

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

- 1 Escape characters or sequences are illegal characters for Python and never get printed as part of the output. When backslash is used in Python programming, it allows the program to escape the next characters.
- 2
- 3 Escape characters can be classified as non-printable characters when backslash precedes them. The print statements do not print escape characters.

```
In [12]: 1 print("my \nname \t is")
          2
```

```
my
name      is
```

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

- 1 "\t" is a tab, "\n" is a newline

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

```
In [13]: 1 print('its a back slash \ in a line')
```

```
its a back slash \ in a line
```

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?

- 1 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

```
In [14]: 1 print("HowI's Moving Castle")
```

```
HowI's Moving Castle
```

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

```
In [15]: 1 print("printing something in new line",end = '\n')
          2 print("newline")
```

```
printing something in new line
newline
```

6. What are the values of the given expressions? 'Hello, world!'[1] 'Hello, world!'[0:5] 'Hello, world!':5] 'Hello, world!':[3:]

```
In [16]: 1 'Hello, world!'[1]
```

```
Out[16]: 'e'
```

```
In [17]: 1 'Hello, world!'[0:5]
```

```
Out[17]: 'Hello'
```

```
In [18]: 1 'Hello, world!':5]
```

```
Out[18]: 'Hello'
```

```
In [19]: 1 'Hello, world!':[3:]
```

```
Out[19]: 'lo, world!'
```

7. What are the values of the following expressions?

```
'Hello'.upper() 'Hello'.upper().isupper() 'Hello'.upper().lower()
```

```
In [20]: 1 'Hello'.upper()
```

```
Out[20]: 'HELLO'
```

```
In [21]: 1 'Hello'.upper().isupper()
```

```
Out[21]: True
```

```
In [22]: 1 'Hello'.upper().lower()
```

```
Out[22]: 'hello'
```

8. What are the values of the following expressions?

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

```
In [23]: 1 'Remember, remember, the fifth of July.'.split()
```

```
Out[23]: ['Remember,', 'remember,', 'the', 'fifth', 'of', 'July.']
```

```
In [24]: 1 '-'.join('There can only one.'.split())
```

```
Out[24]: 'There-can-only-one.'
```

```
In [26]: 1 a = "hello how are you".split()
```

```
In [27]: 1 '-'.join(a)
```

```
Out[27]: 'hello-how-are-you'
```

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

```
In [28]: 1 # Left justified
2 st = "Shiva"
3 st.ljust(10, '-')
```

```
Out[28]: 'Shiva-----'
```

```
In [29]: 1 # Right justified
2 st = "Shiva"
3 st.rjust(10, '-')
```

```
Out[29]: '-----Shiva'
```

```
In [32]: 1 # centre justified
2 st = "Shiva"
3 st.center(11, '*')
```

```
Out[32]: '***Shiva***'
```

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

```
In [34]: 1 st = " gfgfgfgfgfgfgf yuyuyuuyuy "
2 st.lstrip()
```

```
Out[34]: 'gfgfgfgfgfgfgf yuyuyuuyuy '
```

```
In [35]: 1 st = " gfgfgfgfgfgfgf yuyuyuuyuy "
2 st.rstrip()
```

```
Out[35]: ' gfgfgfgfgfgfgf yuyuyuuyuy'
```

In []:	1	
---------	---	--