Statement, Indentation and Comment in Python

Here, we will discuss Statements in Python, Indentation in Python, and Comments in Python. We will also discuss different rules and examples for Python Statement, Python Indentation, Python Comment, and the Difference Between 'Docstrings' and 'Multi-line Comments.

What is Statement in Python

A **Python statement** is an instruction that the Python interpreter can execute. There are different types of **statements in Python** language as Assignment statements, Conditional statements, Looping statements, etc. The token character NEWLINE is used to end a statement in Python. It signifies that each line of a Python script contains a statement. These all help the user to get the required output.

Types of statements in Python?

The different types of Python statements are listed below:

- Multi-Line Statements
- Python Conditional and Loop Statements
  - Python If-else
  - Python for loop
  - Python while loop
  - Python try-except
  - Python with statement
- Python Expression statements
  - Python pass statement
  - Python del statement
  - Python return statement
  - Python import statement
  - Python continue and
  - Python break statement

Example:

Statement in Python can be extended to one or more lines using parentheses (), braces {}, square brackets [], semi-colon (;), and continuation character slash (\). When the programmer needs to do long calculations and cannot fit his statements into one line, one can make use of these characters.

```
Declared using Continuation Character (\):
s = 1 + 2 + 3 + \
    4 + 5 + 6 + \
    7 + 8 + 9


Declared using parentheses () :
```

```
n = (1 * 2 * 3 + 7 + 8 + 9)

Declared using square brackets [] :
footballer = ['MESSI',
          'NEYMAR',
          'SUAREZ']

Declared using braces {} :
x = {1 + 2 + 3 + 4 + 5 + 6 +
     7 + 8 + 9}

Declared using semicolons(;) :
flag = 2; ropes = 3; pole = 4
```

What is Indentation in Python

Whitespace is used for **indentation in Python**. Unlike many other programming languages which only serve to make the code easier to read, **Python indentation** is mandatory. One can understand it better by looking at an example of indentation in Python.

Role of Indentation in Python

A block is a combination of all these statements. Block can be regarded as the grouping of statements for a specific purpose. Most programming languages like C, C++, and Java use braces { } to define a block of code for indentation. One of the distinctive roles of Python is its use of indentation to highlight the blocks of code. All statements with the same distance to the right belong to the same block of code. If a block has to be more deeply nested, it is simply indented further to the right.

Example 1:

The lines print('Logging on to geeksforgeeks...') and print('retype the URL.') are two separate code blocks. The two blocks of code in our example if-statement are both indented four spaces. The final print('All set!') is not indented, so it does not belong to the else-block.

- Python3

```
# Python indentation


site = 'gfg'


if site == 'gfg':
```

```
    print('Logging on to geeksforgeeks...')
else:
    print('retype the URL.')
print('All set !')
```

Output

```
Logging on to geeksforgeeks...
All set !
```

Example 2:
To indicate a block of code in Python, you must indent each line of the block by the same whitespace. The two lines of code in the while loop are both indented four spaces. It is required for indicating what block of code a statement belongs to. For example, j=1 and while(j<=5): is not indented, and so it is not within the while block. So, Python code structures by indentation.

- Python3

```
j = 1
while(j <= 5):
    print(j)
    j = j + 1
```

Output

```
1
2
3
4
5
```

What are Comments in Python

Python comments start with the hash symbol # and continue to the end of the line. Comments in Python are useful information that the developers provide to make the reader understand the source code. It explains the logic or a part of it used in the code. Comments in Python are usually helpful to someone maintaining or enhancing your code when you are no longer around to answer questions about it. These are often cited as useful programming convention that does not take part in the output of the program but improves the readability of the whole program.
Comments in Python are identified with a hash symbol, #, and extend to the end of the line.

Types of comments in Python

A comment can be written on a single line, next to the corresponding line of code, or in a block of multiple lines. Here, we will try to understand examples of comment in Python one by one:

Single-line comment in Python

Python single-line comment starts with a hash symbol (#) with no white spaces and lasts till the end of the line. If the comment exceeds one line then put a hashtag on the next line and continue the comment. Python's single-line comments are proved useful for supplying short explanations for variables, function declarations, and expressions. See the following code snippet demonstrating single line comment:

Example 1:
Python allows comments at the start of lines, and Python will ignore the whole line.

- Python3

```
# This is a comment

# Print “GeeksforGeeks” to console

print("GeeksforGeeks")
```

Output

```
GeeksforGeeks
```

Example 2:
Python also allows comments at the end of lines, ignoring the previous text.

- Python3

```
a, b = 1, 3   # Declaring two integers

sum = a + b   # adding two integers

print(sum)  # displaying the output
```

Output

```
4
```

Multiline comment in Python

Use a hash **(#)** for each extra line to create a multiline comment. In fact, Python multiline comments are not supported by Python's syntax. Additionally, we can use Python multi-line comments by using multiline strings. It is a piece of text enclosed in a delimiter (“””) on each end of the comment. Again there should be no white space between delimiter (“””). They are useful when the comment text does not fit into one line; therefore need to span across lines. Python Multi-line comments or paragraphs serve as documentation

for others reading your code. See the following code snippet demonstrating a multi-line comment:

Example 1:

In this example, we are using an extra # for each extra line to create a Python multiline comment.

- Python3

```
# This is a comment

# This is second comment

# Print “GeeksforGeeks” to console

print("GeeksforGeeks")
```

Output

```
GeeksforGeeks
```

Example 2:

In this example, we are using three double quotes (“) at the start and end of the string without any space to create a Python multiline comment.

- Python3

```
"""

This would be a multiline comment in Python that

spans several lines and describes geeksforgeeks.

A Computer Science portal for geeks. It contains

well written, well thought

and well-explained computer science

and programming articles,

quizzes and more.

…

"""

print("GeeksForGeeks")
```

Output

```
GeeksForGeeks
```

Example 3:

In this example, we are using three single quotes (‘) at the start and end of the string without any space to create a Python multiline comment.

- Python3

```
'''This article on geeksforgeeks gives you a

perfect example of

multi-line comments'''


print("GeeksForGeeks")
```

Output

```
GeeksForGeeks
```

Docstring in Python

Python Docstrings are a type of comment that is used to show how the program works. Docstrings in Python are surrounded by Triple Quotes in Python (“”” “”””). Docstrings are also neglected by the interpreter.

- Python3

```
# program illustrates the use of docstrings


def helloWorld():

  # This is a docstring comment

    """ This program prints out hello world """

    print("Hello World")



helloWorld()
```

Output

```
Hello World
```

Difference Between ‘Docstrings’ and ‘Multi-line Comments

Docstrings and Multi-line comments may look the same but they aren’t.

- Docstrings are written in the functions and classes to show how to use the program.
- Multi-line comments are used to show how a block of code works.