

1. TUPLAS <CLASS 'TUPLE'>.

¿**ALGUNAS CARACTERÍSTICAS?**

- Las tuplas: (**tuple**) [Tuples / tupla](#) : <class 'tuple'>.
- + Es un **dato compuesto** de 0 a N valores, contenido en cualquier tipo de objeto ([literal](#), [variables](#), [expresiones](#), etc.).
- + Los datos que contienen:
 - Forman secuencias **Inmutables** (no pueden ser “modificados” una vez creados).
 - Están **indexados**.
 - Pueden ser de distintos tipos ([numero](#), [cadena](#), [tupla](#), [lista](#), etc.).
 - . Se permiten también la **anidación** de tuplas.
 - Pueden estar repetidos.

1.1. CREACIÓN.

Se pueden crear datos de tipo tupla <class 'tuple'>, utilizando:

- Formato **literal**, entre paréntesis () , una lista de valores, separados entre comas.

```
[var_tupla=] [ ( ] valor1 [ , valorN ] . . . [ ) ]
```

Nota: Para las tuplas, podemos incluso obviar los paréntesis:

- Función constructora: [tuple\(\)](#).

tuple (x) ----- La función [tuple\(\)](#). Devuelve un tipo de dato tupla.

+**Nota:** hay que utilizar los dobles corchetes

```
[var_tupla=] tuple ( ( valor1 [ , valorN ] . . . [ ) )
```

1.2. OPERACIONES:

- De “Secuencia”:

- + Como dato **compuesto**, se puede conocer el número de elementos o longitud de la tupla, con la función incorporada [len\(\)](#).
- + Al **estar indexados**(subíndices), se puede hacer referencia a un elemento por su posición, o a una parte (rebanada) de la tupla, utilizando los **distintos formatos de los corchetes**.
- + Al ser **inmutables**, no puede modificarse su contenido, el nombre de los objetos y el valor de los literales, pero si el contenido de elementos que sean variable de tipo mutable.

- **Otras**, según sus métodos.

1.3. MÉTODOS.

Las tuplas tienen **dos** métodos:

Ver propiedades/métodos de las **tuplas**.

count (x) ----- El método [count\(\)](#). Devuelve el número de veces que se repite el valor en la tupla.

```
[var_num=] tuple.count(valor)
```