

M2C4 Python Assignment

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

Las listas se definen con [] y las tuplas con (). Otra diferencia es que las listas se pueden modificar, es decir, se puede añadir, quitar o cambiar elementos. Esto no es posible hacerlo a una tupla, una vez creada, es inmutable.

Si se quiere hacer cualquier cambio en una tupla primero habría que convertirla en una lista y después volverla a convertir en tupla. Si queremos por ejemplo añadir un elemento al final:

```
cool_tuple = ("Maele", "Lune", "Sciel")
provisional_list = list(cool_tuple)
provisional_list.append("Verso")
new_tuple = tuple(provisional_list)
```

Si quisiéramos ordenarla también habría que convertirla a una lista primero y después de nuevo a tupla.

¿Cuál es el orden de las operaciones?

El orden es:

1. Paréntesis
2. Exponentes
3. Multiplicaciones
4. Divisiones
5. Suma
6. Resta

¿Qué es un diccionario Python?

En un diccionario se almacenan pares de clave y su correspondiente valor. Dichas claves han de ser únicas, y los valores pueden ser de cualquier tipo, números, strings, listas, otros diccionarios... Se definen con {}. Y son modificables.

Por ejemplo:

```
cool_dictionary = {
    "name": "Jon",
    "age": 33,
    "likes": "Gaming"
}
```

Las claves son name, age y likes. Todas únicas. Y se ve que cada una tiene un valor asignado, uno de los cuales es un numero mientras que los otros son strings.

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

El método ordenado, .sort(), solo se puede utilizar con listas y modifica directamente la lista original, sin crear una nueva. Por ejemplo:

```
cool_list = ["Verso", "Maele", "Lune", "Sciel"]  
cool_list.sort()
```

Asi estaria ordenada, si lo imprimiéramos con print(cool_list) obtendríamos la lista ordenada alfabéticamente.

La función de ordenación, sorted(), funciona con listas, tuplas, strings... sin modificarlas en absoluto, devolviendo siempre una lista ordenada, la cual podremos guardar en otra variable. Es otra forma de ordenar la tupla del primer ejemplo:

```
cool_tuple = ("Maele", "Lune", "Sciel", "Verso")  
sorted_list = sorted (cool_tuple)
```

Esto es una lista, que es lo que devuelve la función independientemente de que le hayamos pasado una tupla. Si queremos, podemos convertirla en tupla de nuevo.

```
new_cool_tuple = tuple(sorted_list)
```

¿Qué es un operador de reasignación?

Se usan para hacer cálculos a la vez que se asignan valores a una variable. Por ejemplo:

Aquí se muestra el valor inicial de la variable total.

```
total = 100
```

Si queremos incrementar su valor en 10 podríamos hacer:

```
total = total + 10
```

Esto funciona, pero para que sea mas compacto usaríamos un operador de asignación, en este caso += para incrementar su valor en 10, siendo el valor de la variable total, 110.

```
total += 10
```

Estos son los demás operadores de asignación que podemos hacer:

```
total -= 10
```

```
total *= 2
```

```
total /= 10
```

```
total //= 10
```

```
total **= 2
```

```
total % = 2
```