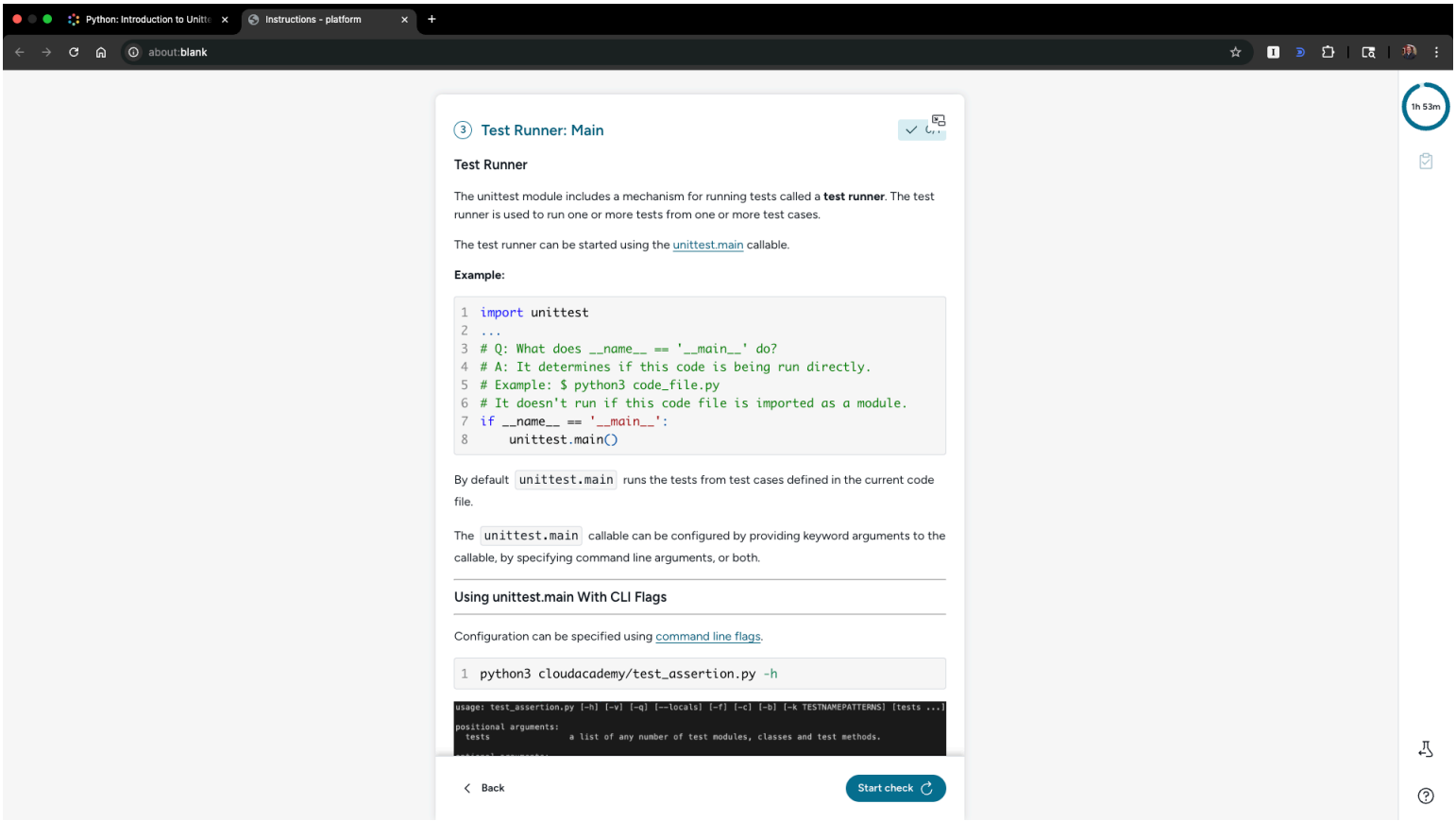
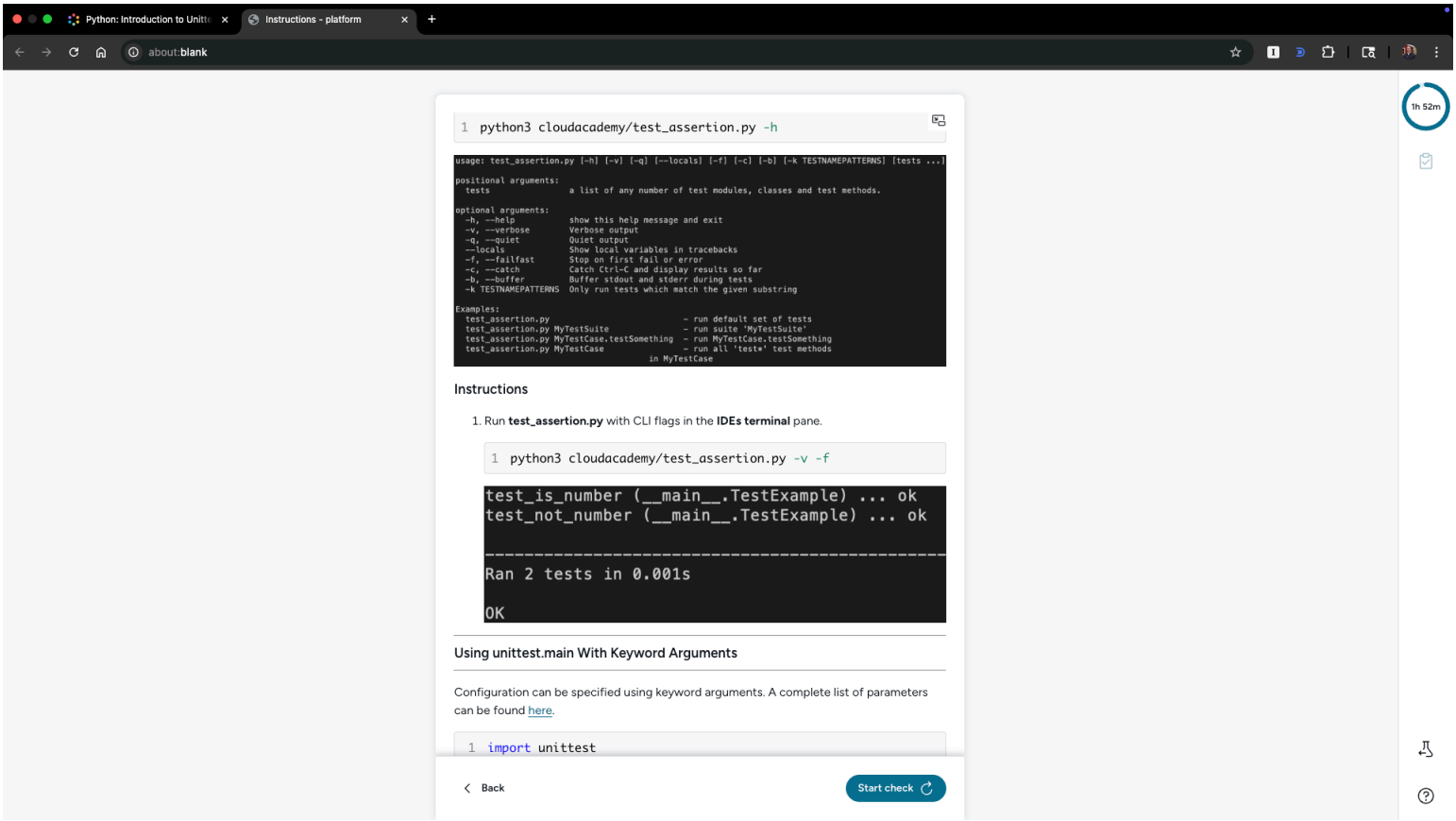
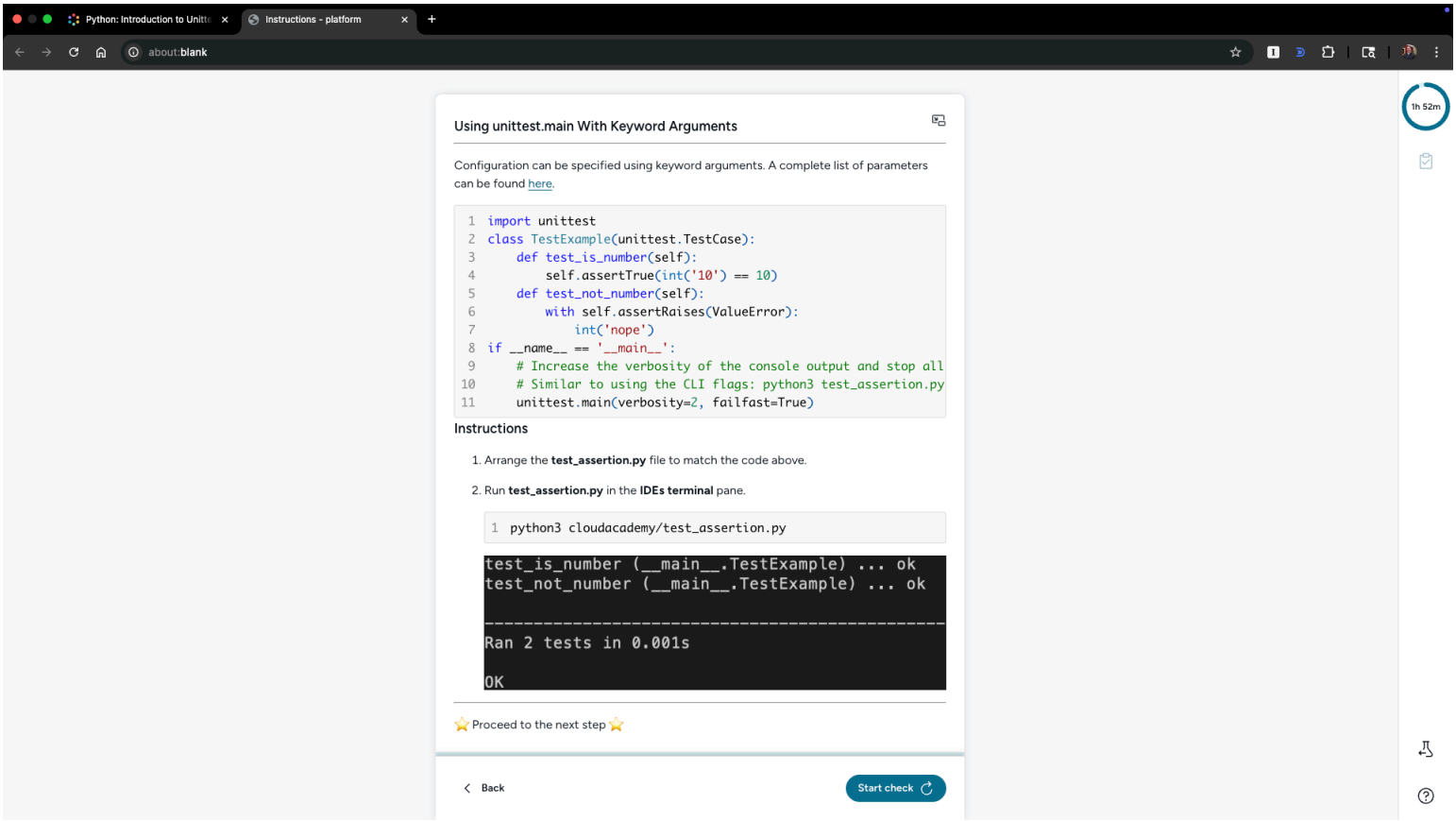

Python: Introduction to Unitte
Instructions - platform
about:blank
1h 53m
3 Test Runner: Main
Test Runner
The unittest module includes a mechanism for running tests called a test runner. The test runner is used to run one or more tests from one or more test cases.
The test runner can be started using the unittest.main callable.
Example:
1 import unittest
2 ...
3 # Q: What does __name__ == '__main__' do?
4 # A: It determines if this code is being run directly.
5 # Example: $ python3 code_file.py
6 # It doesn't run if this code file is imported as a module.
7 if __name__ == '__main__':
8 unittest.main()
By default unittest.main runs the tests from test cases defined in the current code file.
The unittest.main callable can be configured by providing keyword arguments to the callable, by specifying command line arguments, or both.
Using unittest.main With CLI Flags
Configuration can be specified using command line flags.
1 python3 cloudacademy/test_assertion.py -h
usage: test_assertion.py [-h] [-v] [-q] [--locals] [-f] [-c] [-b] [-k TESTNAMEPATTERNS] [tests ...]
positional arguments:
tests a list of any number of test modules, classes and test methods.
Back
Start check


Python: Introduction to Unitte
Instructions - platform
about:blank
1h 52m
1 python3 cloudacademy/test_assertion.py -h
usage: test_assertion.py [-h] [-v] [-q] [--locals] [-f] [-c] [-b] [-k TESTNAMEPATTERNS] [tests ...]
positional arguments:
tests a list of any number of test modules, classes and test methods.
optional arguments:
-h, --help show this help message and exit
-v, --verbose Verbose output
-q, --quiet Quiet output
--locals Show local variables in tracebacks
-f, --failfast Stop on first fail or error
-c, --catch Catch Ctrl-C and display results so far
-b, --buffer Buffer stdout and stderr during tests
-k TESTNAMEPATTERNS Only run tests which match the given substring
Examples:
test_assertion.py - run default set of tests
test_assertion.py MyTestSuite - run suite 'MyTestSuite'
test_assertion.py MyTestCase.testSomething - run MyTestCase.testSomething
test_assertion.py MyTestCase - run all 'test*' test methods
in MyTestCase
Instructions
1. Run test_assertion.py with CLI flags in the IDEs terminal pane.
1 python3 cloudacademy/test_assertion.py -v -f
test_is_number (__main__.TestExample) ... ok
test_not_number (__main__.TestExample) ... ok
Ran 2 tests in 0.001s
OK
Using unittest.main With Keyword Arguments
Configuration can be specified using keyword arguments. A complete list of parameters can be found here.
1 import unittest
Back
Start check


Python: Introduction to Unitte
Instructions - platform
about:blank
1h 52m
Using unittest.main With Keyword Arguments
Configuration can be specified using keyword arguments. A complete list of parameters can be found here.
import unittest
class TestExample(unittest.TestCase):
def test_is_number(self):
self.assertTrue(int('10') == 10)
def test_not_number(self):
with self.assertRaises(ValueError):
int('nope')
if __name__ == '__main__':
# Increase the verbosity of the console output and stop all
# Similar to using the CLI flags: python3 test_assertion.py
unittest.main(verbosity=2, failfast=True)
Instructions
1. Arrange the test_assertion.py file to match the code above.
2. Run test_assertion.py in the IDEs terminal pane.
python3 cloudacademy/test_assertion.py
test_is_number (__main__.TestExample) ... ok
test_not_number (__main__.TestExample) ... ok
Ran 2 tests in 0.001s
OK
Proceed to the next step
Back
Start check