Python - Comments

Python comments are programmer-readable explanation or annotations in the Python source code. They are added with the purpose of making the source code easier for humans to understand, and are ignored by Python interpreter. Comments enhance the readability of the code and help the programmers to understand the code very carefully.

Just like most modern languages, Python supports single-line (or end-of-line) and multi-line (block) comments. Python comments are very much similar to the comments available in PHP, BASH and Perl Programming languages.

There are three types of comments available in Python

- Single line Comments
- Multiline Comments
- Docstring Comments

Single Line Comments

A hash sign (#) that is not inside a string literal begins a comment. All characters after the # and up to the end of the physical line are part of the comment and the Python interpreter ignores them.

Example

Following is an example of a single line comment in Python:

```
# This is a single line comment in python

print ("Hello, World!")
```

Edit & Run

This produces the following result –

```
Hello, World!
```

You can type a comment on the same line after a statement or expression –

```
name = "Madisetti" # This is again comment
```

Multi-Line Comments

Python does not provide a direct way to comment multiple line. You can comment multiple lines as follows –

```
# This is a comment.
# This is a comment, too.
# This is a comment, too.
# I said that already.
```

Following triple-quoted string is also ignored by Python interpreter and can be used as a multiline comments:

```
'''
This is a multiline
comment.
'''
```

Example

Following is the example to show the usage of multi-line comments:

```
'''
This is a multiline
comment.
'''

print ("Hello, World!")
```

Edit & Run

This produces the following result –

```
Hello, World!
```

Docstring Comments

Python docstrings provide a convenient way to provide a help documentation with Python modules, functions, classes, and methods. The **docstring** is then made available via the __doc__ attribute.

```
def add(a, b):
    """Function to add the value of a and b"""
    return a+b
```

Edit & Run

```
print(add.__doc__)
```

This produces the following result –

```
Function to add the value of a and b
```