Python basics assingments-6

1. Escape characters are characters that are used to represent certain characters or strings that cannot be typed directly. They are preceded by a backslash () and are used to create newlines, tabs, and other special characters.
2. The escape characters n and t stand for "newline" and "tab", respectively.
3. To include backslash characters in a string, you can use the escape character "\" to represent a single backslash.
4. The single quote character in the word "Howl's" is not a problem because it is surrounded by double quotes, which signify the beginning and end of the string. If the string had been written with single quotes, then the single quote in "Howl's" would have needed to be escaped with a backslash (e.g. 'Howl's Moving Castle').
5. To write a string of newlines without using the n character, you can use the escape character "\n" to represent a newline. Alternatively, you can use triple quotes (e.g. """Hello, world!""") to create a string that spans multiple lines.
6. 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!'

1. The values of the following expressions are:

* 'Hello'.upper() = 'HELLO'
* 'Hello'.upper().isupper() = True
* 'Hello'.upper().lower() = 'hello'

1. The values of the following expressions are:

* 'Remember, remember, the fifth of July.'.split() = ['Remember,', 'remember,', 'the', 'fifth', 'of', 'July.']
* '-'.join('There can only one.'.split()) = 'There-can-only-one.'

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

* right-justifying: str.rjust(width[, fillchar])
* left-justifying: str.ljust(width[, fillchar])
* centering: str.center(width[, fillchar])

1. The best way to remove whitespace characters from the start or end of a string is to use the string methods str.strip(), str.lstrip(), and str.rstrip(). These methods remove any leading or trailing whitespace characters, respectively