**TUPLES**

Es una lista INMUTABLE (guarda elementos y siempre queda exactamente igual, es decir que no la puedes modificar como quitar o añadir elementos). Es por ello que tiene menos methods disponibles

→ en vez de tener [] se definen con **()**

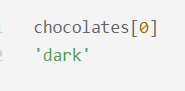
****

→ también pueden tener DICS dentro:

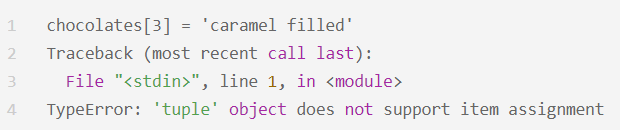
Ej my\_tuple = **(1, “1”, True, [“dog”,”cat”])**

*my\_tuple = [“1”,1,(“dog,”cat”)] esto sería una lista, no un tuple pq está definida con ()*

→ **Podemos recuperar cualquier elemento** de la tupla de la misma forma que lo haríamos con una cadena. Sin embargo, no podemos reasignar nuevos valores a una tupla.

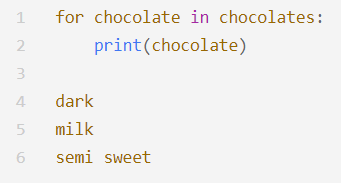


→ Sin embargo, **NO podemos reasignar nuevos valores** a una tupla.



\* Si quieres cambiar una tupla la tendrías que ***convertir antes a un list***

→ **Podemos iterar** a través de tuplas al igual que iteramos a través de listas. Sin embargo, como se mencionó anteriormente, **no podemos cambiar los valores** de la tupla (SOLO “PRINTEAR”)

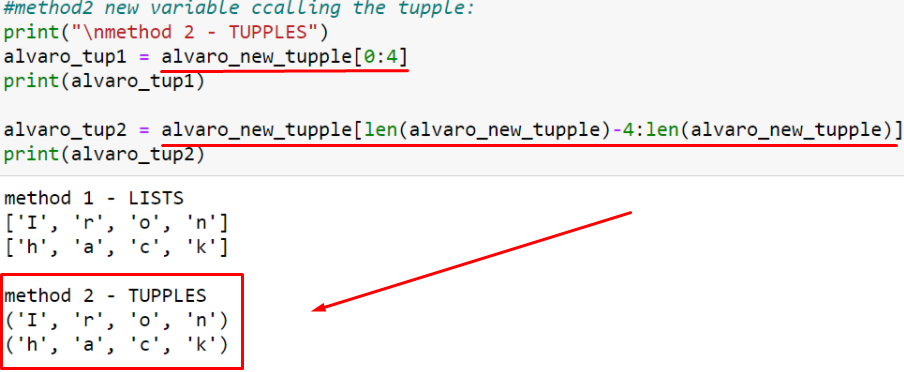


\* Solo podríamos CREAR un nuevo Tuple:

* a MANO:

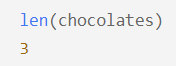
****

* LLAMANDO a otro tuple vía INDEX (ejemplo al final)



*LOS SIGUIENTES MÉTODOS SON COMPARTIDOS DE UNA LISTA Y UN TUPLE*

**len(“x”)**: en un string te dice Nº caracteres, en una lista o tuple te dice Nº elementos

****

**.count(“x”)** → te dice cuantos elementos son iguales dentro de una lista o tuple

my\_tuple.count(1)

= 2 (pq existe el 1 y el True, que realmente es un 1)

my\_tuple.count([“dog”,”cat”])

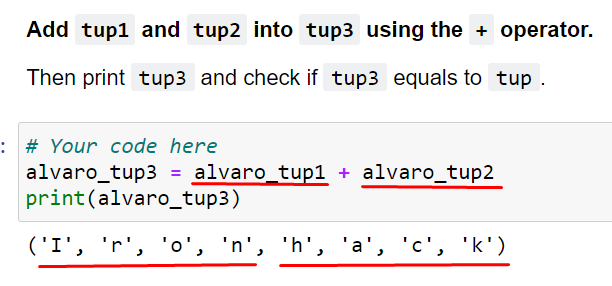
=1

**.index(“x”)** → te dice DÓNDE está ese valor de la tupla.

my\_tuple.index(3)

=[“dog”,”cat”]

**sumar diferentes tupples**

****