

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

An escape character is a backslash \ followed by the character you want to insert. for example \t , \n ,\r etc.Escape characters are characters that are generally used to perform certain tasks and their usage in code directs the compiler to take a suitable action mapped to that character.

In [5]:

```
1 print('mynameis\nlakhan')
```

mynameis
lakhan

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

\n is a newline; \t is a tab.

In [6]:

```
1 print('ram\tshyam')
```

ram shyam

In [7]:

```
1 print('mynameis\nlakhan')
```

mynameis
lakhan

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

In [22]:

```
1 print(r'flow wtih\time')
```

flow wtih\time

In [23]:

```
1 print('flow with\\time')
```

flow with\time

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?

The single quote in Howl's is fine because we have used double quotes to mark the beginning and end of the string.

In [24]:

```
1 "Howl's Moving Castle"
```

Out[24]: "Howl's Moving Castle"

In [25]:

```
1 'Howl's Moving Castle'
```

Input In [25]
 'Howl's Moving Castle'
 ^
SyntaxError: invalid syntax

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

In [36]:

```
1 s="""hey,buddy
2     your are a
3     captain"""
4 print(s)
```

hey,buddy
 your are a
 captain

In [35]:

```
1
```

Out[35]: 'hey,buddy\n your are a \n captain'

6. What are the values of the given expressions?

'Hello, world!'[1] 'Hello, world!'[0:5] 'Hello, world!':[5] 'Hello, world!':[3:]

```
In [37]: 1 'Hello, world!'[1]
Out[37]: 'e'

In [38]: 1 'Hello, world!'[0:5]
Out[38]: 'Hello'

In [39]: 1 'Hello, world!':[5]
Out[39]: 'Hello'

In [40]: 1 'Hello, world!'[3:]
Out[40]: 'lo, world!'
```

7. What are the values of the following expressions?

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

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

In [41]: 1 'Hello'.upper()
Out[41]: 'HELLO'

In [ ]: 1 'Hello'.upper().isupper()
2 '
Out[ ]: True

In [42]: 1 'Hello'.upper().isupper()
Out[42]: True

In [43]: 1 'Hello'.upper().lower()
Out[43]: 'hello'
```

8. What are the values of the following expressions?

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

```
In [47]: 1 'Remember, remember, the fifth of July.'.split()
Out[47]: ['Remember,', 'remember,', 'the', 'fifth', 'of', 'July.']

In [46]: 1 '-'.join('There can only one.'.split() )
Out[46]: 'There-can-only-one.'
```

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

```
In [86]: 1 s=' flow with time '
2 s
Out[86]: ' flow with time '

In [65]: 1 s.rjust(30,'$')
Out[65]: '$$$$$$$$$$$$$$$$$flow with time'

In [67]: 1 s.ljust(40,'@')
Out[67]: 'flow with time@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@'

In [77]: 1 s.center(80,'@')
Out[77]: '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@flow with time@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@'
```

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

The lstrip(),replace() and rstrip() methods remove whitespace from the left and right ends of a string, respectively.

In [87]:

1

s

Out[87]:

' flow with time '

In [89]:

1

s.lstrip(' ')

Out[89]:

'flow with time '

In [90]:

1

s.rstrip(' ')

Out[90]:

' flow with time'

In [91]:

1

s.replace(' ','')

Out[91]:

'flowwithtime'

In []:

1