01→

In python, escape characters are special in sequences of characters that are used to represent certain whitespace or other characters within a backslash (\). Here are some common escape characters and how they are used:

1. Newline (\n)
2. Tab (\t)
3. Backlash (\\)
4. Single Quote (\’)
5. Double Quote (\”)
6. Carriage Return (\r)
7. Backspace (\b)
8. Form Feed (\f)
9. Unicode Characters (\u or \U for 16-bit or 32-bit respectively)
10. Octal and Hexadecimal Notation

02→

\n stands for a newline character while \t stands for a tab character.

03→

To include backslash characters in a string, you need to escape each backslash with another backslash. This means we use \\ to represent a single backslash in the string.

04→
The string “Howl's Moving Castle” is a correct value and does not cause an issue with single quote in Howl's because the string is enclosed in double quotes (“). In python, we can use double quotes to define a string that contains single quotes without needing to escape them.

05→
To write a string with multiple newlines without using the \n escape character, we can use (‘ ’ ’ or “””). Triple quotes allow you to create a multiple string directly.

multi_line_string = “””This is a new code.
This is a new line.
This another line.”””
Print(multi_line_stirng)

#This will produce the same output.

06→

- ‘Hello, world!'[1] → this expression returns the character at index 1 of the string ‘Hello, world!’
- ‘Hello, world!’[0:5] → this will return the substring from the index 0 to 4.
- ‘Hello, world!’[: 5] → this will return the substring from the index 0 to 4.
- 'Hello, world!’[3:] → this will return the substring from the index 3 to the end.

07→

- ‘Hello’.upper() → “Hello”
- “Hello”.upper().is upper() → True
- “Hello”.upper().lower() → “Hello”

08→

What are the values of the following expressions?

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

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

- The value for the “Remember, remember, the fifth of July.”.split() is [“Remember,” , “remember,”, ‘the’, ‘fifth’, ‘of’, ‘July.’]
- ‘-’.join(‘There can only one.’.split()) → In this expression, first of all splits the string into a list of words, resulting in [‘There’, ‘can’, ‘only’, ‘one.’] and then joins these words into a single string with - as the separator. Which will result in ‘There-can-only-one.’.

09→

In Python, the methods for right-justifying, left-justifying, and centring a string are:

- Right-justifying a string:
  str.rjust(width[, fillchar])
  - This method returns a right-justified string of a given width, padded with the specified fillchar(default is a space).
  - Text = “Hello”
    print(text.rjust(10)) #output→” Hello”
    print(text.rjust(10,’-’)) #output→”Hello—-”
- Left-justifying a string:
  str.ljust(width[, fillchar])
  - print(text.ljust(10,’-’) → “-----Hello”
- Centering a string:

str.center(width[, fillchar])

- print(text.center(10, '-') → "--Hello—"

10→

To remove white space from the start and end of a string we to use string.strip() method. This will cause strip towards both of starting and ending side of a string.