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

Ans. Escape characters are special characters that are used to represent certain non-printable or special characters within a string or text-based data. They are used to signal that the character following the escape character has a different meaning or should be interpreted differently.

In many programming languages and markup languages, escape characters are typically represented by a backslash () followed by another character. The backslash serves as an escape character itself, indicating that the following character is to be treated differently. Here are a few common examples:

**Newline character (\n):** It represents a line break or a new line. When encountered in a string, it causes the text to move to the next line. Example: "Hello\nWorld"

**Tab character (\t):** It represents a horizontal tab. When encountered in a string, it adds horizontal spacing equivalent to a tab stop. Example: "Hello\tWorld" outputs: "Hello World"

**Quotation mark ("):** It represents a double quotation mark. When used within a string delimited by double quotes, it allows you to include a double quotation mark without terminating the string. Example: "She said, \"Hello!\"" outputs: "She said, "Hello!""

**Backslash (\):** It represents a literal backslash character. When you need to include a backslash in a string, you can use two backslashes to escape it. Example: "C:\\Users\\Username" outputs: "C:\Users\Username"

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

**Ans. Newline character (\n):** It represents a line break or a new line. When encountered in a string, it causes the text to move to the next line. Example: "Hello\nWorld"

And **Tab character (\t):** It represents a horizontal tab. When encountered in a string, it adds horizontal spacing equivalent to a tab stop. Example: "Hello\tWorld" outputs: "Hello World"

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

Ans. Escape characters are typically represented by a backslash () followed by another character. The backslash serves as an escape character itself, indicating that the following character is to be treated differently.

**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" is correctly written without escaping the single quote character. The double quotes surrounding the entire string indicate the beginning and end of the string, and the single quote within the string is treated as a regular character.

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

Ans. if input is:- "Hello\nWorld"

Output:- Hello

World

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

**'Hello, world!'[1]**

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

**'Hello, world!'[:5]**

**'Hello, world!'[3:]**

Ans. 'Hello, world!'[1] evaluates to 'e'.

'Hello, world!'[0:5] evaluates to 'Hello'.

'Hello, world!'[:5] evaluates to 'Hello'.

'Hello, world!'[3:] Evaluates to ’lo, world!’

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

'Hello'.upper() =Evaluates to ‘HELLO’

'Hello'.upper().isupper() =evaluates to True

'Hello'.upper().lower() =evaluates to False

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

Ans. 'Remember, remember, the fifth of July.'.split():-This expression splits the string 'Remember, remember, the fifth of July.' into a list of substrings using whitespace as the delimiter. The split() method with no arguments splits the string at spaces, tabs, and newlines by default. The resulting list contains each word as a separate element.The value of this expression is the list ['Remember,', 'remember,', 'the', 'fifth', 'of', 'July.'].**'-'.join('There can only one.'.split()):-** evaluates to 'There-can-only-one.'.

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

Ans. **Right-justifying**: To right-justify a string, you can use the **rjust()** method. It pads the string with spaces on the left side to reach the specified width.

**Left-justifying**: To left-justify a string, you can use the **ljust() method**. It pads the string with spaces on the right side to reach the specified width.

**Centering**: To center a string, you can use the **center()** method. It adds equal spaces on both sides of the string to reach the specified width, ensuring that the string is centered.

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

Ans. **strip() method:** The strip() method removes leading and trailing whitespace characters from a string. It returns a new string with the whitespace removed. Example as below code

text = " Hello, World! "

stripped\_text = text.strip()

print(stripped\_text)

OUTPUT:- Hello , World!