

M2C4 TEORÍA

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

La principal diferencia entre una lista y una tupla es que una lista se puede modificar y una tupla no. La primera es mutable se puede transformar, es decir ampliar, reducir y también ordenar y la segunda no al menos no como la lista.

Ambas tanto listas como tuplas son conjuntos o colecciones de datos pueden estar formados por cadenas, integar, floats.

Otra diferencia es referente a su sintaxis las listas usa corchetes y las tuplas paréntesis.

Ejemplos:

Lista

```
directors = ['lang', 'villeneuve', 'hitchcock', 'donen', 'nolan']
```

Tuple

```
actors = ('gable', 'grant', 'power', 'firth', 'holden')
```

Las listas pueden utilizar métodos para ser ordendas como son `sort()` y `.sorted()` y para ser ampliadas `append()` `extend()` , para quitar elementos `remove()` `pop()`. Las tuplas no pueden ser modificadas ni ordenadas son inmutables. (se puede utilizar la función `.sorted()` pero sigue mantiene la tuple original).

Las utilización de uno u otro de tipo de colección o agrupación de datos dependenerá de la finalidad del trabajo a realizar.

2. ¿Cuál es el orden de las operaciones?

El orden de las operaciones siguen el orden del acronimo **PEDMAS** ó también se puede usar tip nemotécnico de **Please, excuse my dear aunt Sally**. Es decir, que el orden sería

1. Paréntesis
2. Exponentes
3. Multiplicaciones
4. División
5. adición (suma)
6. sustracción (restas)

Ejemplos

```
jfl= 15+16*(25+36)+26-43**3 /25  
print (jfl)  
  
-2163,28
```

3. ¿Qué es un diccionario Python?

Es una colección, un conjunto de datos que están agrupados por parejas es decir, clave o Key y valor.

La sintaxis correspondiente a ambos es clave y valor están entre comillas y separados por dos puntos. El final y el principio del diccionario , es decir se abre y se cierra con llaves {}

Ejemplo:

```
dictionary = { "table" : "mesa" , "chair": "silla" , "Lamp" : "lampara" , "door" : "puerta" }
```

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

La diferencia entre .sort() y .sorted().

El primero crea una nueva lista ordenada puede ser en orden creciente o decreciente al igual que alfabéticamente que puede ser ascendente o descendente ; esto puede ocasionar que sea más lento y consume más memoria; en cambio .sorted () crea una nueva lista o colección de datos es decir puede ser utilizado en listas, tuplas, diccionarios. y conserva el conjunto o colección original

Ejemplos:

```
directors = ['lang', 'scott', 'hitchcock', 'donen']  
directors.sort()  
print(directors) ['donen', 'hitchcock', 'lang', 'scott']
```

```
actors = ('gable', 'pacino', 'bardem', 'isaac')  
actors_sorted= sorted(actors)  
print(actors_sorted) ['bardem', 'gable', 'isaac', 'pacino']
```

5. ¿Qué es un operador de reasignación?

El operador de asignación = nos permite directamente una serie de operaciones como sumar o añadir un elemento "+=" ó realizar una operación aritmética este es el caso de "*=", "/=", "%="

Ejemplos

```
x=8
```

```
y+=1
```

```
print(x) 9
```

```
y=10
```

```
y-=1
```

```
print(y) 9
```

```
z=30
```

```
z/=3
```

```
print (z) 10,0
```