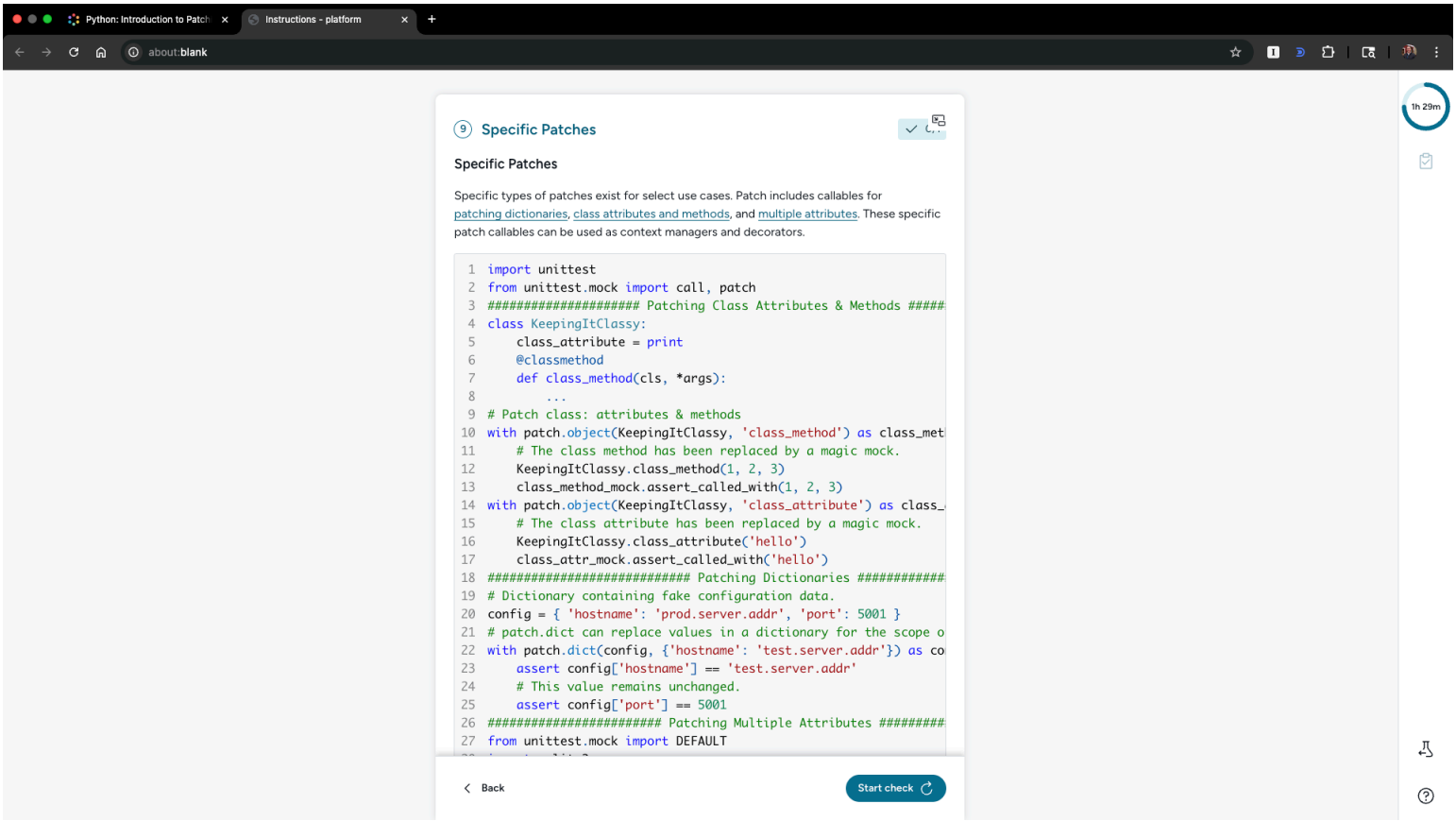
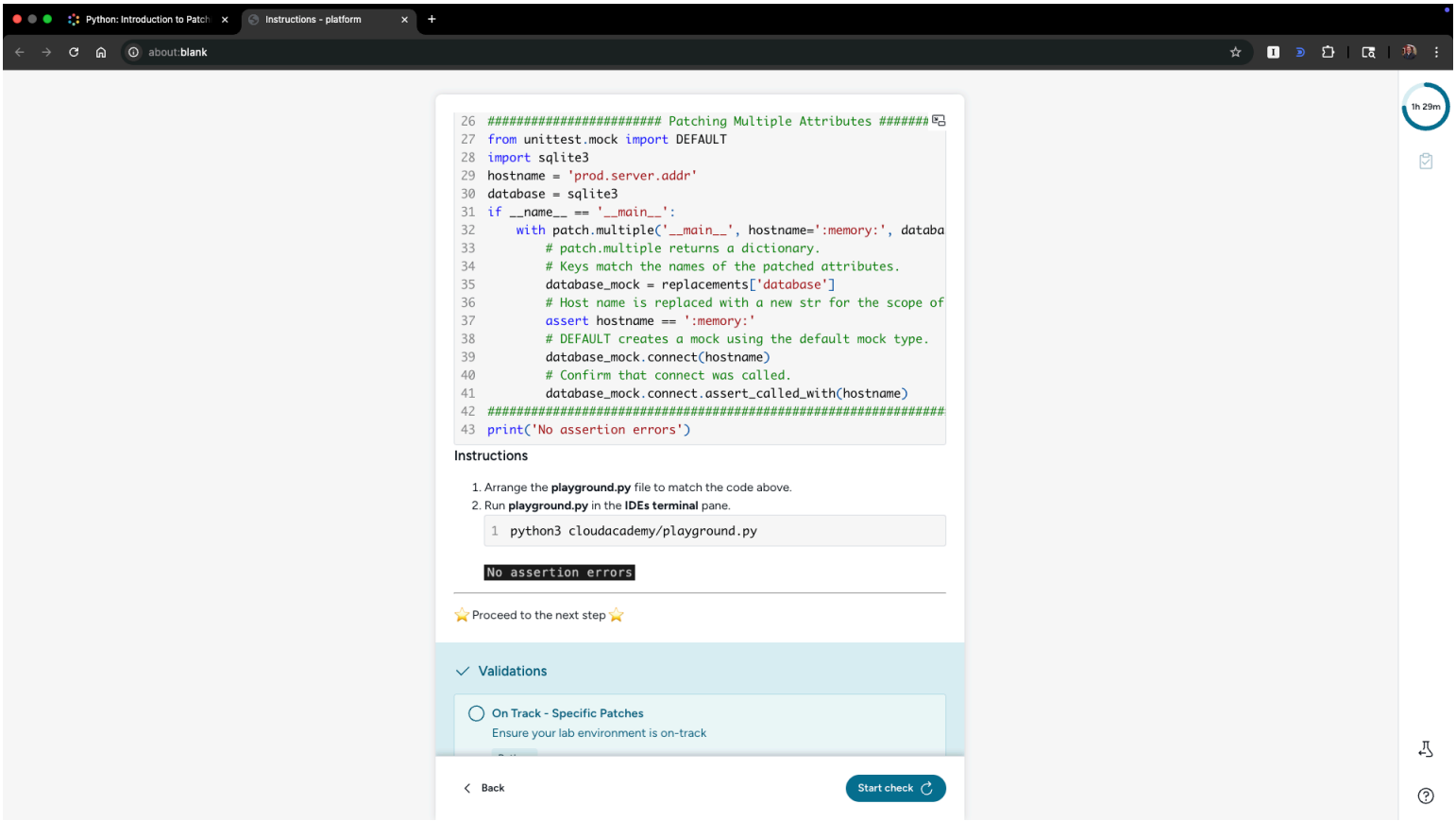

Python: Introduction to Patch
Instructions - platform
about:blank
1h 29m
9 Specific Patches
Specific Patches
Specific types of patches exist for select use cases. Patch includes callables for patching dictionaries, class attributes and methods, and multiple attributes. These specific patch callables can be used as context managers and decorators.
import unittest
from unittest.mock import call, patch
##################### Patching Class Attributes & Methods #####
class KeepingItClassy:
class_attribute = print
@classmethod
def class_method(cls, *args):
...
Patch class: attributes & methods
with patch.object(KeepingItClassy, 'class_method') as class_met
# The class method has been replaced by a magic mock.
KeepingItClassy.class_method(1, 2, 3)
class_method_mock.assert_called_with(1, 2, 3)
with patch.object(KeepingItClassy, 'class_attribute') as class_
# The class attribute has been replaced by a magic mock.
KeepingItClassy.class_attribute('hello')
class_attr_mock.assert_called_with('hello')
########################### Patching Dictionaries ############
# Dictionary containing fake configuration data.
config = { 'hostname': 'prod.server.addr', 'port': 5001 }
# patch.dict can replace values in a dictionary for the scope o
with patch.dict(config, {'hostname': 'test.server.addr'}) as co
assert config['hostname'] == 'test.server.addr'
This value remains unchanged.
assert config['port'] == 5001
######################## Patching Multiple Attributes #########
from unittest.mock import DEFAULT
Back
Start check


Python: Introduction to Patch
Instructions - platform
about:blank
26 ####################### Patching Multiple Attributes ######
27 from unittest.mock import DEFAULT
28 import sqlite3
29 hostname = 'prod.server.addr'
30 database = sqlite3
31 if __name__ == '__main__':
32 with patch.multiple('__main__', hostname=':memory:', databa
33 # patch.multiple returns a dictionary.
34 # Keys match the names of the patched attributes.
35 database_mock = replacements['database']
36 # Host name is replaced with a new str for the scope of
37 assert hostname == ':memory:'
38 # DEFAULT creates a mock using the default mock type.
39 database_mock.connect(hostname)
40 # Confirm that connect was called.
41 database_mock.connect.assert_called_with(hostname)
42 ##
43 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
No assertion errors
Proceed to the next step
Validations
On Track - Specific Patches
Ensure your lab environment is on-track
Back
Start check
1h 29m