

1. What are escape characters, and how do you use them?

Answer: Escape characters represent characters in string values that would otherwise be difficult or impossible to type into code. In Python strings, the backslash `"\"` is a special character, also called the "escape" character. It is used in representing certain whitespace characters: `"\t"` is a tab, `"\n"` is a newline, and `"\r"` is a carriage return.

2. What do the escape characters `n` and `t` stand for?

Answer: `\n` is a newline; `\t` is a tab.

3. What is the way to include backslash characters in a string?

Answer: The `\` escape character will represent a backslash character.

4. The string `"Howl's Moving Castle"` is a correct value. Why isn't the single quote character in the word `Howl's` not escaped a problem?

Answer: The single quote in `Howl's` is fine because we have used double quotes to mark the beginning and end of the string.

5. How do you write a string of newlines if you don't want to use the `n` character?

Answer: Multiline strings allow you to use newlines in strings without the `\n` escape character.

6. What are the values of the given expressions?

`'Hello, world!'[1]`

`'Hello, world!'[0:5]`

'Hello, world!':5]

'Hello, world!':3:]

Answer:

```
In [1]: 1 'Hello, world!':[1]
```

```
Out[1]: 'e'
```

```
In [2]: 1 'Hello, world!':[0:5]
```

```
Out[2]: 'Hello'
```

```
In [3]: 1 'Hello, world!':5]
```

```
Out[3]: 'Hello'
```

```
In [4]: 1 'Hello, world!':3:]
```

```
Out[4]: 'lo, world!'
```

7. What are the values of the following expressions?

'Hello'.upper()

'Hello'.upper().isupper()

'Hello'.upper().lower()

Answer:

```
In [5]: 1 'Hello'.upper()
```

```
Out[5]: 'HELLO'
```

click to scroll output; double click to hide

```
In [6]: 1 'Hello'.upper().isupper()
```

```
Out[6]: True
```

```
In [7]: 1 'Hello'.upper().lower()
```

```
Out[7]: 'hello'
```

8. What are the values of the following expressions?

`'Remember, remember, the fifth of July.'.split()`

`'-'.join('There can only one.'.split())`

Answer:

```
In [1]: 1 'Remember, remember, the fifth of July.'.split()
```

```
Out[1]: ['Remember,', 'remember,', 'the', 'fifth', 'of', 'July.']
```

```
In [2]: 1 '-'.join('There can only one.'.split())
```

```
Out[2]: 'There-can-only-one.'
```

9. What are the methods for right-justifying, left-justifying, and centering a string?

Answer: The `rjust()`, `ljust()`, and `center()` string methods, respectively

10. What is the best way to remove whitespace characters from the start or end?

Answer: The `lstrip()` and `rstrip()` methods remove whitespace from the left and right ends of a string, respectively.