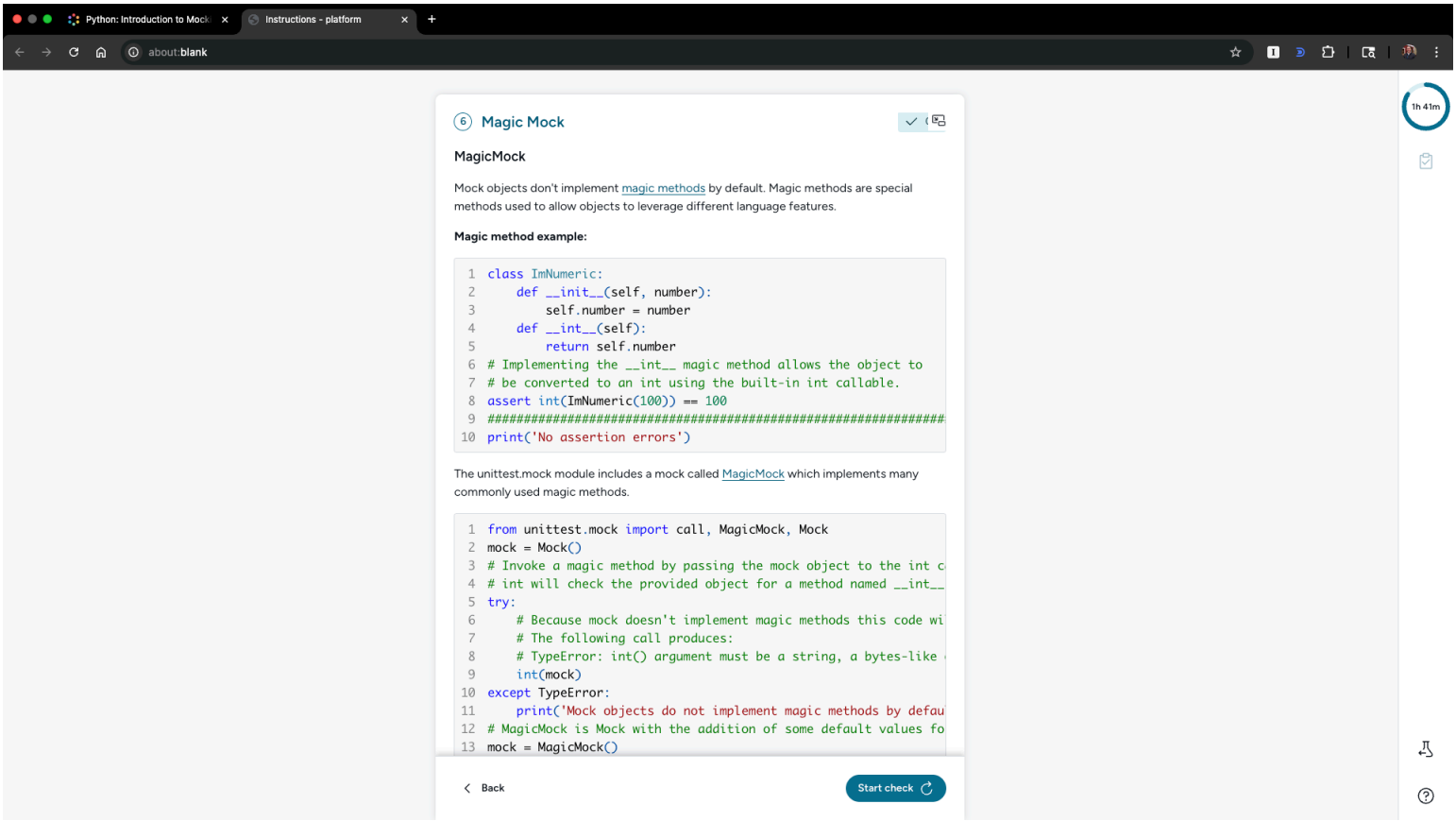
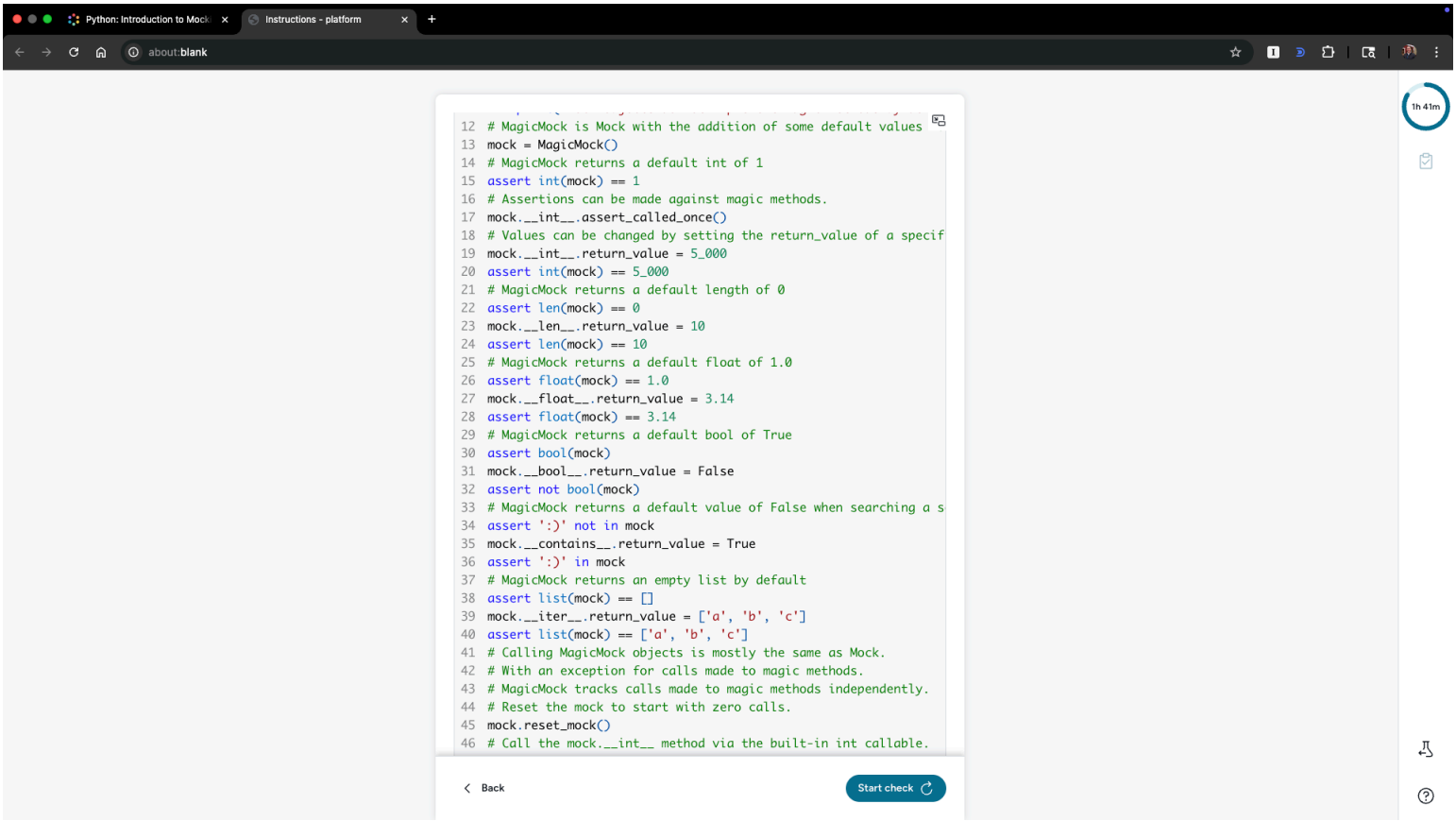
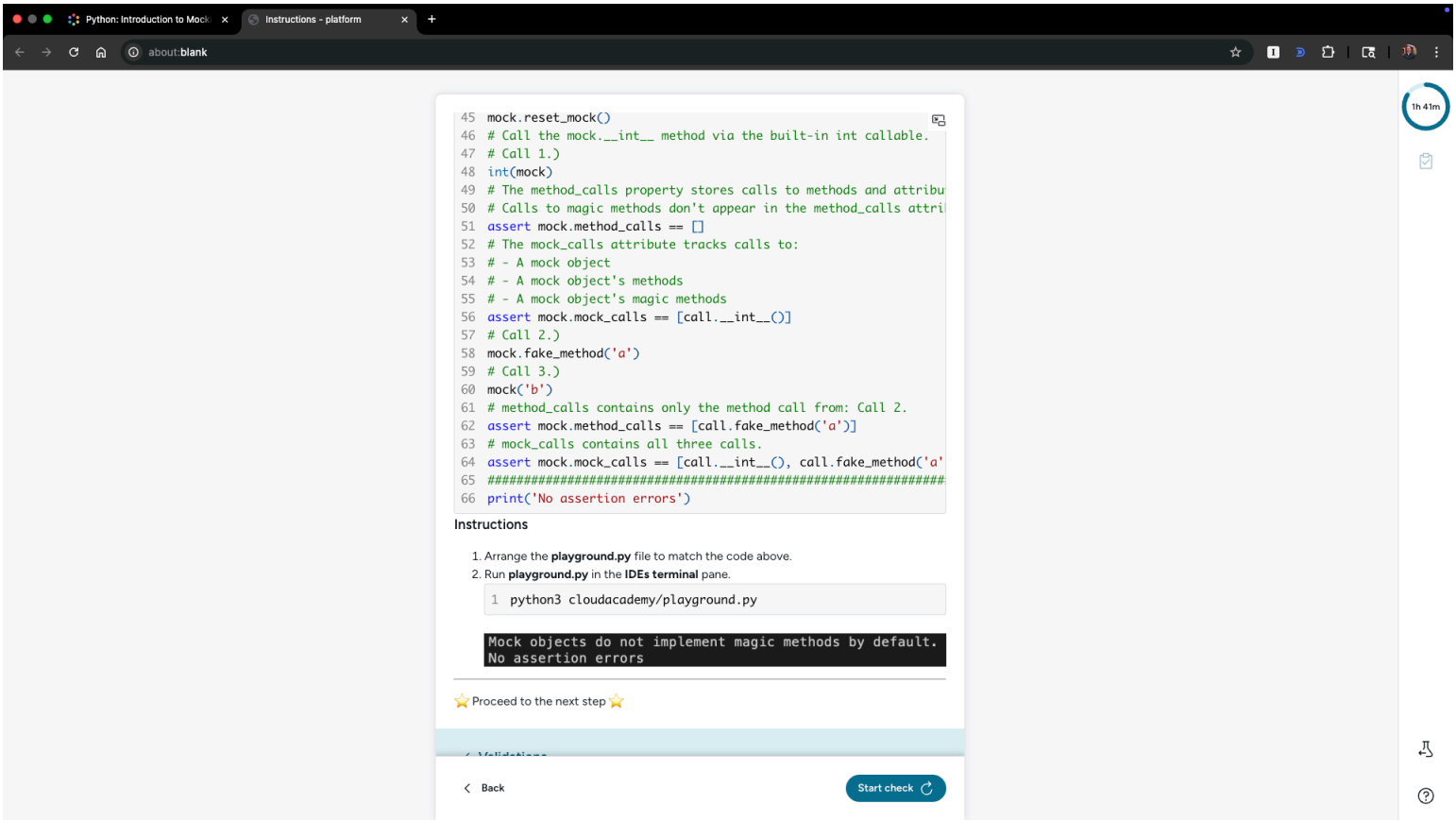

Python: Introduction to Mock
Instructions - platform
about:blank
6 Magic Mock
MagicMock
Mock objects don't implement magic methods by default. Magic methods are special methods used to allow objects to leverage different language features.
Magic method example:
class ImNumeric:
def __init__(self, number):
self.number = number
def __int__(self):
return self.number
Implementing the __int__ magic method allows the object to
be converted to an int using the built-in int callable.
assert int(ImNumeric(100)) == 100
print('No assertion errors')
The unittest.mock module includes a mock called MagicMock which implements many commonly used magic methods.
from unittest.mock import call, MagicMock, Mock
mock = Mock()
# Invoke a magic method by passing the mock object to the int c
# int will check the provided object for a method named __int__
try:
# Because mock doesn't implement magic methods this code wi
# The following call produces:
# TypeError: int() argument must be a string, a bytes-like
int(mock)
except TypeError:
print('Mock objects do not implement magic methods by defau
# MagicMock is Mock with the addition of some default values fo
mock = MagicMock()
Back
Start check
1h 41m


Python: Introduction to Mock
Instructions - platform
about:blank
1h 41m
# MagicMock is Mock with the addition of some default values
mock = MagicMock()
# MagicMock returns a default int of 1
assert int(mock) == 1
# Assertions can be made against magic methods.
mock.__int__.assert_called_once()
# Values can be changed by setting the return_value of a specif
mock.__int__.return_value = 5_000
assert int(mock) == 5_000
MagicMock returns a default length of 0
assert len(mock) == 0
mock.__len__.return_value = 10
assert len(mock) == 10
MagicMock returns a default float of 1.0
assert float(mock) == 1.0
mock.__float__.return_value = 3.14
assert float(mock) == 3.14
MagicMock returns a default bool of True
assert bool(mock)
mock.__bool__.return_value = False
assert not bool(mock)
MagicMock returns a default value of False when searching a s
assert ':)' not in mock
mock.__contains__.return_value = True
assert ':)' in mock
MagicMock returns an empty list by default
assert list(mock) == []
mock.__iter__.return_value = ['a', 'b', 'c']
assert list(mock) == ['a', 'b', 'c']
# Calling MagicMock objects is mostly the same as Mock.
# With an exception for calls made to magic methods.
# MagicMock tracks calls made to magic methods independently.
# Reset the mock to start with zero calls.
mock.reset_mock()
# Call the mock.__int__ method via the built-in int callable.
Back
Start check


Python: Introduction to Mock
Instructions - platform
about:blank
1h 41m
45 mock.reset_mock()
46 # Call the mock.__int__ method via the built-in int callable.
47 # Call 1.)
48 int(mock)
49 # The method_calls property stores calls to methods and attribu
50 # Calls to magic methods don't appear in the method_calls attri
51 assert mock.method_calls == []
52 # The mock_calls attribute tracks calls to:
53 # - A mock object
54 # - A mock object's methods
55 # - A mock object's magic methods
56 assert mock.mock_calls == [call.__int__()]
57 # Call 2.)
58 mock.fake_method('a')
59 # Call 3.)
60 mock('b')
61 # method_calls contains only the method call from: Call 2.
62 assert mock.method_calls == [call.fake_method('a')]
63 # mock_calls contains all three calls.
64 assert mock.mock_calls == [call.__int__(), call.fake_method('a'
65 ##
66 print('No assertion errors')
Instructions
1. Arrange the playground.py file to match the code above.
2. Run playground.py in the IDEs terminal pane.
1 python3 cloudacademy/playground.py
Mock objects do not implement magic methods by default.
No assertion errors
Proceed to the next step
Back
Start check