Lista, tupla, diccionario

Información general sobre el laboratorio

En Python, los tipos de datos numéricos y los Strings suelen utilizarse en grupos denominados “colecciones”. Python proporciona colecciones como la lista (list), la tupla (tuple) y el diccionario (dictionary).

En este laboratorio, realizará lo siguiente:

* Utilizar el tipo de dato “lista” (list)
* Utilizar el tipo de datos de “tupla” (tuple)
* Utilizar el tipo de dato de “diccionario” (dictionary)

Tiempo estimado para finalizar

45 minutos

Ejercicio 1: Tipo de dato “lista”

Definición de una lista

Editará un script en Python para mantener una colección de nombres de frutas, almacenada en una lista.

1. Haga clic en la carpeta **work** en la lista de archivos:
2. Haga clic en el archivo **list\_tuple\_dictionary.py** que se creó para usted:
3. Escriba lo siguiente:
4. myFruitList= ["apple", "banana", "cherry"]
5. print(myFruitList)

print(type(myFruitList))

1. Guarde el archivo.
2. Para ejecutar el programa, escriba lo siguiente en la terminal:

python3.6 list\_tuple\_dictionary.py

1. Verifique que el script se ejecuta correctamente y que el resultado es el esperado.

Acceso a una lista por posición

También se puede acceder al contenido de una lista por posición. En esta actividad, mostrará cada elemento de nuestra lista por su posición:

1. En la mayoría de lenguajes de programación, la posición en una lista comienza en el cero. Los corchetes indican a Python que está indicando la posición deseada en la lista. Para leer el primer string “manzana”, escriba lo siguiente:

print(myFruitList[0])

1. Para leer el segundo string “plátano”, escriba lo siguiente:

print(myFruitList[1])

1. Para leer el string “cereza”, escriba lo siguiente:

print(myFruitList[2])

1. Guarde el archivo.
2. Para ejecutar el programa, escriba lo siguiente en la terminal:

python3.6 list\_tuple\_dictionary.py

1. Verifique que el script se ejecuta correctamente y que el resultado es el esperado.

Cambiar los valores en una lista

Los valores de una lista se pueden cambiar. En esta actividad, cambiará “cereza” por “naranja”.

1. En Python, la posición en la lista comienza por cero, por lo que tiene que utilizar el número 2 para acceder a la tercera posición. Escriba lo siguiente:

myFruitList[2] = "orange"

1. Muestre la lista actualizada:

print(myFruitList)

1. Guarde el archivo.
2. Para ejecutar el programa, escriba lo siguiente en la terminal:

python3.6 list\_tuple\_dictionary.py

1. Verifique que el script se ejecuta correctamente y que el resultado es el esperado.
2. ~ $ python3.6 list\_tuple\_dictionary.py
3. ['apple', 'banana', 'cherry']
4. <class 'list'>
5. apple
6. banana
7. cherry
8. ['apple', 'banana', 'orange']

~ $

Ejercicio 2: Tipo de dato “tupla”

Definir una tupla

Una tupla es similar a una lista, pero no se puede modificar. Un tipo de dato que no se puede cambiar después de su creación se denomina inmutable. Para definir una tupla, se utilizan paréntesis en lugar de corchetes. Continuemos con el script en Python:

1. Escriba lo siguiente para crear una tupla:
2. myFinalAnswerTuple = ("apple", "banana", "pineapple")
3. print(myFinalAnswerTuple)

print(type(myFinalAnswerTuple))

1. Guarde el archivo.
2. Para ejecutar el programa, escriba lo siguiente en la terminal:

python3.6 list\_tuple\_dictionary.py

1. Verifique que el script se ejecuta correctamente y que el resultado es el esperado.

Acceso a una tupla por posición

Al igual que con una lista, también se puede acceder a los elementos de una tupla por su posición:

1. Para leer el primer string “manzana”, escriba lo siguiente:

print(myFinalAnswerTuple[0])

1. Para leer el segundo string “plátano”, escriba lo siguiente:

print(myFinalAnswerTuple[1])

1. Para leer el string “piña”, escriba lo siguiente:

print(myFinalAnswerTuple[2])

1. Guarde el archivo.
2. Para ejecutar el programa, escriba lo siguiente en la terminal:

python3.6 list\_tuple\_dictionary.py

1. Verifique que el script se ejecuta correctamente y que el resultado es el esperado.
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

~ $

Ejercicio 3: tipo de dato “diccionario”

Definición de un diccionario

Un diccionario es una lista cuyas posiciones tienen nombres asignados (claves). Imagine que su lista muestra la fruta favorita de distintas personas.

1. Vuelva al script en Python y escriba lo siguiente:
2. myFavoriteFruitDictionary = {
3. "Adam" : "apple",
4. "Ben" : "banana",
5. "Penny" : "pineapple"

}

1. Utilice la función **print()**para mostrar el diccionario en el shell:

print(myFavoriteFruitDictionary)

1. Utilice la función **type()** para mostrar el tipo de dato en el shell:

print(type(myFavoriteFruitDictionary))

1. Guarde el archivo.
2. Para ejecutar el programa, escriba lo siguiente en la terminal:

python3.6 list\_tuple\_dictionary.py

Acceso al diccionario por nombre

En lugar de utilizar números, utilizará el nombre de las personas para conocer su fruta favorita.

1. Para leer la fruta favorita de Adam, escriba lo siguiente:

print(myFavoriteFruitDictionary["Adam"])

1. Para leer la fruta favorita de Ben, escriba lo siguiente:

print(myFavoriteFruitDictionary["Ben"])

1. Para leer la fruta favorita de Penny, escriba lo siguiente:

print(myFavoriteFruitDictionary["Penny"])

1. Guarde el archivo.
2. Para ejecutar el programa, escriba lo siguiente en la terminal:

python3.6 list\_tuple\_dictionary.py

1. Verifique que el script se ejecuta correctamente y que el resultado es el esperado.
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

~ $

¡Felicitaciones! Ahora puede utilizar tipos de datos como lista, tupla y diccionario.

Fin

Ha completado correctamente el laboratorio.