`pass` Statement

The `pass` statement does nothing. It can be used in a region of code where the program needs some statements for it to execute and at a time when you as the program writer has not yet thought of the actual statements to write. In other words, the `pass` statement acts as a placeholder for the statements that you are about to write. It is used in cases where you know you want to write a function, so you write the function header, but you do not yet know what statements to write for that function; therefore, you first write the `pass` statement.

```python
# foo.py

# at this point, you know that you have to write some function that needs to do "something".
# however, you have not thought of the code that you need to write yet, so you write the `pass` statement.
def some_function():
    pass

# some meaningless function that returns integer 2.
def other_function():
    return 1 + 1

if __name__ == "__main__":
    some_function()
    other_function()
```

In the above example, you know that you have to call `some_function()` before the `other_function()`, and you have already written some code for the `other_function()`, but you don't know what to write for the body of `some_function()`, so you write the `pass` statement. If you do not write the `pass` statement in `some_function()`, the program will produce an error when you execute it so you will not even be able to execute `other_function()`.