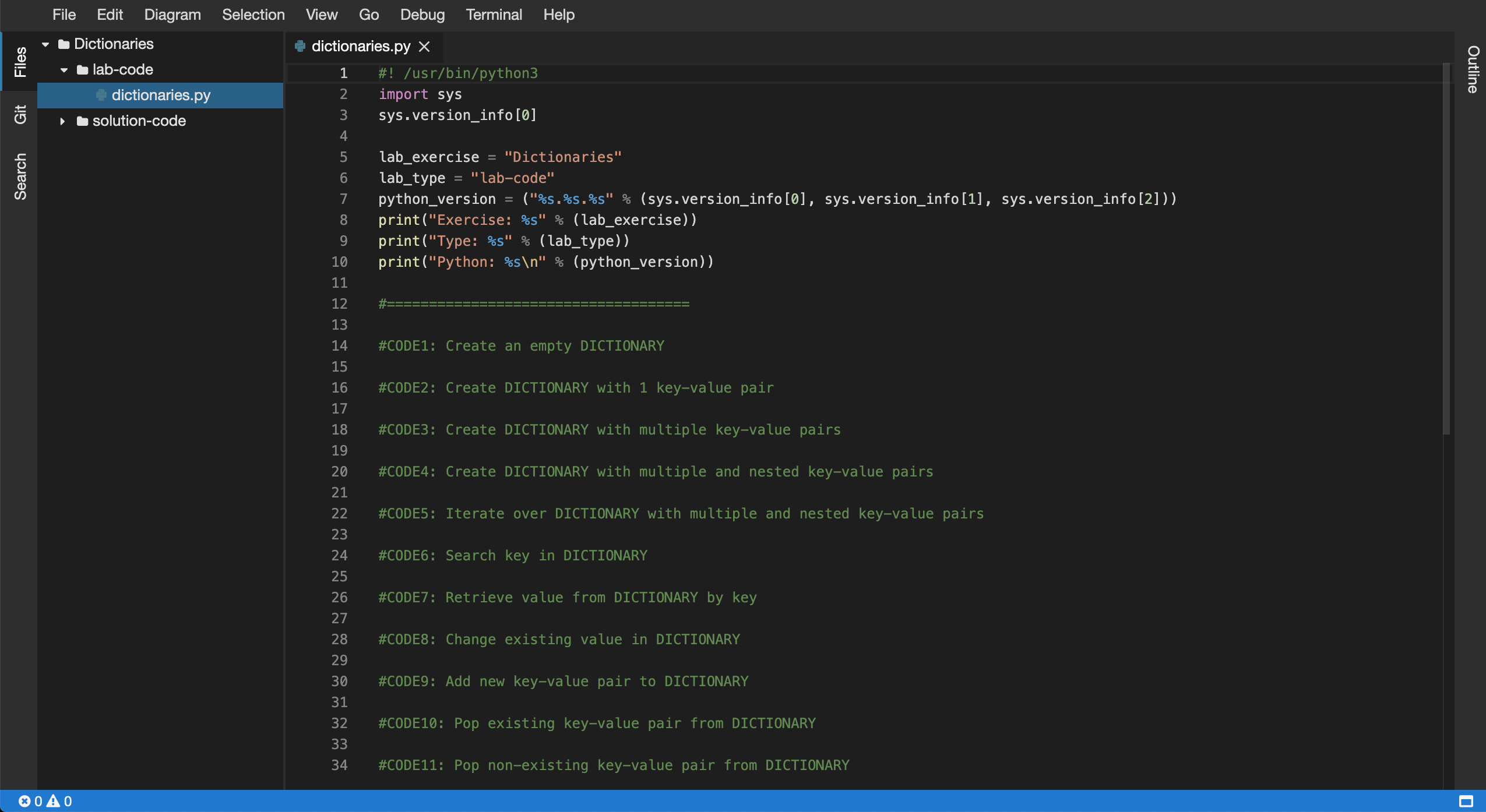
Dictionaries

Open the dictionaries.py source file within the editor. Take some time to review the uncompleted code within this file to understand the intended design:



5. Next, you will be required to complete the code in the following source files:

* dictionaries.py

6. Within the editor ensure that the dictionaries.py file now has focus.

7. Replace the //Code1: comment, create an empty DICTIONARY:

[**Copy code**](https://app.qa.com/lab/coding-python-collection-types/exercise-4-dictionaries/?context_id=637&context_resource=lp)

#CODE1: Create an empty DICTIONARY

dict1 = {}

print("CODE1:")

print(f"dict1 = {dict1}")

print(f"data type = {type(dict1)}")

print(f"length = {len(dict1)}")

print()

8. Replace the //Code2: comment, create a DICTIONARY with 1 key-value pair:

[**Copy code**](https://app.qa.com/lab/coding-python-collection-types/exercise-4-dictionaries/?context_id=637&context_resource=lp)

#CODE2: Create DICTIONARY with 1 key-value pair

dict2 = {"name": "cloudacademy"}

print("CODE2:")

print(f"dict2 = {dict2}")

print(f"data type = {type(dict2)}")

print(f"length = {len(dict2)}")

print()

9. Replace the //Code3: comment, create a DICTIONARY with multiple key-value pairs:

[**Copy code**](https://app.qa.com/lab/coding-python-collection-types/exercise-4-dictionaries/?context_id=637&context_resource=lp)

#CODE3: Create DICTIONARY with multiple key-value pairs

dict3 = {"name": "cloudacademy", "color": "red", "count": 1000}

print("CODE3:")

print(f"dict3 = {dict3}")

print(f"data type = {type(dict3)}")

print(f"length = {len(dict3)}")

print()

10. Replace the //Code4: comment, create a DICTIONARY with multiple and nested key-value pairs:

[**Copy code**](https://app.qa.com/lab/coding-python-collection-types/exercise-4-dictionaries/?context_id=637&context_resource=lp)

#CODE4: Create DICTIONARY with multiple and nested key-value pairs

dict4 = {"name": "cloudacademy", "color": "red", "count": 1000, "data": {"val1": 1, "val2": 2}}

print("CODE4:")

print(f"set4 = {dict4}")

print(f"data type = {type(dict4)}")

print(f"length = {len(dict4)}")

print()

11. Replace the //Code5: comment, iterate over a DICTIONARY with multiple and nested key-value pairs:

Note: Ensure the indentation within the following code block is maintained within the editor view when you perform the copy and paste operation.

[**Copy code**](https://app.qa.com/lab/coding-python-collection-types/exercise-4-dictionaries/?context_id=637&context_resource=lp)

#CODE5: Iterate over DICTIONARY with multiple and nested key-value pairs

print("CODE5:")

for key, value in dict4.items():

print(f"key={key}, value={value}")

print()

12. Replace the //Code6: comment, search for a key within a DICTIONARY:

[**Copy code**](https://app.qa.com/lab/coding-python-collection-types/exercise-4-dictionaries/?context_id=637&context_resource=lp)

#CODE6: Search key in DICTIONARY

print("CODE6:")

print ("name" in dict4)

print ("cloudacademy" in dict4)

print()

13. Replace the //Code7: comment, retrieve a value from a DICTIONARY by key:

[**Copy code**](https://app.qa.com/lab/coding-python-collection-types/exercise-4-dictionaries/?context_id=637&context_resource=lp)

#CODE7: Retrieve value from DICTIONARY by key

print("CODE7:")

item0 = dict4["name"]

item1 = dict4["color"]

print(f"item0 = {item0}")

print(f"item1 = {item1}")

print()

14. Replace the //Code8: comment, change an existing value in a DICTIONARY:

[**Copy code**](https://app.qa.com/lab/coding-python-collection-types/exercise-4-dictionaries/?context_id=637&context_resource=lp)

#CODE8: Change existing value in DICTIONARY

print("CODE8:")

dict4["name"] = "blah"

dict4["color"] = "blue"

print(f"dict4 = {dict4}")

print(f"data type = {type(dict4)}")

print(f"length = {len(dict4)}")

print()

15. Replace the //Code9: comment, add a new key-value pair to a DICTIONARY:

[**Copy code**](https://app.qa.com/lab/coding-python-collection-types/exercise-4-dictionaries/?context_id=637&context_resource=lp)

#CODE9: Add new key-value pair to DICTIONARY

print("CODE9:")

dict4['qwerty'] = 'fast'

print(f"dict4 = {dict4}")

print(f"data type = {type(dict4)}")

print(f"length = {len(dict4)}")

print()

16. Replace the //Code10: comment, pop an existing key-value pair from a DICTIONARY:

[**Copy code**](https://app.qa.com/lab/coding-python-collection-types/exercise-4-dictionaries/?context_id=637&context_resource=lp)

#CODE10: Pop existing key-value pair from DICTIONARY

print("CODE10:")

test = dict4.pop('qwerty', None)

print(f"test = {test}")

print(f"dict4 = {dict4}")

print(f"data type = {type(dict4)}")

print(f"length = {len(dict4)}")

print()

17. Replace the //Code11: comment, pop a non-existing key-value pair from a DICTIONARY:

[**Copy code**](https://app.qa.com/lab/coding-python-collection-types/exercise-4-dictionaries/?context_id=637&context_resource=lp)

#CODE11: Pop non-existing key-value pair from DICTIONARY

print("CODE11:")

test = dict4.pop('cat', None)

print(f"test = {test}")

print(f"dict4 = {dict4}")

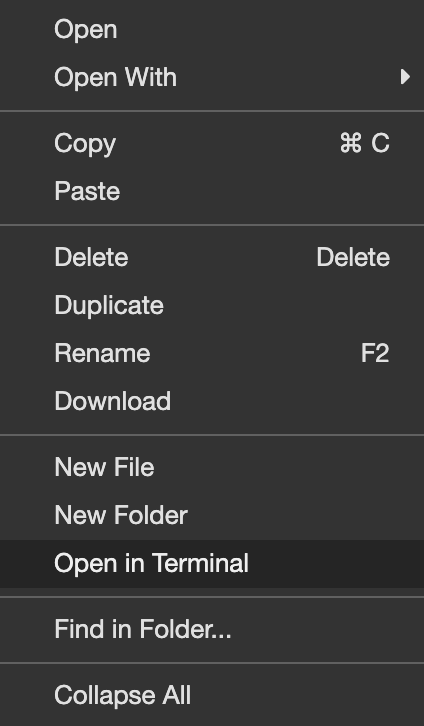
print(f"data type = {type(dict4)}")

print(f"length = {len(dict4)}")

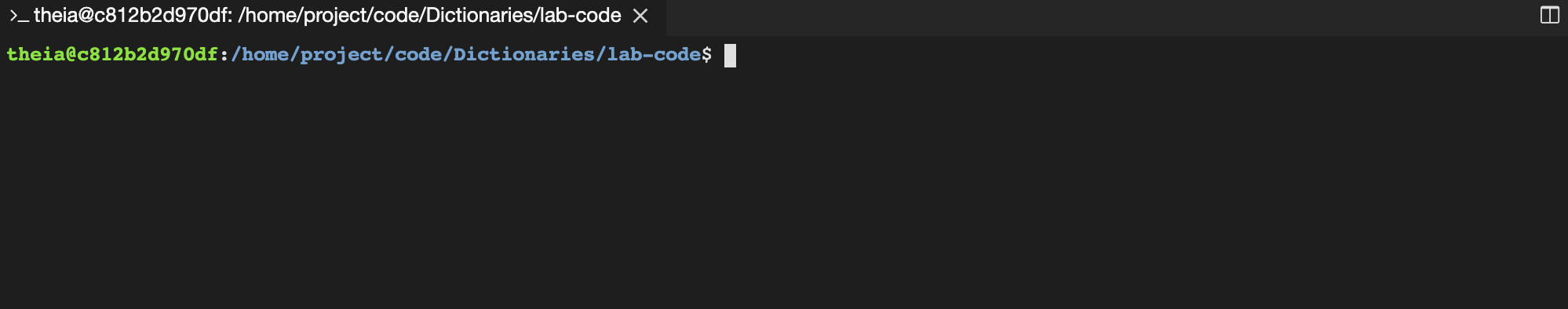
print()

18. Ok the dictionaries.py Python script is now ready to be executed by the Python interpreter. To do so you will need to use the embedded terminal to launch Python.

19. Within the Files tree view (lefthand side menu), select the lab-code folder and right click and select the Open in Terminal option:

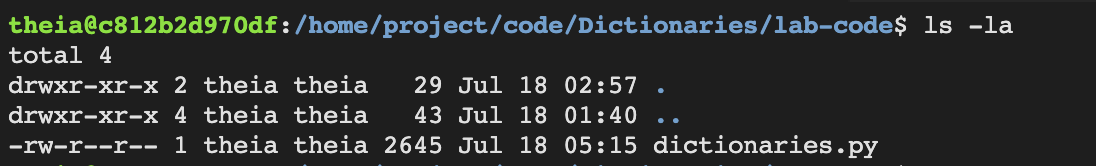


20. Access to the embedded terminal is now provided in the bottom pane of the current view:

  
21. Peform a directory listing on the current directory to ensure the presence of the dictionaries.py file, like so:

[**Copy code**](https://app.qa.com/lab/coding-python-collection-types/exercise-4-dictionaries/?context_id=637&context_resource=lp)

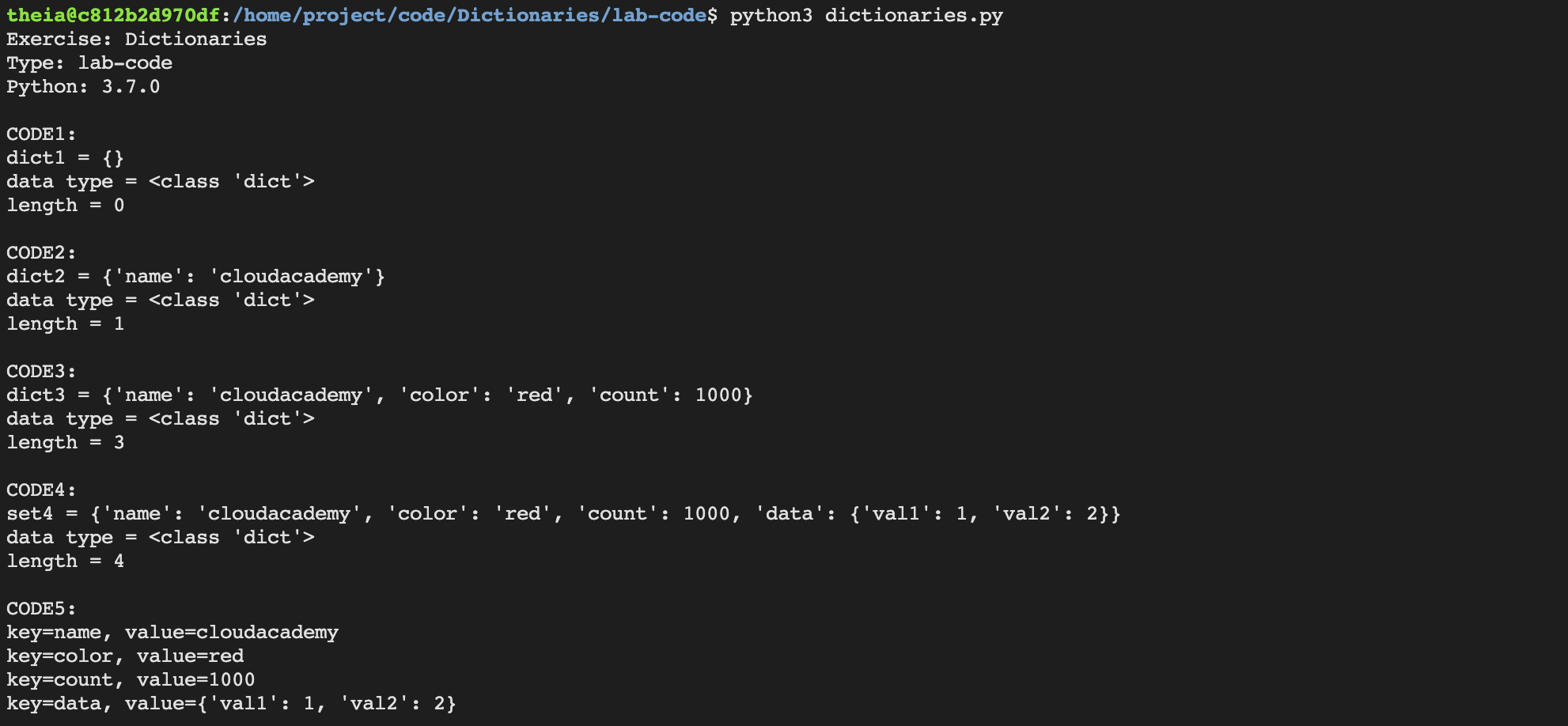
ls -la

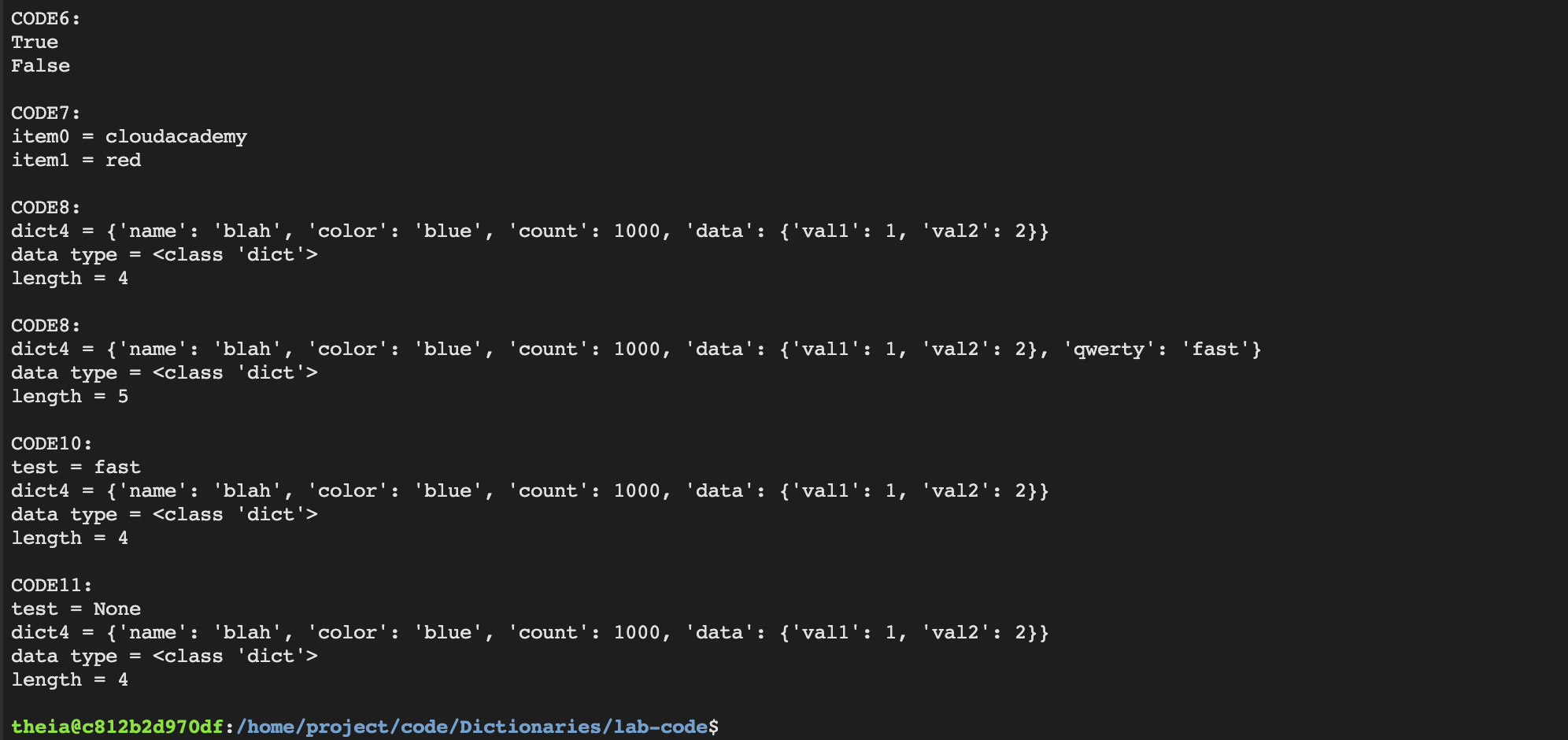


22. Let's now execute the updated dictionaries.py script using the python3 command. Within the terminal enter the following command:

[**Copy code**](https://app.qa.com/lab/coding-python-collection-types/exercise-4-dictionaries/?context_id=637&context_resource=lp)

python3 dictionaries.py





23. Great! Your dictionaries.py script has executed successfully! This demonstrates how to work with Dictionaries.

24. Try updating the dictionaries.py script and then re-executing using the same steps above. See the following documentation for ideas:

[https://docs.python.org/3.3/tutorial/datastructures.html#dictionaries](https://docs.python.org/3.3/tutorial/datastructures.html" \l "dictionaries)

### 

### Summary

In this Lab Step, you opened the **Dictionaries** Workspace and then updated the dictionaries.py file to store and manipulate data using Dictionaries. Next you used the embedded terminal to launch and debug the dictionaries.py script file.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Solution code

#! /usr/bin/python3

import sys

sys.version\_info[0]

lab\_exercise = "Dictionaries"

lab\_type = "solution-code"

python\_version = ("%s.%s.%s" % (sys.version\_info[0], sys.version\_info[1], sys.version\_info[2]))

print("Exercise: %s" % (lab\_exercise))

print("Type: %s" % (lab\_type))

print("Python: %s\n" % (python\_version))

#====================================

#CODE1: Create an empty DICTIONARY

dict1 = {}

print("CODE1:")

print(f"dict1 = {dict1}")

print(f"data type = {type(dict1)}")

print(f"length = {len(dict1)}")

print()

#CODE2: Create DICTIONARY with 1 key-value pair

dict2 = {"name": "cloudacademy"}

print("CODE2:")

print(f"dict2 = {dict2}")

print(f"data type = {type(dict2)}")

print(f"length = {len(dict2)}")

print()

#CODE3: Create DICTIONARY with multiple key-value pairs

dict3 = {"name": "cloudacademy", "color": "red", "count": 1000}

print("CODE3:")

print(f"dict3 = {dict3}")

print(f"data type = {type(dict3)}")

print(f"length = {len(dict3)}")

print()

#CODE4: Create DICTIONARY with multiple and nested key-value pairs

dict4 = {"name": "cloudacademy", "color": "red", "count": 1000, "data": {"val1": 1, "val2": 2}}

print("CODE4:")

print(f"set4 = {dict4}")

print(f"data type = {type(dict4)}")

print(f"length = {len(dict4)}")

print()

#CODE5: Iterate over DICTIONARY with multiple and nested key-value pairs

print("CODE5:")

for key, value in dict4.items():

print(f"key={key}, value={value}")

print()

#CODE6: Search key in DICTIONARY

print("CODE6:")

print ("name" in dict4)

print ("cloudacademy" in dict4)

print()

#CODE7: Retrieve value from DICTIONARY by key

print("CODE7:")

item0 = dict4["name"]

item1 = dict4["color"]

print(f"item0 = {item0}")

print(f"item1 = {item1}")

print()

#CODE8: Change existing value in DICTIONARY

print("CODE8:")

dict4["name"] = "blah"

dict4["color"] = "blue"

print(f"dict4 = {dict4}")

print(f"data type = {type(dict4)}")

print(f"length = {len(dict4)}")

print()

#CODE9: Add new key-value pair to DICTIONARY

print("CODE8:")

dict4['qwerty'] = 'fast'

print(f"dict4 = {dict4}")

print(f"data type = {type(dict4)}")

print(f"length = {len(dict4)}")

print()

#CODE10: Pop existing key-value pair from DICTIONARY

print("CODE10:")

test = dict4.pop('qwerty', None)

print(f"test = {test}")

print(f"dict4 = {dict4}")

print(f"data type = {type(dict4)}")

print(f"length = {len(dict4)}")

print()

#CODE11: Pop non-existing key-value pair from DICTIONARY

print("CODE11:")

test = dict4.pop('cat', None)

print(f"test = {test}")

print(f"dict4 = {dict4}")

print(f"data type = {type(dict4)}")

print(f"length = {len(dict4)}")

print()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

output console

Exercise: Dictionaries

Type: solution-code

Python: 3.7.0

CODE1:

dict1 = {}

data type = <class 'dict'>

length = 0

CODE2:

dict2 = {'name': 'cloudacademy'}

data type = <class 'dict'>

length = 1

CODE3:

dict3 = {'name': 'cloudacademy', 'color': 'red', 'count': 1000}

data type = <class 'dict'>

length = 3

CODE4:

set4 = {'name': 'cloudacademy', 'color': 'red', 'count': 1000, 'data': {'val1': 1, 'val2': 2}}

data type = <class 'dict'>

length = 4

CODE5:

key=name, value=cloudacademy

key=color, value=red

key=count, value=1000

key=data, value={'val1': 1, 'val2': 2}

CODE6:

True

False

CODE7:

item0 = cloudacademy

item1 = red

CODE8:

dict4 = {'name': 'blah', 'color': 'blue', 'count': 1000, 'data': {'val1': 1, 'val2': 2}}

data type = <class 'dict'>

length = 4

CODE8:

dict4 = {'name': 'blah', 'color': 'blue', 'count': 1000, 'data': {'val1': 1, 'val2': 2}, 'qwerty': 'fast'}

data type = <class 'dict'>

length = 5

CODE10:

test = fast

dict4 = {'name': 'blah', 'color': 'blue', 'count': 1000, 'data': {'val1': 1, 'val2': 2}}

data type = <class 'dict'>

length = 4

CODE11:

test = None

dict4 = {'name': 'blah', 'color': 'blue', 'count': 1000, 'data': {'val1': 1, 'val2': 2}}

data type = <class 'dict'>

length = 4

theia@production-session-93142-6dd54db79c-z8dtq:/home/project/code/Dictionaries/solution-code$