List, Tuple, Dictionary

Lab Overview

In Python, string and numeric data types are often used in groups called collections. Three such collections supported by Python are the list, the tuple, and the dictionary.

In this lab, you will:

* Use the list data type
* Use the tuple data type
* Use the dictionary data type

Estimated Completion Time

45 minutes

Exercise 1: The List Data Type

Define a List

You will edit a Python script to hold a collection of fruit names, or a list.

1. Click on **work** folder in the file list:
2. Click on the file **list\_tuple\_dictionary.py** which has been created for you:
3. Type the following:
4. myFruitList= ["apple", "banana", "cherry"]
5. print(myFruitList)

print(type(myFruitList))

1. Save the file.
2. To run the program, type the following at the terminal:

python3.6 list\_tuple\_dictionary.py

1. Verify that the script runs correctly and that the output displays as you expect it to.

Access a List by Position

The contents of a list can also be accessed by position. In this activity, you will print out each item in our list by their position:

1. In programming languages, the list position starts at zero. The brackets tell Python you are about to tell it the position in the list you want. To access the "apple" string, type the following:

print(myFruitList[0])

1. To access the "banana" string, type the following:

print(myFruitList[1])

1. To access the "cherry" string, type the following:

print(myFruitList[2])

1. Save the file.
2. To run the program, type the following at the terminal:

python3.6 list\_tuple\_dictionary.py

1. Verify that the script runs correctly and that the output displays as you expect it to.

Change the Values in a List

The values of a list can be changed. In this activity, you will change "cherry" to "orange."

1. In Python, list position starts at zero, so you have to use the number 2 to access the third position. Type the following:

myFruitList[2] = "orange"

1. Print the updated list:

print(myFruitList)

1. Save the file.
2. To run the program, type the following at the terminal:

python3.6 list\_tuple\_dictionary.py

1. Verify that the script runs correctly and that the output displays as you expect it to.
2. ~ $ python3.6 list\_tuple\_dictionary.py
3. ['apple', 'banana', 'cherry']
4. <class 'list'>
5. apple
6. banana
7. cherry
8. ['apple', 'banana', 'orange']

~ $

Exercise 2: The Tuple Data Type

Define a Tuple

The tuple is like a list, but it cannot be changed. A data type that cannot be changed after creation is called immutable. To define a tuple, you use parentheses instead of brackets. Let’s continue the Python script:

1. Type the following to create a tuple:
2. myFinalAnswerTuple = ("apple", "banana", "pineapple")
3. print(myFinalAnswerTuple)

print(type(myFinalAnswerTuple))

1. Save the file.
2. To run the program, type the following at the terminal:

python3.6 list\_tuple\_dictionary.py

1. Verify that the script runs correctly and that the output displays as you expect it to.

Access a Tuple by Position

Like a list, the items of a tuple can also be accessed by position:

1. To access the "apple" string, type the following:

print(myFinalAnswerTuple[0])

1. To access the "banana" string, type the following:

print(myFinalAnswerTuple[1])

1. To access the "pineapple" string, type the following:

print(myFinalAnswerTuple[2])

1. Save the file.
2. To run the program, type the following at the terminal:

python3.6 list\_tuple\_dictionary.py

1. Verify that the script runs correctly and that the output displays as you expect it to.
2. ~ $ python3.6 list\_tuple\_dictionary.py
3. ['apple', 'banana', 'cherry']
4. <class 'list'>
5. apple
6. banana
7. cherry
8. ['apple', 'banana', 'orange']
9. ('apple', 'banana', 'pineapple')
10. <class 'tuple'>
11. apple
12. banana
13. pineapple

~ $

Exercise 3: The Dictionary Data Type

Define a Dictionary

A dictionary is a list with named positions (keys). Imagine your list shows people’s favorite fruit.

1. Return to the Python script, and type the following:
2. myFavoriteFruitDictionary = {
3. "Adam" : "apple",
4. "Ben" : "banana",
5. "Penny" : "pineapple"

}

1. Use the **print()** function to write the dictionary to the shell:

print(myFavoriteFruitDictionary)

1. Use the **type()** function to write the data type to the shell:

print(type(myFavoriteFruitDictionary))

1. Save the file.
2. To run the program, type the following at the terminal:

python3.6 list\_tuple\_dictionary.py

Access a Dictionary by Name

Instead of using numbers, you will use the name of the individuals to get their favorite fruit.

1. To access Adam’s favorite fruit, type the following:

print(myFavoriteFruitDictionary["Adam"])

1. To access Ben’s favorite fruit, type the following:

print(myFavoriteFruitDictionary["Ben"])

1. To access Penny’s favorite fruit, type the following:

print(myFavoriteFruitDictionary["Penny"])

1. Save the file.
2. To run the program, type the following at the terminal:

python3.6 list\_tuple\_dictionary.py

1. Verify that the script runs correctly and that the output displays as you expect it to.
2. ~ $ python3.6 list\_tuple\_dictionary.py
3. ['apple', 'banana', 'cherry']
4. <class 'list'>
5. apple
6. banana
7. cherry
8. ['apple', 'banana', 'orange']
9. ('apple', 'banana', 'pineapple')
10. <class 'tuple'>
11. apple
12. banana
13. pineapple
14. {'Adam': 'apple', 'Ben': 'banana', 'Penny': 'pineapple'}
15. <class 'dict'>
16. apple
17. banana
18. pineapple

~ $

Congratulations! You can now use the list, tuple, and dictionary data types.

STOP

You have successfully completed this lab.