

3. Estructuras de datos

Índice

- 1. Listas
- 2. Tuplas
- 3. Diccionarios
- 4. Conjuntos

1. Listas

Una lista (**list**) es una colección de elementos que pueden ser de distintos tipos:

```
colors = ["red", "blue", "green"]
stuff = ["thing", 3, True]
```

Se accede por su índice numérico (empieza en cero, acaba en menos uno):

```
print(colors[0])
print(colors[-1])
```

Se pueden añadir y eliminar elementos:

```
colors.remove("blue")
colors.append("orange")
print(colors)
```

Se pueden recorrer:

```
for color in colors:
    print(color)
```

2. Tuplas

Una tupla (**tuple**) es también una colección de elementos que pueden ser de distintos tipos:

```
position = (2, 3, -1)
```

Las tuplas no se pueden modificar parcialmente, pero sí totalmente:

```
position[0] = 6 # Error
position.remove(2) # Error
```

3. Diccionarios

Esto son ejemplos de diccionarios (**dict**):

```
dictionary = {1:"X", "X":2}
godness = {"name": 'Unicornio', "lastname": 'Rosa Invisible', 'age': 31}
```

Se accede por su clave:

```
print(dictionary[1])
print(godness["name"])
```

Se pueden añadir y eliminar elementos:

```
godness["place"] = "rainbow"
print(godness)

godness.pop("place")
print(godness)

# Otra opción para borrar:
del godness["place"]
```

Se pueden recorrer:

```
# Claves:

for key in godness:
    print(key)

# Otra forma:

for key in godness.keys():
    print(key)

# Valores:

for value in godness:
    print(godness[value])

# Otra forma:
```

```
for value in godness.values():  
    print(value)  
  
# Claves y valores:  
  
for key, value in godness.items():  
    print(key, value)
```

4. Conjuntos

Este es un ejemplo de conjunto (**set**):

```
colorBalls = {"red", "blue", "green"}
```

Como son estructuras de datos no indexadas, no es posible acceder directamente a sus elementos, pero sí se pueden añadir:

```
colorBalls.add("black")
```

Se puede comprobar si existe (o no) un elemento para, por ejemplo, borrarlo:

```
if "blue" in colorBalls:  
    colorBalls.remove("blue")
```

Se pueden recorrer:

```
for ball in colorBalls:  
    print(ball)
```

Se pueden unir dos conjuntos para crear conjuntos nuevos:

```
colorBalls = {"red", "blue", "green"}  
numberBalls = {1, 2, 3}  
  
totalBalls = colorBalls.union(numberBalls)
```

O buscar la intersección entre dos conjuntos para ver sus elementos comunes:

```
balls1 = {"red", 2, "green", 4}
balls2 = {1, 2, "green", 3}

commonBalls = balls1.intersection(balls2)
```

Resumen comparativo

	Ordenado	Mutable	Indexado	Duplicados
Lista	Sí	Sí	Sí	Sí
Tupla	Sí	No	Sí	Sí
Diccionario	Sí	Sí	Sí	Claves no, valores sí
Conjunto	No	No	No	No

Referencias

- Listas
- Tuplas
- Diccionarios
- Conjuntos