

Assignment - 6

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

```
In [1]: #An escape character is a backslash (\), followed by a character user want to insert.
```

```
In [2]: txt = "Hi "World"."      #ERROR

Cell In[2], line 1
      txt = "Hi "World"."      #ERROR
            ^
SyntaxError: invalid syntax
```

```
In [3]: txt = "Hi \"World\"."
#ERROR removed by the escape character "\".
```

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

\n for new line

\t for tab

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

```
In [4]: string = "\\Hello busy people" #By using Double backslash
string
```

```
Out[4]: '\\Hello busy people'
```

```
In [6]: string = r"\\Hello busy people" #By using r"\"
string
```

```
Out[6]: '\\Hello busy people'
```

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?

```
In [7]: string = "Howl's Moving Castle"
string # the single quote character in the word won't throw an error
```

```
Out[7]: "Howl's Moving Castle"
```

```
In [8]: string = 'Howl's Moving Castle'
string # Error # Single quote in single quotes throw error
```

```
Cell In[8], line 1
      string = 'Howl's Moving Castle'
                ^
SyntaxError: unterminated string literal (detected at line 1)
```

```
In [9]: string = 'Howl"s Moving Castle'
        string # Error # Double quote in single quotes does not throw error
```

```
Out[9]: 'Howl"s Moving Castle'
```

```
In [10]: string = "Howl"s Moving Castle"
         string # Error # Double quote in double quotes throw error
```

```
Cell In[10], line 1
      string = "Howl"s Moving Castle"
                  ^
```

```
SyntaxError: unterminated string literal (detected at line 1)
```

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

Using empty braces

```
In [11]: print("Assignment")
         print()           #EMPTY Braces
         print("6")
```

```
Assignment
```

```
6
```

6. What are the values of the given expressions?

'Hello, world!'[1]

'Hello, world!'[0:5]

'Hello, world!':5]

'Hello, world!'[3:]

INDEX in square boxes

```
In [12]: 'Hello, world!'[1]
```

```
Out[12]: 'e'
```

```
In [13]: 'Hello, world!'[0:5]
```

```
Out[13]: 'Hello'
```

```
In [14]: 'Hello, world!':5]
```

```
Out[14]: 'Hello'
```

```
In [15]: 'Hello, world!'[3:]
```

```
Out[15]: 'lo, world!'
```

7. What are the values of the following expressions?

'Hello'.upper()

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

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

```
In [16]: 'Hello'.upper()
```

```
Out[16]: 'HELLO'
```

```
In [17]: 'Hello'.upper().isupper()
```

```
Out[17]: True
```

```
In [18]: 'Hello'.upper().lower()
```

```
Out[18]: 'hello'
```

8. What are the values of the following expressions?

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

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

```
In [19]: 'Remember, remember, the fifth of July.'.split()
```

```
Out[19]: ['Remember,', 'remember,', 'the', 'fifth', 'of', 'July.']
```

```
In [20]: '-'.join('There can only one.'.split())
```

```
Out[20]: 'There-can-only-one.'
```

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

```
In [22]: string = "India is my country"

print ("The original string is : \n", string, "\n")

print ("The left aligned string is:")
print (string.ljust(40))

print ()
print ("The left aligned string with char :")
print (string.ljust(40, "#"))

print ()
print ("The right aligned string is:")
print (string.rjust(40))

print ()
print ("The right aligned string with char :")
print (string.rjust(40, "#"))

print ()
print ("The center aligned string is:")
print (string.center(40), "\n")

print ()
print ("The center aligned string with char:")
print (string.center(40, "#"))
```

The original string is :
 India is my country

The left aligned string is:
 India is my country

The left aligned string with char :
 India is my country#####

The right aligned string is:
 India is my country

The right aligned string with char :
 #####India is my country

The center aligned string is:
 India is my country

The center aligned string with char:
 #####India is my country#####

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

In [23]: *# .strip is the best way*

```
mystring = '      Hola      '

n = mystring.strip()

# before strip
print(mystring)
print(len(mystring))

# after strip
print(n)
print(len(n))
```

Hola

29
 Hola
 4