Comments in Python

Comments are descriptions that help programmers better understand the intent and functionality of the program.

They are completely ignored by the Python interpreter.

Advantages of Using Comments

Using comments in programs makes our code more understandable. It makes the program more readable which helps us remember why certain blocks of code were written.

Other than that, comments can also be used to ignore some code while testing other blocks of code. This offers a simple way to prevent the execution of some lines or write a quick pseudo-code for the program.

Single-Line Comments in Python

In Python, we use the hash symbol `#` to write a single-line comment.

Example 1: Writing Single-Line Comments

```
# printing a string
print('AECS-1,Jaduguda')
```

Output

```
AECS-1,Jaduguda
```

Here, the comment is:

```
# printing a string
```

This line is ignored by the Python interpreter.

Everything that comes after `#` is ignored. So, we can also write the above program in a single line as:

```
print('AECS-1,Jaduguda') #printing a string
```

The output of this program will be the same as in **Example 1**. The interpreter ignores all the text after `#`.

Multi-Line Comments in Python

Python doesn't offer a separate way to write multiline comments. However, there are other ways to get around this issue.We can use `#` at the beginning of each line of comment on multiple lines.

Example 2: Using multiple

```
# it is a
# multiline
# comment
```

Here, each line is treated as a single comment and all of them are ignored.

String Literals for Multi-line Comments

Even though there is no unique way to write multiline comments in Python, we know that the Python interpreter ignores the string literals that are not assigned to a variable.

So, we can even write a single-line comment as:

```
#this is a comment
'this is an unassigned string as a comment '
```

Here, we can see that the second line of the program is a string but is not assigned to any variable or function. So, the interpreter ignores the string.

In a similar way, we can use multiline strings (triple quotes) to write multiline comments.

The quotation character can either be `'` or `"`.

Example 3: Using String Literals to write Multi-line Comments

```
'''
I am a
multiline comment!
'''
print("AECS-1,Jaduguda")
```

Here, the multiline string isn't assigned to any variable, so it is ignored by the interpreter. Even though it is not technically a multiline comment, it can be used as one.