Tuplas en Python

Tuplas

¿Qué son las tuplas en Python?

Las tuplas son parecidas a las listas pero con ligeras diferencias y utilizadas para otros propósitos.

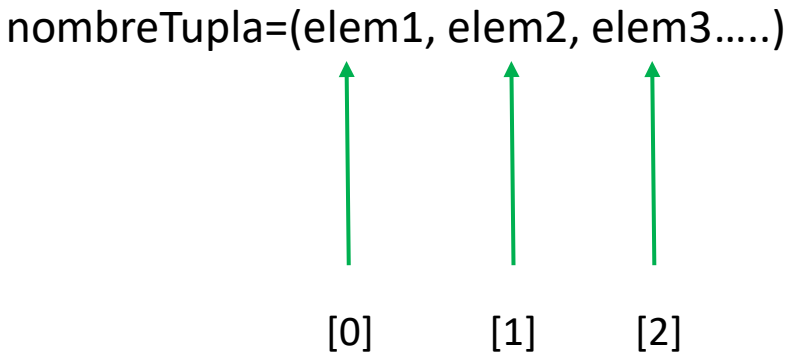
Ambas son un conjunto ordenado de valores, pero las listas presentan una serie de funciones adicionales que permiten manipular de muy diversas maneras los valores que contienen. Podemos decir entonces que las listas son dinámicas o mutables, mientras que las tuplas son estáticas o inmutables.

¿Por qué utilizar entonces una tupla si las listas parecen ser más polivalentes? Fundamentalmente por optimización de recursos. Las tuplas solo necesitan un bloque de la memoria para almacenarse mientras que las listas necesitan dos. Y además como consecuencia de esto, las tuplas son mucho más rápidas de manejar. Esto se nota en la ejecución de un programa cuando vamos a almacenar un gran volumen de datos en ambas estructuras.

¿Cuándo utilizar una lista y cuándo una tupla? Si no vas a manipular la información almacenada entonces siempre será mejor la tupla. Si vas a manipular la información, bien añadiendo o eliminado o moviendo o actualizando la información almacenada, entonces será mejor una lista.

También las utilizarás cuando vayas a formatear Strings, y cuando necesites utilizarlas como claves de diccionarios (los diccionarios los veremos en la próxima entrega de documentación del curso).

Sintaxis de las tuplas



Los elementos almacenados en una lista tienen todos una posición o índice representado entre []. El índice comienza siempre en [0]. Es frecuente para el principiante confundir índice

con nº de elemento así que cuidado con este detalle (sobre todo cuando abordemos en el futuro las estructuras de control de flujo).

Ejemplos de tuplas:

```
misDatos=("Juan", 13, 1, 2002)
```

Es frecuente a la hora de programar en Python que surja la necesidad de convertir una tupla en lista. De esta forma podemos manipular agregando, eliminando etc elementos de la tupla (aunque en realidad no lo hacemos sobre la tupla sino sobre la lista resultante de convertir tupla a lista). ¿Cómo se hace? Se declara una variable y se utiliza el método **list()** al que se le pasa la tupla como parámetro. Se muestra en la siguiente imagen:

```
misDatos=("Juan", 13, 1, 2002)

misDatosLista=list(misDatos)
```

Por supuesto el proceso inverso también es posible, es decir, el paso de lista a tupla:

```
misDatos=["Juan", 13, 1, 2002]

misDatosTupla=tuple(misDatos)
```

Operaciones frecuentes en tuplas:

(los siguientes ejemplos hay que fijarse bien que las operaciones se hacen sobre la tupla misDatosTupla que es el resultado de convertir la lista misDatos a tupla):

Ver si un elemento se encuentra en una tupla

```
misDatos=["Juan", 13, 1, 2002]

misDatosTupla=tuple(misDatos)

print("Juan" in misDatosTupla)
```

Imprimirá en consola **True** si el elemento se encuentra y **False** si no se encuentra.

Cuántas veces se encuentra un elemento en una tupla:

```
misDatos=["Juan", 13, 1, 2002]

misDatosTupla=tuple(misDatos)

print(misDatosTupla.count("Juan"))
```

En el programa de la imagen se imprimirá en consola 1, porque 1 son las veces que aparece "Juan" en la tupla.

Obtener la longitud de una tupla:

```
misDatos=["Juan", 13, 1, 2002, "Juan"]

misDatosTupla=tuple(misDatos)

print(len(misDatosTupla))
```

Desempaquetar una tupla:

Consiste en almacenar en variables de una forma rápida los valores almacenados en la tupla. Se muestra en la siguiente imagen:

```
misDatos=("Juan", 13, 1, 2002,)

nombre, dia, mes, agno=misDatos
```

En la imagen anterior, los valores de la tupla se almacenan en las variables en el mismo orden de almacenamiento (nombre="Juan", día=13 etc)