
  - Text-processing
  - Pattern Matching
  - Cleaning up data

- It's part of Python's **standard library**, so no additional installations are required

Overview

The string module contains a **set of constants**:

- Ascii letters
- Digits
- Punctuations
- Printables
- Whitespaces

and **helper functions**:

- Template
- Formatter
- Capwords

To access it, use ***`import string`***

ascii_letters

A string containing all ASCII letters (a-z and A-Z).

Python	Output

`import string`
`print(string.ascii_letters)`
	`abcdefghijklmnopqr`
	`stuvwxyzABCDEFGHIJ`
	`KLMNOPQRSTUVWXYZ`

ascii_lowercase

A string containing all lowercase ASCII letters.

Python	Output
`import string` `print(string.ascii_lowercase)`	`abcdefghijklmnopq` `rstuvwxyz`

ascii_uppercase

A string containing all lowercase ASCII letters.

Python	Output
`import string` `print(string.ascii_uppercase)`	ABCDEFGHIJKLMNO PQRSTUVWXYZ

digits

A string containing all digit characters **(0-9).**

Python	Output
`import string` `print(string.digits)`	`0123456789`

hexdigits

A string containing all hexadecimal digit characters **(0-9 and a-f or A-F).**

Python	Output
`import string` `print(string.hexdigits)`	`0123456789abcdefABCDEF`

octdigits

A string containing all octal digit characters **(0-7).**

Python	Output
`import string` `print(string.octdigits)`	`01234567`

punctuation

A string containing all **punctuation** characters.

Python	Output
`import string` `print(string.punctuation)`	`!"#$%&'()*+,-` `./:;<=>?@[\]^_`{|}~`

whitespace

A string containing **all characters** that are considered **whitespace** (space, tab, newline, etc.).

Python	Output
`import string` `print(string.whitespace)`	`\t\n\r\x0b\x0c`

printable

A string containing **all characters** that are considered printable (includes digits, letters, punctuation, and whitespace).

Python	Output
`import string` `print(string.printable)`	`0123456789abcdefghijk` `lmnopqrstuvwxyzABCDEF` `GHIJKLMNOPQRSTUVWXYZ!` `"#$%&'()*+,-` `./:;<=>?@[\]^_`{|}~` `<spaces of all types>`

Summary Table

Collections	Content
ascii_letters	(a-z and A-Z)
ascii_lowercase	a-z
ascii_uppercase	A-Z
digits	(0-9)
hexdigits	(0-9 and a-f or A-F)
octdigits	(0-7)
punctuation	!"#$%&'()*+,./:;<=>?@[\]^_`{|}~-
whitespace	\t\n\r\x0b\x0c
printable	(a-z, A-Z, 0-9)

02

String Functions

String Functions

The string module contains **helper functions**:

- Template Class
- Formatter Class
- Capwords() Function

Template Class

The **string** module includes a **Template** class, which provides an easy and flexible way to perform substitutions in strings.

The **$** symbol is used as a placeholder, which can be replaced using the **substitute()** method.

substitute(): Replaces placeholders with the values provided. Raises a **KeyError** if a placeholder is missing.

safe_substitute(): Similar to **substitute()** but doesn't raise an error if a placeholder is missing; instead, it leaves the placeholder unchanged.

Template Class

Python

```
from string import Template

template = Template(“Course: $item\nPrice:$price")
result = template.substitute(item="Python Mastery
Series", price="₹ 0")
print(result)
```

Output

```
Course: Python Mastery Series
Price:₹ 0
```

Formatter()

The **Formatter** class in the string module is a low-level string formatting class that provides an advanced way to implement custom string formatting, beyond the usual **str.format()** method.

Python	Output

`from string import Formatter`
 	
`formatter = Formatter()`
 	
`print(formatter.format("Hi {}", "amar"))`
	`Hi amar`

capwords()

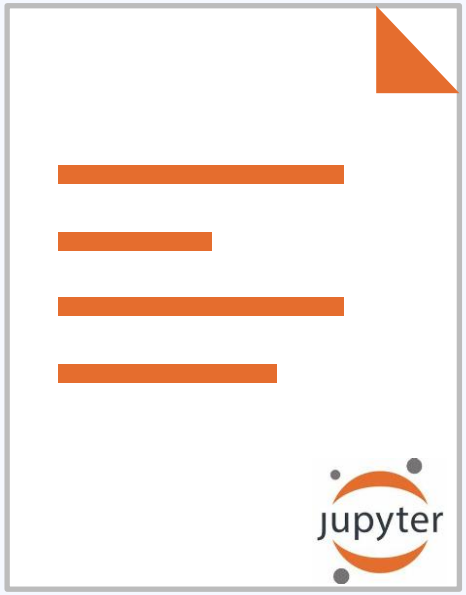
- Capitalizes the first letter of every word
- Lowers the rest of the letters
- Replaces multiple spaces between words with a single space.

Python	Output

`import string`
 	
`text = "    hello,   world   ! "`
`print(string.capwords(text))`
	`Hello, World !`

Use Cases

- Generating random strings (passwords, tokens).
- Filtering characters (e.g., removing punctuation).
- String templating for text replacements.
- Checking if a string contains only specific character sets.

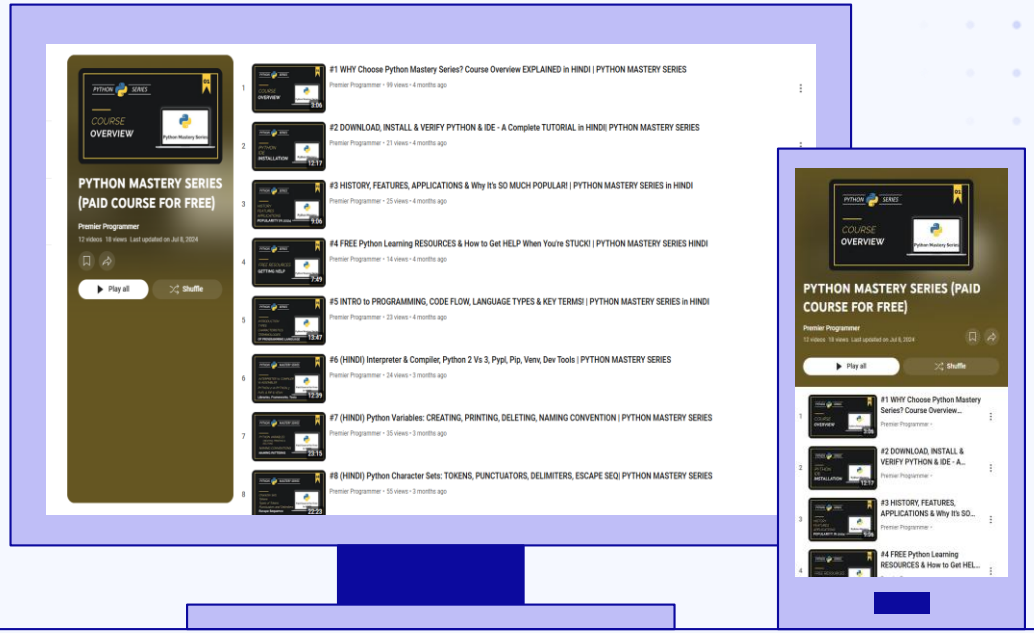


String Module

Download Link in
Description
and
Pinned Comment

WATCH

Level up your coding with each episode in this focused Python series.



Next Video!

Practice Set Solution