

Print

To print the output

```
>> print("Hello world")
```

```
>> Hello world
```

You can use either single quotes or double quotes python will treat them as same.

If you need to print some string which is having single quotes within it like 'Harry's pen' Then you need to use double quotes otherwise it will show

```
>> print('Harry's pen')
```

```
SyntaxError : Invalid syntax
```

```
>> print("Harry's pen")
```

```
>> Harry's pen
```

You can print the values of the variable by

```
>> var = 12
```

```
>> print(var)
```

```
>> 12
```

You can also add string and variable in the print statement

```
>> var1 = 3
```

```
>> print('value of var1 is ' + var1)
```

```
>> value of var1 is 3
```

But best practice is

```
>> var1 = 3
```

```
>> print('Value of var1 is %s' %(var1))
```

```
>> Value of var1 is 3
```

You can also use %d or %f depending upon the datatype of the variable you are assigning to those places but %s is recommended.

```
>> var1 = 3  
  
>> print('value of var1 is %d' %(var1))  
  
>> value of var1 is 3
```

Example program:

```
>> num1 = 10  
  
>> num2 = 20  
  
>> print('The addition of %s and %s is %s' %(num1,num2,num1 + num2))  
  
>> The addition of 10 and 20 is 30
```

Formatting the print statement.

Let us say I need to print one quote with four lines then you need to use triple quotes like the following.

```
>> print(""" Alone we can  
Do so little;  
Together we can  
Do so much.""")  
  
>>> Alone we can  
Do so little;  
Together we can  
Do so much.
```

You can also do this by writing in a single row using escape sequence like

`\n` which will go to new line

`\t` which will give tab space to it

`\s` which will give space to it

```
>> print('Alone we can\nDo so little;\nTogether we can\nDo so much.')
```

```
>> Alone we can
```

```
Do so little;
```

```
Together we can
```

```
Do so much.
```

Let's say you want to print the file path

```
>> print('C:\user\bin')
```

```
SyntaxError : Unicode error
```

If you type like this it will take `\u` as escape sequence so you need to add `'\'` before `'\'` to tell that is not escape sequence.

```
>> print('C:\\user\\bin')
```

```
>> C:\user\bin
```

There is also other thing you can do. You can tell the print statement to print the entire statement without searching for escape sequence.

```
>> print(r'C:\user\bin')
```

```
>>C:\user\bin
```

This method is called passing the string as raw string it will not search for anything in the string. You just need to add `'r'` or `'R'` before the quotes in print statement.

.format() Method

```
>> num1 = 10
```

```
>> num2 = 20
```

```
>> print('The addition of {} and {} is {}'.format(num1,num2,num1 + num2))
```

```
>>The addition of 10 and 20 is 30
```

If we did not give any value between the brackets{} then first one will take the value of first variable in format and second will take the value of second variable in format. Otherwise,

```
>> print('The addition of {0} and {0} is {0}'.format(num1,num2,num1 + num2))
```

```
>> The addition of 10 and 10 is 10
```

```
>> print('The addition of {1} and {1} is {0}'.format(num1,num2,num1 + num2))
```

```
>> The addition of 20 and 20 is 10
```

The {0} will take the value of first variable in format method and replaces it in string.

You can also define the length of the string as shown below

```
>> print('The addition of {0:3d} and {1:3d} is  
{2:4d}'.format(num1,num2,num1 + num2))
```

```
>> The addition of _10 and _20 is _30.
```

```
>> num1 = 100
```

```
>> num2 = 200
```

```
>> print('The addition of {0:3d} and {1:3d} is  
{2:4d}'.format(num1,num2,num1 + num2))
```

```
>> The addition of 100 and 200 is _300.
```

*The length of
the output is
same*

In the above example you can see that {0} and {1} are replaced by 3 decimal places and {2} are replaced by 4 decimal spaces. You will not get “_” in output in order to show difference I have included that.

If the value exceeds the defined length the output will not be truncated

```
>> num1 = 10000
```

```
>> print('The addition of {0:3d} and {1:3d} is  
{2:4d}'.format(num1,num2,num1 + num2))
```

```
>> The addition of 10000 and 200 is 10200.
```

