

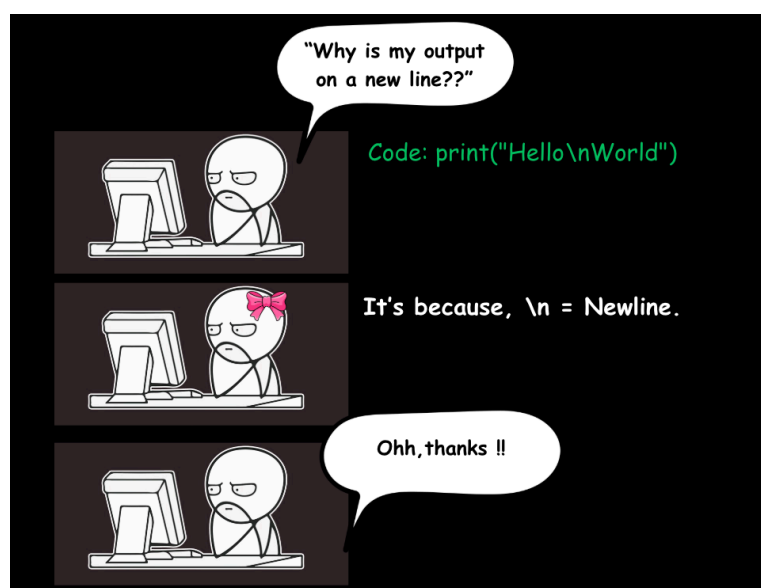
Python Escape Characters -

Practice Programs (Beginner):

1. Print a string with a new line between words
2. Use tab space between two words
3. Use double quotes inside a string
4. Use single quotes inside a string
5. Print a backslash
6. Print a multi-line string using `\n`
7. Combine tab and newline in a string
8. Use carriage return
9. Raw strings to ignore escape characters
10. Print the file-path, without using Raw strings

Solutions:

1. Print a string with a new line between words



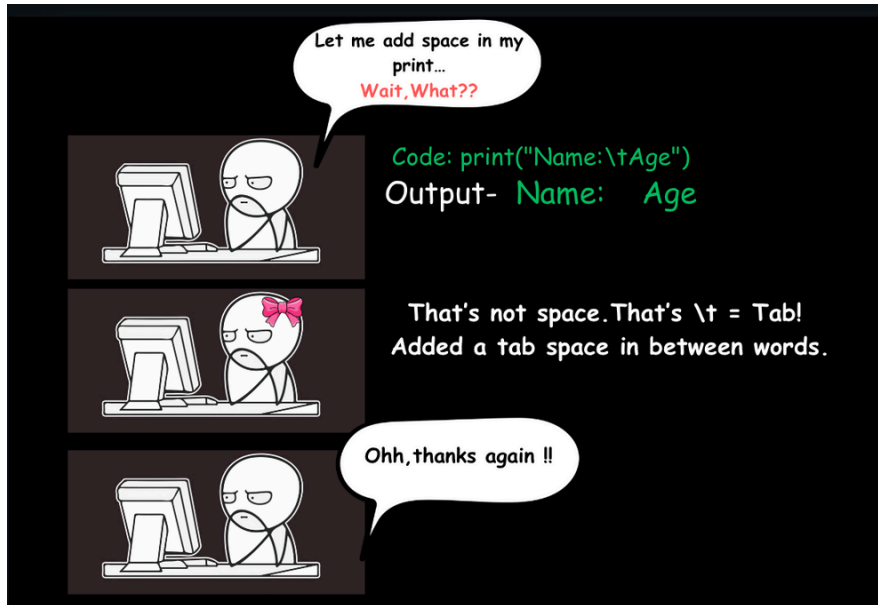
Code:

```
print("Hello\nWorld")
```

Output :

```
Hello  
World
```

2. Use tab space between two words



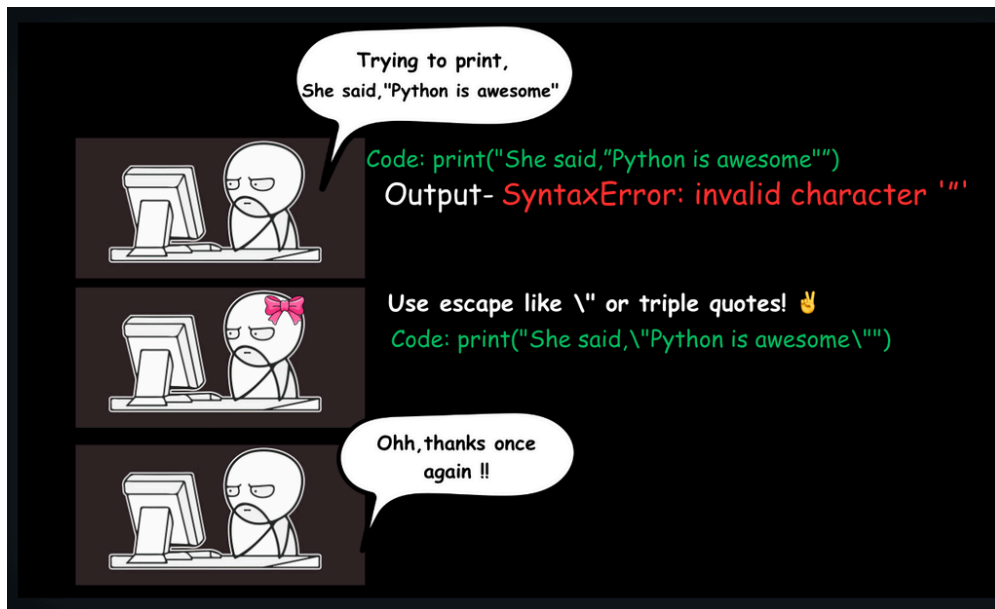
Code:

```
print("Python\tRocks")
```

Output :

```
Python Rocks
```

3. Use double quotes inside a string



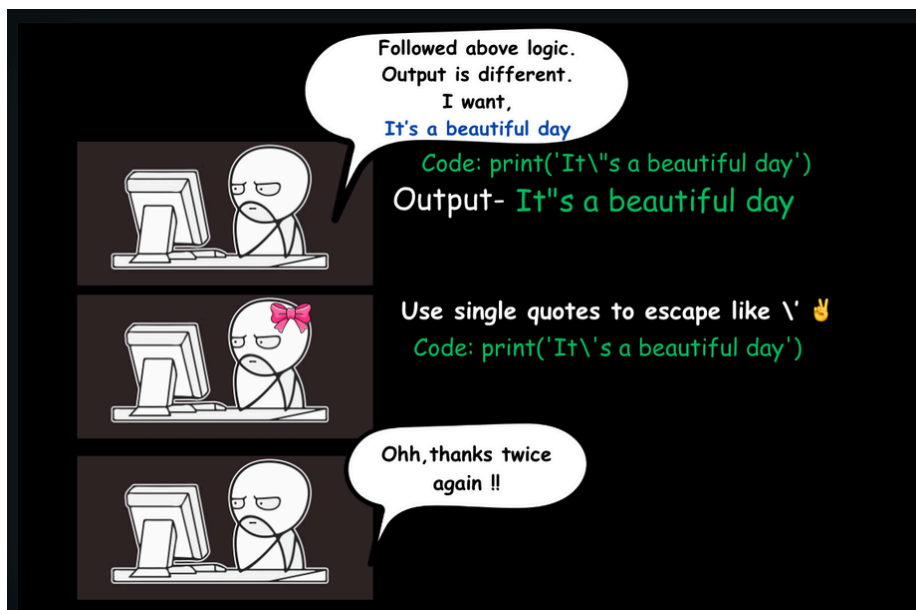
Code :

```
print("She said, \"Python is awesome!\")
```

Output:

She said, "Python is awesome!"

4. Use single quotes inside a string



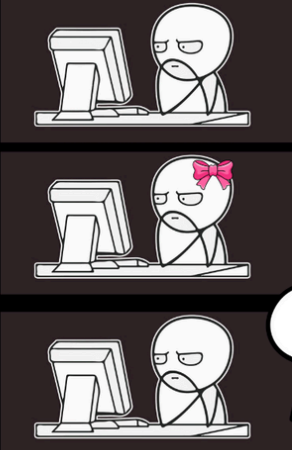
Code :

```
print('It\'s a beautiful day!')
```

Output:

It's a beautiful day!

5. Print a backslash



I wrote this to print a folder path:

```
Code: print("C:\newfolder\code")
```

Output- C:
ewfolder\code

Because \n means newline! Use double backslashes like this:

```
Code: print("C:\\newfolder\\code")
```

Ohh, gotcha !!

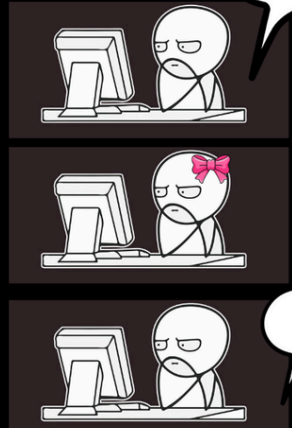
Code :

```
print("This is a backslash: \\")
```

Output:

This is a backslash: \

6. Print a multi-line string using \n



Can i create multiline string using '\n'

Yes ,you can . \n adds newline. Like this.

```
Code: print("Line1\nLine2\nLine3")
```

Output: Line1
Line2
Line3

Cool!

Code :

```
print("Line1\nLine2\nLine3")
```

Output:

```
Line1
Line2
Line3
```

7. Combine tab and newline in a string

Code :

```
print("Name:\tJohn\nAge:\t25")
```

Output:

```
Name:  John
Age:   25
```

8. Use carriage return



Code :

```
print("12345\rAB")
```

Output:

9. Raw strings to ignore escape characters

Newbie vs. Escape Characters



Me trying to print a file path:
`"C:\newfolder\python\code.py"`



❌ Invalid escape sequence detected.

Newbie vs. Escape Characters



Use raw strings like
`r"C:\newfolder\python\code.py"`



✅ `C:\newfolder\python\code.py`

Code :

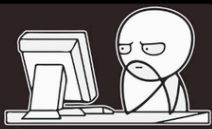
```
print(r"C:\\newfolder\\file.txt")
```

Output:

```
C:\\newfolder\\file.txt
```

10. Print the file-path.without using Raw strings

What just happened??

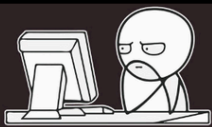
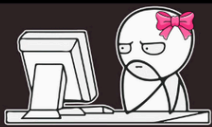


Code: `path = input("Input file location: ")`
`print(r(path))`

Output: ❌ `NameError: name 'r' is not defined`

⚠️ You can't use `r(path)` —
`r''` works only with hardcoded strings,
 not with variables.

Ask the user to use double backslashes:



Alright, thanks!!

Code :

```
print("C:\\newfolder\\python")
```

Output:

```
C:\newfolder\python
```

****A Bonus Question :**

10. Write code to use bell sound .

 a. Check system sound settings:

- Make sure your system sounds are not muted.
- On Mac, go to System Settings > Sound > Sound Effects > Play user interface sound effects (should be checked).
- On Windows, check Control Panel > Sound > Sounds tab — ensure "Default Beep" is enabled for system events.

 b. Try running in a terminal:

Windows :

```
python -c "print('Bell sound playing...\a')"
```

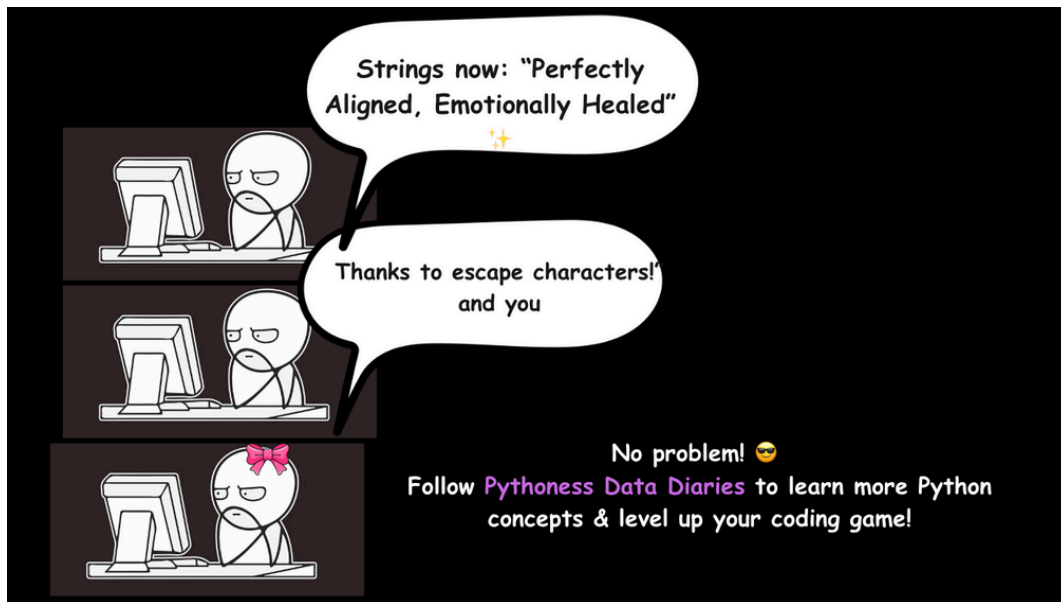
Mac :

```
bhawanasaxena@Bhawanas-MacBook-Air ~ % python3 -c "print('Bell sound playing...\a')"
```

Output :

(with a small beep)**

```
Bell sound playing...
```



Tips:

- `\n` = newline
- `\t` = tab space
- `\"` = double quote
- `\'` = single quote
- `\\` = backslash
- `\r` = carriage return (cursor goes to beginning)
- `\a` = bell sound (doesn't work on all terminal)
- `r"string"` = raw string (escape characters not processed)