

Práctica Tuples 1

Utiliza un método de tuplas para contar la cantidad de veces que aparece el valor `2` en la siguiente tupla, y muestra el resultado (integer) en pantalla:

```
mi_tupla = (1, 2, 3, 2, 3, 1, 3, 2, 3, 3, 3, 1, 3, 2, 2, 1, 3, 2)
```

Práctica Tuples 2

Convierte a **lista** la siguiente **tupla**, y almacénala en una variable llamada `mi_lista`.

```
mi_tupla = (1, 2, 3, 2, 3, 1, 3, 2)
```

Práctica Tuples 3

Extrae los elementos de la siguiente tupla en cuatro variables: `a`, `b`, `c`, `d`

```
mi_tupla = (1, 2, 3, 4)
```

Práctica Sets 1

Une los siguientes sets en uno solo, llamado `mi_set_3`:

```
{1, 2, "tres", "cuatro"}
```

```
{"tres", 4, 5}
```

Práctica Sets 2

Elimina un elemento al azar del siguiente set, utilizando métodos de sets.

```
1 | sorteo = {"Camila", "Margarita", "Axel", "Jorge", "Miguel", "Mónica"}
```

Práctica Sets 3

Agrega el nombre **Damián** al siguiente set, utilizando *métodos de sets*:

```
sorteo = {"Camila", "Margarita", "Axel", "Jorge", "Miguel", "Mónica"}
```

Práctica Booleanos 1

Realiza una comparación que arroje como resultado un **booleano** y almacena el resultado (True/False) en una variable llamada `prueba`

Práctica Booleanos 2

Verifica si `17834/34` es mayor que `87*56` y muestra el resultado (**booleano**) en pantalla utilizando `print()`

Práctica Booleanos 3

Verifica si la raíz cuadrada de 25 es igual a 5 y muestra el resultado (booleano) en pantalla utilizando `print()`