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

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

3. What is the way to include backslash characters 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?

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

6. What are the values of the given expressions?

'Hello, world!'[1]

'Hello, world!'[0:5]

'Hello, world!'[:5]

'Hello, world!'[3:]

7. What are the values of the following expressions?

'Hello'.upper()

'Hello'.upper().isupper()

'Hello'.upper().lower()

8. What are the values of the following expressions?

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

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

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

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

**ANS:  
1. Escape characters are special characters used to represent characters that are difficult to type or would cause syntax errors if directly included in a string. They are preceded by a backslash (\) in Python strings. For example, \n represents a newline character.**

**2. In Python strings, the escape character \n represents a newline, while \t represents a tab.**

**3. To include a backslash character in a string, you can use a double backslash (\\). For example, "This is a backslash: \\".**

**4. The single quote character in the word "Howl's" is not escaped because the string is enclosed in double quotes (" "). In Python, you can use single quotes within a string enclosed in double quotes without needing to escape them.**

**5. To write a string of newlines without using the \n character, you can use triple quotes (''' ''') to create a multiline string. For example:**

**```python**

**print('''This is line 1**

**This is line 2**

**This is line 3''')**

**```**

**6. Values of the given expressions:**

**- 'Hello, world!'[1] => 'e'**

**- 'Hello, world!'[0:5] => 'Hello'**

**- 'Hello, world!'[:5] => 'Hello'**

**- 'Hello, world!'[3:] => 'lo, world!'**

**7. Values of the following expressions:**

**- 'Hello'.upper() => 'HELLO'**

**- 'Hello'.upper().isupper() => True**

**- 'Hello'.upper().lower() => 'hello'**

**8. Values of the following expressions:**

**- 'Remember, remember, the fifth of July.'.split() => ['Remember,', 'remember,', 'the', 'fifth', 'of', 'July.']**

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

**9. The methods for right-justifying, left-justifying, and centering a string are:**

**- Right-justifying: `rjust()`**

**- Left-justifying: `ljust()`**

**- Centering: `center()`**

**10. The best way to remove whitespace characters from the start or end of a string is to use the `strip()` method. It removes leading and trailing whitespace characters (spaces, tabs, newlines) from the string.**