Preguntas teóricas del Full Stack ChekPoint 4

¿Cuál es la diferencia entre una lista y una tupla en Python?

Una lista es una estructura de datos que permite almacenar colecciones de datos tanto del mismo tipo como de tipos distintos. Las listas se definen usando paréntesis planos con los miembros de la colección separados por comas. Por ejemplo, la lista siguiente:

mixed_type_list=[“string”, 1, True]

contiene tres miembros, una cadena de caracteres, un valor numérico y una valor booleano. El primer elemento de la colección, en este caso la cadena de caracteres, se encuentra en la posición 0, a la cual se puede acceder usando el nombre de la lista seguido de dos paréntesis planos con el índice 0: “mixed_type_list[0]”. El segundo elemento se encuentra en la posición 1, y así sucesivamente.

Las tuplas se definen de manera parecida, pero usando paréntesis, como el ejemplo siguiente, que contiene los mismos datos de la lista ejemplo:

mixed_type_tupla=(“string”, 1, True)

La principal diferencia entre estas dos estructuras de datos es que las tuplas son inmutables, lo que significa que una vez creadas no pueden modificar-se. Aunque existen algunos trucos que permiten modificar su contenido, añadir o borrar elementos de una tupla, las buenas prácticas de programación aconsejan no modificarlas. Su modificación debe restringirse a casos donde no tengamos control sobre la creación de dicha tupla pero tengamos que usarla.

¿Cuál es el orden de las operaciones?

Primero se obtienen los valores entre paréntesis, seguidos de los exponentes. Después de los exponentes siguen las operaciones de multiplicación y de división. Las últimas operaciones en realizar-se son las adiciones y las substracciones.

¿Qué es un diccionario Python?

Un diccionario Python es una estructura de datos que permite almacenar todo tipo de datos identificados por una clave, de la misma manera que en un diccionario enciclopédico buscamos palabras para obtener información (definición de la palabra). Las claves de entrada del diccionario se escriben entre comillas y van seguidas por “:”, a la cual sigue la información correspondiente a la entrada. Cada par de valores clave-entrada está separado por una “,”. El código siguiente muestra un ejemplo de diccionario Python con dos claves, cada una almacenando una cadena de caracteres:

```
my_dict={
  “key1”: “cat”,
  “key2”: “dog”
}
```

Para acceder a la información correspondiente a una entrada del diccionario se debe escribir el nombre de la variable vdiccionario, seguida de la clave que se quiere consultar entre corchetes. Siguiendo con el ejemplo anterior, para obtener la información, cadena de

caracteres en este caso, que corresponde a la clave “key1”, debemos proceder de la siguiente manera: my_dict [“key1”].

¿Cuál es la diferencia entre el método ordenado y la función de ordenación?

Las listas en Python disponen del método “sort” que permite ordenar de manera ascendente o descendente una lista numérica o alfabéticamente si se trata de cadenas de caracteres. Por ejemplo, si tenemos la lista siguiente y la queremos ordenar alfabéticamente, podemos usar el código siguiente:

my_list = [“ab”, “bc”,”cd”]

my_list.sort().

Este método ordena la lista, pero no retorna ningún valor, por lo que si en vez de ordenar la lista, lo que queremos es retornar la lista ordenada para almacenarla en otra, debemos usar la función “sorted”. Esta utiliza una lista como argumento y retorna la lista ordenada. Por ejemplo, el código siguiente crea una nueva lista como la anterior pero ordenada alfabéticamente:

sorted_list = sorted(my_list)

¿Qué es un operador de reasignación?

El operador de asignación coincide con el operador asignación “=”. No obstante, cuando la variable ya existe, y por tanto se le ha asignado un valor literal previamente, si esta es modificada, se dice que se esta reasignando.

La reasignación puede consistir en la modificación del valor por otro literal o su mutación, por ejemplo, sumando una constante al valor existente. En estos casos especiales donde el valor existente se muta, se usan variantes del operador asignación precedidas de el operador que marca la modificación del valor existente previo. Por ejemplo, el código siguiente:

X = 10

X = 22

X +=1

Realiza una asignación inicial del número 10 a la variable X. Seguidamente, la variable X es reasignada al valor 22. Finalmente, la variable es reasignada partiendo de su valor actual (22) al que se le añade 1.