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

Las listas son mutables y las tuplas son inmutables. En las primeras, si usas una función o cualquier acción que la modifique, ese cambio se mantendrá. Las segundas, sin embargo, no se pueden modificar. Para mantener un cambio en una tupla, se tiene que crear una nueva, por ejemplo, almacenándola en una nueva variable.

```
Ejemplo de la diferencia entre lista y tupla en código Python:
lista = [“El perro está cansado”, “La comida es saludable”, “El carro está roto”]
tupla = (“El perro está cansado”, “La comida es saludable”, “El carro está roto”)
```

¿Cuál es el orden de las operaciones?

El orden de operaciones en Python es el orden utilizado en álgebra al resolver una expresión matemática.

Orden de operaciones:

Paréntesis	**( )**	Ejemplo
Exponente	**	`58 – (3*5) + 32 + ((4*3)/3)`
Multiplicación	*	`58 – (3*5) + 32 + (12/3)`
Division	/	`58 – 15 + 32 + 4 = 79`
Suma	+	
Resta	-	

¿Qué es un diccionario Python?

Un tipo de datos dentro de las colecciones, en el que se pueden agrupar los datos en grupos de llave-valor. Es decir, una serie de dos elementos, en el que uno es la llave a la cual va asociada un valor. La llave puede ser cualquier tipo de dato que sea inmutable (*booleans*, cadenas, tuplas, cadenas) pero no acepta los mutables. El valor puede ser cualquier tipo de dato, incluidas otros diccionarios. Este último caso se conoce como *nested dictionary*. A diferencia de las listas, en los diccionarios no se puede trabajar con índices. Para acceder a los valores se emplean las llaves.

La forma de escribir los diccionarios es entre {}, las llaves se separan de los valores por : y los conjuntos llave-valor se separan por , . Los conjuntos llave-valor deben estar indentados hacia adentro respecto al elemento de la izquierda (nombre del diccionario). En cuanto al separador entre llave y valor, se debe dejar un espacio entre este y el valor. Dejar un espacio entre la llave y los dos puntos dependerá del programador, ambas prácticas están aceptadas. A continuación, ejemplos en los que se pueden utilizar diccionarios.

Diccionario simple:

```
music_groups = {
  "london_grammar": ["Hannah Reid, “Dot Major", “Dan
                                                  Rothman”],
  "delaporte": ["Sandra Delaporte", “Sergio Salvi"],
  "myrkur": [“Amalie Bruun”, “others”]
}
```

Nested dictionary:

```
music_groups = [
  {
    'london_grammar': {
      'cantante': 'Hannah Reid',
      ‘teclado y bateria': 'Dot Major',
      'guitarra': 'Dan Rothman',
    }
  },
  {
    'delaporte': {
      'cantante': 'Sandra Delaporte',
      'mesa_sonido': 'Sergio Salvi',
    }
  },
]
```

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

Método de ordenación: sort() Cambia el orden de la lista y realmente cambia la lista original, moviendo los elementos dentro de la misma. No se puede guardar en una variable, si se intenta, devuelve None. Este método es específico de las listas.

Función de ordenación sorted() devuelve una lista nueva con el orden de elementos cambiado, pero sin alterar la original. Esto permite poder hacer operaciones con ella, como guardarla dentro de una variable.

Cuándo usar cada uno dependerá de la intención que se quiera dar. Se usa sort() si se quiere cambiar una lista y se sabe qué elementos tiene y no se necesita utilizar esa lista sin cambiar más. En el caso de sorted() es conveniente usarlo cuando se necesita cambiar el orden de la lista y hacer operaciones con este nuevo orden, pero se necesita también mantener la lista original intenta.

¿Qué es un operador de asignación?

Un símbolo que hace operaciones le da un valor al operando de la izquierda. En el caso de los operadores de asignación con dos signos, al valor de la izquierda de se hace la operación con el valor de la derecha. Por ejemplo:

```
precio = 58
```

`precio += 10` → esto devolvería 10 sumado (+) al valor previo de 'precio', es decir, 68.

Los operadores de asignación más comunes se muestran en la siguiente tabla. Cabe mencionar que los resultados de las operaciones se guardan en el operando de la izquierda.

Operador de asignación	=	Asigna el valor de la expresión de la dcha al de la izq.
Operador de suma	+=	Suma el operando de la izq y el de la dcha.
Operador de resta	-=	Resta el operando de la dcha al de la izq.
Operador de multiplicación	*=	Multiplica los operandos de la izq.
Operador de división	/=	Divide el operando de la izq entre el de la dcha.
Operador módulo	%=	Calcula el módulo del operando de la izq dividido por el de la dcha.
Operador exponencial	**=	Eleva el operando de la izq al operando de la dch.
Operador de división entera a la baja	//=	Divide el valor de la izq por el de la dcha y redondea el resultado a la baja en un número entero. Importante: solo devuelve números enteros.