

# Python Basics Assignment 6

*1. What are Escape characters ? and how do you use them ?*

**Ans:** Escape characters represent characters in string values that would otherwise be difficult or impossible to type into code. we can use the backslash character to escape a single character or symbol

**example:** \t, \n

```
txt = "We are the so-called \"iNeuronites\" from India."
txt
```

'We are the so-called "iNeuronites" from India.'

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

**Ans:** \n is a newline, \t is a tab

```
txt = "Hello\nWorld!"
print(txt)
```

```
txt1 = "Hello\tWorld!"
print(txt)
```

```
Hello
World!
Hello
World!
```

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

**Ans:** The \\ escape character will represent the backslash character in a string

*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 ?*

**Ans:** The string "Howl's Moving Castle" escaped the problem because it is wrapped inside double quotes. if its wrapped inside single quotes then we have to use escape character \' to show single quote in the final output

'Howl\'s Moving Castle' -> 'Howl's Moving Castle'

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

**Ans:** Multiline string allow you to use newlines in string without the \n escape character

*# Example:*

```
a = """iNeuron full stack
Awesome Data Science Course"""
print(a)
```

*6.What are the values of the given expressions ?*

```
'Hello, world!'[1]  
'Hello, world!'[0:5]  
'Hello, world!':[5]  
'Hello, world!'[3:]
```

**Ans:** The values for the given expressions are:

```
'Hello, world!'[1] -> 'e'  
'Hello, world!'[0:5] -> 'Hello'  
'Hello, world!':[5] -> 'Hello'  
'Hello, world!'[3:] -> 'lo, world!'
```

*7.What are the values of the following expressions ?*

```
'Hello'.upper()  
'Hello'.upper().isupper()  
'Hello'.upper().lower()
```

**Ans:** The values for the given expressions are:

```
'Hello'.upper() -> 'HELLO'  
'Hello'.upper().isupper() -> True  
'Hello'.upper().lower() -> 'hello'
```

*8.What are the values of the following expressions ?*

```
'Remember, remember, the fifith of July.'.split()  
-''.join('There can only one'.split())
```

**Ans:** The values for the given expressions are:

```
'Remember, remember, the fifith of July.'.split() -> ['Remember,', 'remember,', 'the', 'fifith', 'of', 'July.']  
-''.join('There can only one'.split()) -> 'There-can-only-one'
```

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

**Ans:** The `rjust()`, `ljust()`, `center()` string methods, respectively

```
str.ljust(s, width[, fillchar])  
  
str.rjust(s, width[, fillchar])  
  
str.center(s, width[, fillchar])
```

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

**Ans:** The `lstrip()` and `rstrip()` methods remove whitesapce characters from the left and right ends of a string respectively

```
str = "---iNeuron---
```

*# using strip() to delete all '-'*

```
print ( " String after stripping all '-' is : ", end="")
```

```
print ( str.strip('-') )
```

```
# using lstrip() to delete all trailing '-'
```

```
print ( " String after stripping all leading '-' is : ", end="" )
```

```
print ( str.lstrip('-') )
```

```
# using rstrip() to delete all leading '-'
```

```
print ( " String after stripping all trailing '-' is : ", end="" )
```

```
print ( str.rstrip('-') )
```

String after stripping all '-' is : iNeuron

String after stripping all leading '-' is : iNeuron---

String after stripping all trailing '-' is : ---iNeuron