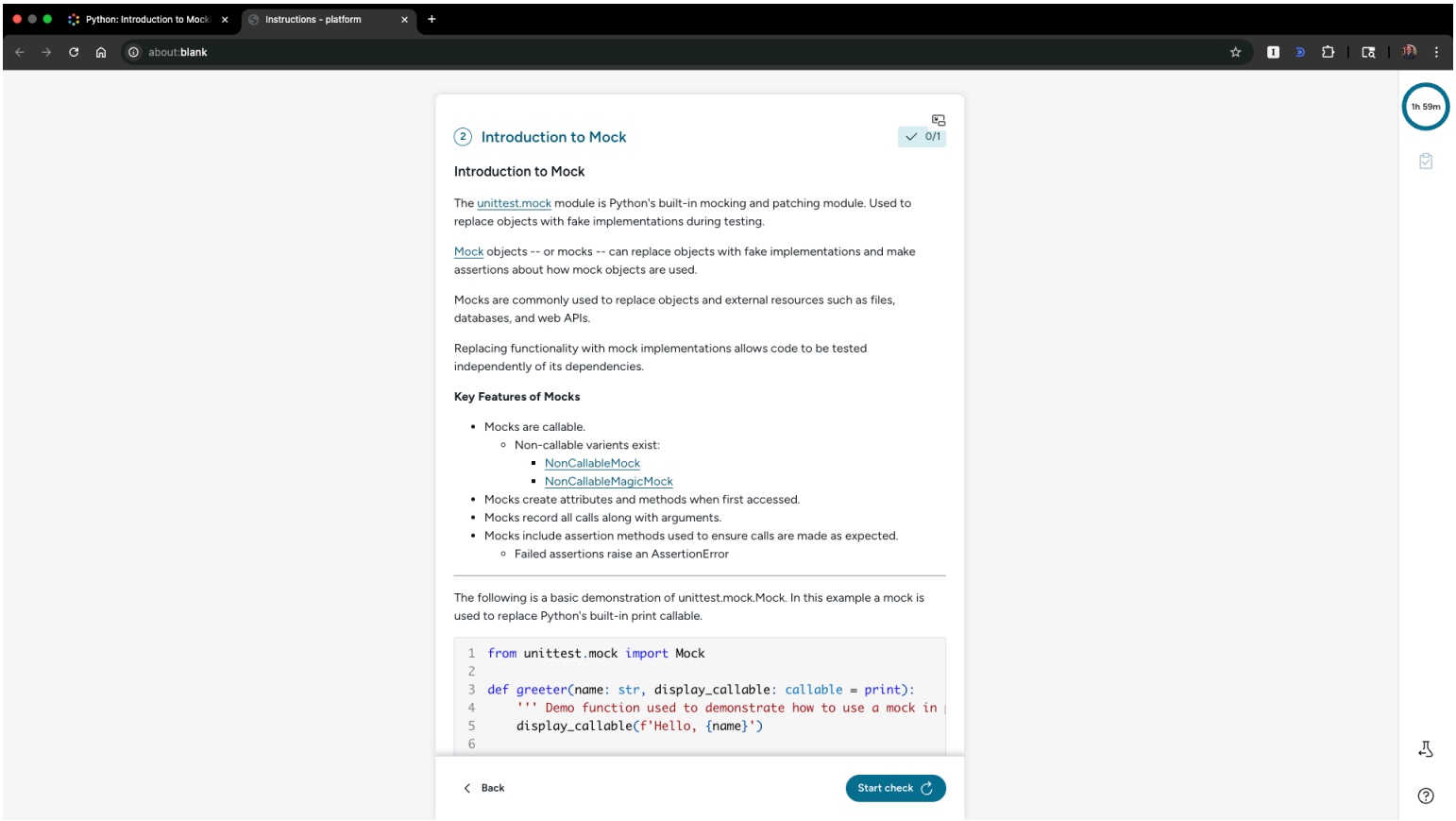

Python: Introduction to Mock
Instructions - platform
about:blank
1h 59m
2 Introduction to Mock
0/1
Introduction to Mock
The unittest.mock module is Python's built-in mocking and patching module. Used to replace objects with fake implementations during testing.
Mock objects -- or mocks -- can replace objects with fake implementations and make assertions about how mock objects are used.
Mocks are commonly used to replace objects and external resources such as files, databases, and web APIs.
Replacing functionality with mock implementations allows code to be tested independently of its dependencies.
Key Features of Mocks
Mocks are callable.
Non-callable varients exist:
NonCallableMock
NonCallableMagicMock
Mocks create attributes and methods when first accessed.
Mocks record all calls along with arguments.
Mocks include assertion methods used to ensure calls are made as expected.
Failed assertions raise an AssertionError
The following is a basic demonstration of unittest.mock.Mock. In this example a mock is used to replace Python's built-in print callable.
1 from unittest.mock import Mock
2
3 def greeter(name: str, display_callable: callable = print):
4 ''' Demo function used to demonstrate how to use a mock in
5 display_callable(f'Hello, {name}')
6
Back
Start check

Python: Introduction to Mock | Instructions - platform

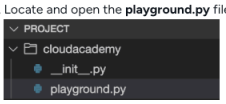
about:blank

1h 59m

```python
from unittest.mock import Mock

def greeter(name: str, display_callable: callable = print):
    ''' Demo function used to demonstrate how to use a mock in
    display_callable(f'Hello, {name}')

# This will display Hello, World in the console because the bui
# function is the default argument for the display_callable par
greeter('World')
# Call again passing a mock as the display_callable.
display = Mock()
# This will not display in the console because the mock is call
# place of the print function.
greeter('World', display)
# Mocks include different assertion methods used to determine i
# mock implementation is called as expected.
# Verify that the mock implementation of print is called with t
# argument value.
display.assert_called_with('Hello, World')

#################################################################
print('No assertion errors')
```

Instructions

1. Locate and open the **playground.py** file using the **IDE's Explorer/project** pane.



2. Arrange the **playground.py** file to match the code above.
3. Run **playground.py** in the **IDEs terminal** pane: **(Terminal > New Terminal)**

```
python3 cloudacademy/playground.py
```

```
Hello, World
```

Back

Start check