**Python Comments**[**¶**](#gjdgxs)

When writing code in Python, it’s important to make sure that your code can be easily understood by others. Giving variables obvious names, defining explicit functions, and organizing your code are all great ways to do this. Another awesome and easy way to increase the readability of your code is by using comments!

Comments are very important while writing a program.

It describes what's going on inside a program so that a person looking at the source code does not have a hard time figuring it out. If you just wrote Which is program and how the program will get executed, other person reviews the code in easiest way.

In python, we use HASH (#) symbol to start writing comment

In [1]:

*#Program to insert comment*  
  
*#print hello*   
print("Hello")  
print("bye")

Hello  
bye

Multi-Line comment

If we want to comment more than one statement, there are two ways to give comment to statements

First is, give hash(#) in the beginning of each line.

In [2]:

*#Program to insert multi-line comment*  
  
*# Print hello*   
*# Also print your name*  
  
print("Hello Python")

Hello Python

You can also comment more then one line by giving three double quotes in the begnning of line and end with three double quotes at the end of line.

In [3]:

*"""*  
*If I really hate pressing 'enter' and*  
*typing all those hash marks, I could just do this instead*  
*"""*  
print("This is easy way to give comments to multi-line statements")

This is easy way to give comments to multi-line statements

**Python Indentation**[**¶**](#30j0zll)

1. Most of the programming languages like C, C++, Java use braces { } to define a block of code. Python uses indentation.
2. A code block (body of a function, loop etc.) starts with indentation and ends with the first unindented line. The amount of indentation is up to you, but it must be consistent throughout that block.
3. Generally four whitespaces are used for indentation and is preferred over tabs.

In [4]:

**for** i **in** range(10):  
 print (i)

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

Indentation can be ignored in line continuation. But it's a good idea to always indent. It makes the code more readable.

In [5]:

**for** i **in** range (5):  
 print ("I am in loop")  
print("I am out of loop")

I am in loop  
I am in loop  
I am in loop  
I am in loop  
I am in loop  
I am out of loop

In [6]:

**if** **True**:  
 print ("Machine Learning")  
 c = "AAIC"  
 print(c)

Machine Learning  
AAIC

**Python Statements**[**¶**](#1fob9te)

Different instructions that a Python interpreter can execute is called as Statements

In [7]:

*#For example*  
  
asd = "Hello" *#single statements*  
print(asd)

Hello

**Multi-Line Statements**[**¶**](#3znysh7)

In Python, end of a statement is marked by a newline character.

But we can make a statement extend over multiple lines with the line continuation character ().

In [8]:

d = 1 + 3 + 5 + 7 + \  
 9 + 11 + 13 + \  
 15 + 17 + 19  
   
print(d)

100

In [9]:

*# Another way is*  
  
s = (0 + 2 + 4 + 6 + 8 +   
 10 + 12 + 14 + 16 +  
 18)  
  
print(s)

90

We can also put multiple statements in one line by using ; symbol

In [10]:

*#For example*  
  
w = 10; e = 24; \_R = 14  
  
print(w)  
print(e)  
print(\_R)

10  
24  
14

**Assignments**[**¶**](#2et92p0)

1)Write a multiline comment on Python programming language.