Python - Assert Statement

Python provides the **assert** statement to check if a given logical expression is true or false. Program execution proceeds only if the expression is true and raises the **AssertionError** when it is false. The following code shows the usage of the assert statement.

Example: assert

```
num=int(input('Enter a number: '))
assert num>=0
print('You entered: ', num)
```

The print statement will display if the entered number is greater than or equal to 0. Negative numbers result in aborting the program after showing the AssertionError.

Result:

```
Enter a number: 100
You entered 100

Enter a number: -10
Traceback (most recent call last):
  File "C:/python36/xyz.py", line 2, in <module>
    assert num>=0
AssertionError
```

The assert statement can optionally include an error message string, which gets displayed along with the AssertionError. In the above code, you can change the assert statement to the following and run the code:

Example: assert

```
num=int(input('Enter a number: '))
assert num>=0, "Only positive numbers accepted."
print('You entered: ', num)
```

Result:

```
Enter a number: -10
Traceback (most recent call last):
  File "C:/python36/xyz.py", line 2, in <module>
    assert num>=0, "Only positive numbers accepted."
    AssertionError: Only positive numbers accepted.
```

The AssertionError is also a built-in exception and can be handled using the try...except construct as follows:

Example: AssertionError

```
try:
    num=int(input('Enter a number: '))
    assert(num >=0), "Only positive numbers accepted."
    print(num)
except AssertionError as msg:
    print(msg)
```

When the input causes an AssertionError, the program will not terminate, as was the case earlier. Instead, it will be handled by the except block. The error message in the assert statement will be passed as argument to the exception argument msg, using keyword *as*.

Result:

```
Enter a number: -10
Only positive numbers accepted.
```