Python Backslash

If this Python Tutorial saves you hours of work, please whitelist it in your ad blocker 😭 and

Donate Now

(https://www.pythontutorial.net/donation/)

to help us ❤️ pay for the web hosting fee and CDN to keep the website running.

Summary: in this tutorial, you'll learn about the Python backslash character as a part of a special sequence character or to escape characters in a string.

Introduction to the Python backslash

In Python, the backslash(`\`) is a special character. If you use the backslash in front of another character, it changes the meaning of that character.

For example, the `t` is a literal character. But if you use the backslash character in front of the letter `t` , it'll become the tab character (`\t`).

Generally, the backslash has two main purposes.

First, the backslash character is a part of special character sequences such as the tab character `\t` or the new line character `\n` .

The following example prints a string that has a newline character:

```
print('Hello,\n World')
```

Output:

```
Hello,
World
```

The \n is a single character, not two. For example:

```
s = '\n'
print(len(s)) # 1
```

Second, the backslash (`\`) escape other special characters. For example, if you have a string that has a single quote inside a single-quoted string like the following string, you need to use the backslash to escape the single quote character:

```
s = '"Python\'s awesome" She said'
print(s)
```

Output:

```
"Python's awesome" She said
```

Backslash in f-strings

PEP-498 specifies that an f-string (https://www.pythontutorial.net/python-basics/python-f-strings/) cannot contain a backslash character as a part of the expression inside the curly braces `{}` .

The following example will result in an error:

```
colors = ['red','green','blue']
s = f'The RGB colors are:\n {'\n'.join(colors)}'
print(s)
```

Error:

```
SyntaxError: f-string expression part cannot include a backslash
```

To fix this, you need to join the strings in the `colors` list before placing them in the curly braces:

```
colors = ['red','green','blue']
rgb = '\n'.join(colors)
s = f"The RGB colors are:\n{rgb}"
print(s)
```

Output:

```
The RGB colors are:
red
green
blue
```

Backslash in raw strings

Raw strings (https://www.pythontutorial.net/python-basics/python-raw-strings/) treat the backslash character (`\`) as a literal character. The following example treats the backslash character `\` as a literal character, not a special character:

```
s = r'\n'
print(s)
```

Output:

```
\n
```

Summary

- Python backslash character (`\`) is a special character used as a part of special sequence such as `\t` and `\n` .
- Use the Python backslash (`\`) to escape other special characters in a string.
- F-strings cannot contains the backslash a part of expression inside the curly braces `{}` .
- Raw strings treat the backslash (\) as a literal character.