

Tuplas

Son unas colecciones muy parecidas a las listas con la peculiaridad de que son inmutables:

```
tupla = (100, "Hola", [1, 2, 3], -50)
tupla
```

```
(100, 'Hola', [1, 2, 3], -50)
```

Indexación y slicing

```
print(tupla)
print(tupla[0])
print(tupla[-1])
print(tupla[2:])
print(tupla[2][-1])
```

```
(100, 'Hola', [1, 2, 3], -50)
100
-50
([1, 2, 3, 4], -50)
4
```

Inmutabilidad

```
tupla[0] = 50
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-9-b45433b4cee9> in <module>()
----> 1 tupla[0] = 50
TypeError: 'tuple' object does not support item assignment
```

Función len()

Igual que si fuera una lista podemos utilizarla para saber la longitud de una tupla:

```
len(tupla)
```

```
4
```

```
len(tupla[2])
```

```
3
```

Métodos integrados

index()

Sirve para buscar un elemento y saber su posición en la tupla:

```
tupla.index(100)
```

```
0
```

```
tupla.index('Hola')
```

```
1
```

Da error si no se encuentra:

```
tupla.index('Otro')
```

```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-18-640d616163a2> in <module>()  
----> 1 tupla.index('Otro')  
ValueError: tuple.index(x): x not in tuple
```

count()

Sirve para contar cuantas veces aparece un elemento en una tupla:

```
tupla.count(100)
```

```
1
```

```
tupla.count('Algo')
```

```
0
```

```
tupla = (100, 100, 100, 50, 10)
tupla.count(100)
```

```
3
```

append() ?

Al ser inmutables, las tuplas **no disponen** de métodos para modificar su contenido:

```
tupla.append(10)
```

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-23-758d195ec9d7> in <module>()
----> 1 tupla.append(10)
AttributeError: 'tuple' object has no attribute 'append'
```