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

🡪 Escape characters has a special meaning when we use it inside the string. Usually backslash is referred to as an escape character. Backslash signifies that the next character after it has a different meaning. It could be printing double quotes or indicating a new line. We add backslash and a specific character after backslash to use escape characters.

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

🡪 The escape character n stands for new line and t stands for tab space. When we use \n in print function we get a new line and when we use \t we get a tab space.

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

🡪 Backslash is a special character used in strings also called as escape character. It is used in whitespace character like ‘\t’ or ‘\n’. By using the escape character in strings that is by prefixing backslash with a special 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?

🡪 “Howl’s Moving Castle” is a correct value as it is written between double quotes (“”). Thus, the single quote character in the word Howl’s is not escaped.

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

🡪 If we don’t want to use the n character and write a string of newlines we have two options-

* Using multiple print functions- Using multiple print functions each print function will print the output on a new line as the print function has an attribute ‘end’, which is set to ‘\n’ as default.

Example- print(“hello”)

print(“world”)

print(“hello”)

It will give output as-

hello

world

hello

* Using triple quotes in single print function- By using triple quotes in a single print function and by writing the string in new line as required in output, will print the put in newlines.

Example- print(‘’’hello

world

hello’’’)

It will give output as-

hello

world

hello

6. What are the values of the given expressions?

'Hello, world!'[1]

'Hello, world!'[0:5]

'Hello, world!'[:5]

'Hello, world!'[3:]

🡪 The values of 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()

🡪 The values of the following 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 fifth of July.'.split()

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

🡪 The values of the following expressions are-

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

Output is a list = [‘Remember,’ , ‘remember,’ , ‘the’ , ‘fifth’ , ‘of’ , ‘July.’]

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

Output is a string = ‘There-can-only-one.’

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

🡪 There are two methods for right-justifying, left-justifying, and centering a string-

* Aligning values within a specified length of text by using <,> or ^ symbols to specify left align, right align or centering respectively.
* Aligning using the inbuilt functions- ljust,rjust and center. ljust() justifies/aligns the string to left within the given width, similarly rjust() justifies/align the string to right and center() aligns the string in between the available width.

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

🡪 To remove whitespace characters from the start or end we use following methods-

* lstrip() - It removes the white space at the beginning of a string.
* rstrip() - It removes the white space at the end of a string.
* strip() - It removes the white space at the beginning and the end (both) of a string.