Python pass Statement

In Python programming, the `pass` statement is a null statement which can be used as a placeholder for future code.

Suppose we have a loop or a function that is not implemented yet, but we want to implement it in the future. In such cases, we can use the `pass` statement.

The syntax of the `pass` statement is:

```
pass
```

Using pass With Conditional Statement

```
n = 10

# use pass inside if statement
if n > 10:
    pass

print('Hello')
```

Here, notice that we have used the `pass` statement inside the if statement .

However, nothing happens when the pass is executed. It results in no operation (NOP).

Suppose we didn't use pass or just put a comment as:

```
n = 10

if n > 10:
    # write code later

print('Hello')
```

Here, we will get an error message: `IndentationError: expected an indented block`

Note: The difference between a comment and a `pass` statement in Python is that while the interpreter ignores a comment entirely, `pass` is not ignored.

Use of pass Statement inside Function or Class

We can do the same thing in an empty function or class as well. For example,

```
def function(args):
    pass

class Example:
    pass
```

Also Read:

- Python break and continue

Table of Contents