**Assignment\_6**

Q1. What are escape characters, and how do you use them?

Ans: print("it's Sunday")

print('it\'s Sunday')

output: it's Sunday

its's Sunday

Q2. What do the escape characters n and t stand for?

Ans: "\t" is a tab, "\n" is a newline

Q3. What is the way to include backslash characters in a string?

Ans: print('its a back slash \\ in a sentence')

Q4. 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 single quote in Howl's is fine because you've used double quotes to mark the beginning and end of the string.

like wise we can use double quotes in with single quotes.

print("HowI's Moving Castle")

output: HowI's Moving Castle

Q5. How do you write a string of newlines if you don't want to use the n character?

Ans: we can use the parameter end ='\n' in print function.

print("printing something in multiple lines",end = '\n')

print("newline")

output: printing something in multiple lines

newline

Q6. 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] 🡪 ‘e’

'Hello, world!'[0:5] 🡪 ‘Hello’

'Hello, world!'[:5] 🡪 ‘Hello’

'Hello, world!'[3:] 🡪 ‘lo’,’world’

Q7. What are the values of the following expressions?

'Hello'.upper()

'Hello'.upper().isupper()

'Hello'.upper().lower()

Ans:

'Hello'.upper() 🡪 ‘HELLO’

'Hello'.upper().isupper() 🡪 ‘True’

'Hello'.upper().lower() 🡪 ‘hello’

Q8. What are the values of the following expressions?

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

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

Ans: 'Remember, remember, the fifth of July.'.split() -> splits with space and returns list of words

'-'.join('There can only one.'.split()) splits with space and join with '\_'

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

Output: ['Remember,', 'remember,', 'the', 'fifth', 'of', 'July.']

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

Output: 'There-can-only-one.'

Q9. What are the methods for right-justifying, left-justifying, and centering a string?

Ans: The following methods are used for justifying strings

ljust()

rjust()

center()

str ="Hello"

str.ljust(10,"-")

output: 'Hello-----'

str ="How are you"

str.rjust(40,'\*')

output: '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*How are you'

str ="How are you"

str.center(40,'\*')

output: '\*\*\*\*\*\*\*\*\*\*\*\*\*\*How are you\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

Q10. What is the best way to remove whitespace characters from the start or end?

Ans: lstrip() -> removes white spaces from left of the string

rstrip() -> removes whitespaces from right of the string

str =" hkjhjdfhjd jkhjhd "

str.lstrip()

output: ‘hkjhjdfhjd jkhjhd '

str =" hkjhjdfhjd jkhjhd "

str.rstrip()

output: ' hkjhjdfhjd jkhjhd'