

An **Escape sequence** is a combination of characters that is translated into another character or sequences of characters that may be difficult to represent directly.

```
print('What's your name')
```

SyntaxError: unterminated string literal (detected at line 1)

```
print('What\'s Your Name')
```

What's Your Name

```
print('What\'s Your Name?')
```

What's Your Name?

```
print("The boy replies ,\"My name is Anil\" ")
```

SyntaxError: invalid syntax. Perhaps you forgot a comma?

```
print("The boy replies ,\"My name is Anil\" ")
```

The boy replies ,\"My name is Anil\"

```
print("I have studied many languages \
```

```
Python is the Best")
```

```
>>> I have studied many languages Python is the Best
```

If a backslash is added at the end of the line then

it indicates the String is continued to the next line

'\o' & '\x' represent the values in Octal & Hexadecimal respectively.

## **String Formatting**

Syntax-

```
format(value,format_specifier)
```

```
format('Hello','<30')
```

```
'Hello'
```

```
format('Hello','>30')
```

```
'Hello'
```

```
print('Hello',format('-','-<10'),'World')
```

```
Hello ----- World
```

```
print('Hello',format('*', '*<20'),'World')
```

```
Hello ***** World
```

```
print('Hello',format('*', '-<10'),'World')
```

```
Hello *----- World
```

```
print('Hello',format('-', '-<10'),'World')
```

```
Hello -<10 World
```

```
print('Hello',format('-', '->10'),'World')
```

```
Hello ----- World
```

## **Variables & Identifiers**

Variables are reserved memory location that stores values

There are some rules for declaring a variable --

1. The first character of an Identifier/Variable must be an

'\_'underscore or a letter (Uppercase or Lowercase)

2. The variable names are case sensitive

## Checking for Valid names:

1. 1num
2. my-var
3. Basic Sal
4. %tag
5. tag123
6. var\_28
7. HR&A

## Datatypes:

We have seen that variables can hold values of different types called Datatype

Python has different Standard datatypes that are used to define the operation possible on them.

Based on the datatypes interpreter reserves memory in it and also determines the size of the memory required.

We can also define our datatypes like Classes There are 5 standard datatypes supported by Python

Numbers, List, String, Tuple, & Dictionary.

Python is purely object oriented , it refers to everything as a object including numbers and strings.

```
num = 7
```

```
amt = 123.45
```

```
code = 'A'
```

```
pi = 3.1415926536
```

```
msg = "Hi"
```

```
print("num = ",num,type(num))
```

```
print("amt = ",amt,type(amt))
```

```
print("code = ",code,type(code))
```

```
print("PI = ",pi,type(pi))
```

```
print("Msg = ",msg,type(msg))
```

```
print("num = "+str(num))
```

Whenever putting '+' symbol between Strings we need to convert

the variable into String before the printing the Statement

```
num = 7 <class 'int'>
amt = 123.45 <class 'float'>
code = A <class 'str'>
PI = 3.1415926536 <class 'float'>
Msg = Hi <class 'str'>
Everything is class & object <class 'int'>
```

### **Multiple Assignment**

```
sum = flag = a = b = 0
sum,a,b,msg = 0,3,5,'result' Multiple assignemt
print("sum = ",sum,"a = ",a,"b = ",b,"msg = ",msg)
>>> sum = 0 a = 3 b = 5 msg = result
```

### **Variable Deletion**

```
str = "Hello"
num = 10
age = 20
print(str)
Hello
print(num+age)
30
del num
print(num)
```

### **Boolean**

Boolean is another datatype in python it can have either of the 2 values TRUE or FALSE.

```
bool_var = True
```

```
print(bool_var)
```

True

```
20!=30
```

True

```
20==30
```

False

```
90<=90
```

True

```
87>87.0
```

False

```
87<87.0
```

False

```
"Python" != "Python 3.4"
```

True

```
"Python" == "Python"
```

True