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

La diferencia entre una lista y una tupla es que las tuplas son inmutables, mientras que en las listas se puede modificar elementos. Para las listas se usan corchetes [] y para las tuplas paréntesis () para almacenar varios elementos. (Ejemplo: list_example = [‘b’, ‘d’, a’, ‘c’] tuple_example = (‘b’, ‘d’, ‘a’, ‘c’). Si usamos sort function: list_example.sort() y tuple_example.sort() en caso de la lista el orden de los elementos se cambiará y print(list_example) nos dará [‘a’, ‘b’, ‘c’, ‘d’] y en caso de la tupla nos dará error.)

¿Cuál es el orden de las operaciones?

El orden de las operaciones es siguiente: P(parentesis) E(exponents) M(multiplication) D(division) A(addition) S(subtraction). Se lo podemos memorizar con la frase “Please Excuse My Dear Aunt Sally”. (Ejemplo: calculation = 6 + 2 ** 3 – (10 / 2 + 1) * 2. Primero realizamos las operaciones dentro de los paréntesis con el orden correcto 10/2 = 5 y sumamos 1: 5 + 1 = 6. Después realizamos la exponenciación: 2 ** 3 = 8. Luego realizamos la multiplicación del resultado en los paréntesis por 2: 6 * 2 = 12. Y finalmente realizamos suma y resta de izquierda a derecha 6 + 8 – 12 = 2. Por lo tanto, la respuesta es 2.)

¿Qué es un diccionario Python?

Un diccionario – es un tipo de collecciones en Python, permite almacenar pares clave-valor y luego pasar la clave para acceder al valor. Por ejemplo: employees = {‘admin’: ‘Kate’, ‘secretary’: ‘Maria’, ‘consultant’: ‘Jacob’}. Para acceder al valor ‘Kate’ y guardarlo en una variable utilizamos la clave: first_employe = eemployees[‘admin’]. También se puede tener colecciones anidadas; obtener elementos, por ejemplo, print(list(employees.values())[1]); añadir nuevos pares clave-valor (ejemplo: employees[‘manager’] = ‘Peter’); o borrar elementos (ejemplo: del employees[‘admin’]).

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

La diferencia entre método sort y función sorted es que la función sorted nos pemite guardar el valor en una variable y luego usar y llamar la colección original y cambiada cuando sea necesario. Por ejemplo: tenemos una lista de edades: age = [5, 15, 10, 1] y si usamos el método sort: age.sort(), nos dará como resultado una lista en orden ascendente [1, 5, 10, 15]. Y si usamos la función sorted: sorted_list = sorted(age), nos dará lo mismo. Pero con la función sorted podemos guardar esa lista ordenada en la variable sorted_list y seguir teniendo acceso a la lista original age en caso de que la necesitemos en otro momento, mientras que con el método sort la lista original se modifica permanentemente.

¿Qué es un operador de asignación?

El operador de asignación es el signo igual (=), que se utiliza para asignar un valor. Otros operadores de asignación son += (suma en asignación), -= (resta en asignación), *= (multiplicación en asignación), /= (división en asignación), //= (división entera en asignación), **= (exponenta en asignación) y %= (módulo en asignación). Estos operadores realizan una operación matemática y luego asignan el resultado a la variable. Por ejemplo: si tenemos una variable_example = 5 y queremos añadir 10, podemos escribir código (variable_example += 10) y este hace lo mismo que (variable_example = variable_example + 10).