

8 Patching Test Cases

Patching Test Cases

The `patch` callable is commonly used during testing. Patch can decorate all or individual methods by applying the decorator to classes or methods.

```
1 import unittest
2 from unittest.mock import call, patch
3 def greeter(name: str):
4     ''' A function used to demonstrate the use of the patch call
5     print(f'Hello, {name}')
```

6 def greet_everyone(*names):

7 ''' A function used to demonstrate the use of the patch call

8 for name in names:

9 print(f'Hello, {name}')

10 ##### Patching Methods #####

11 #

12 class GreeterTest(unittest.TestCase):

13 # Replace the built-in print function with a Mock for this

14 @patch('builtins.print')

15 def test_greeter(self, print_mock):

16 # The patch decorator passes mocked objects as input pa

17 #

18 # With the print function replaced with a mock for the

19 # this method no data is printed to the console.

20 greeter('World')

21 # Inspect the mock version of the print function and de

22 # was provided with the expected input.

23 print_mock.assert_called_with('Hello, World')

24 print_mock.assert_called_once()

25 ##### Patching Classes #####

26 #

27 # Replace the built-in print function for all methods of this c

28 @patch('builtins.print')

< Back

Start check ↻





```
25 ##### Patching Classes #####
26 #
27 # Replace the built-in print function for all methods of this class
28 @patch(f'builtins.print')
29 class GreetEveryoneTest(unittest.TestCase):
30     def test_greeter(self, print_mock):
31         # The patch decorator passes mocked objects as input parameters
32         greeter('World')
33         # Inspect the mock version of the print function and determine
34         # what was provided with the expected input.
35         print_mock.assert_called_with('Hello, World')
36         print_mock.assert_called_once()
37     def test_greet_everyone(self, print_mock):
38         # The patch decorator passes mocked objects as input parameters
39         greet_everyone('World', 'Universe', 'Multiverse', 'Ultraverse')
40         # The expected calls made to the mock implementation of print
41         expect = [
42             call('Hello, World'),
43             call('Hello, Universe'),
44             call('Hello, Multiverse'),
45             call('Hello, Ultraverse'),
46         ]
47         # Ensure the calls made to print are passed the expected arguments
48         assert print_mock.mock_calls == expect
49         assert print_mock.call_count == len(expect)
50 #####
51 if __name__ == '__main__':
52     unittest.main(verbosity=2, failfast=True)
```

Instructions

1. Arrange the **playground.py** file to match the code above.
2. Run **playground.py** in the IDE's terminal pane.

```
1 python3 cloudatacademy/playground.py
```

< Back

Start check