

Clase2_2_Conjuntos

March 24, 2023

1 Seminario de Lenguajes - Python

1.1 Conjuntos en Python

2 Conjuntos en Python

- Un conjunto es una colección de datos heterogénea, **desordenada**, **NO indexada** y **sin elementos duplicados**.

```
[18]: bandas = {"AC/DC", "Metallica", "Greta Van Fleet", "Soda Stéreo", "Los Piojos"}
      type(bandas)
```

```
[18]: set
```

```
[19]: bandas_nacionales = set(("Soda Stéreo", "La Renga"))
      bandas_nacionales
```

```
[19]: {'La Renga', 'Soda Stéreo'}
```

```
[20]: bandas_internacionales = set(["Greta Van Fleet", "Led Zeppelin"])
      bandas_internacionales
```

```
[20]: {'Greta Van Fleet', 'Led Zeppelin'}
```

```
[21]: letras = set("alabanza")
      letras
```

```
[21]: {'a', 'b', 'l', 'n', 'z'}
```

3 No confundir

```
[22]: dicci = {}
      conjunto = set()

      type(dicci)
```

```
[22]: dict
```

```
[23]: c1 = {'alabanza'}
      c2 = set('alabanza')
      print(c1)
      print(c2)
```

```
{'alabanza'}
{'a', 'b', 'l', 'z', 'n'}
```

4 ¿Cómo recorremos un conjunto?

```
[24]: bandas = {"AC/DC", "Metallica", "Greta Van Fleet", "Soda Stéreo", "Los Piojos"}

      for elem in bandas:
          print(elem)
```

```
AC/DC
Los Piojos
Greta Van Fleet
Soda Stéreo
Metallica
```

5 Y, ¿cómo sabemos la cantidad de elementos de un conjunto?

```
[25]: len(bandas)
```

```
[25]: 5
```

```
[26]: c2 = set('alabanzas')
      len(c2)
```

```
[26]: 6
```

6 Operaciones con conjuntos

- Pensemos en las operaciones matemáticas sobre conjuntos:
 - **in**: retonar si un elemento pertenece o no a un conjunto.
 - **|**: unión entre dos conjuntos.
 - **&**: intersección entre dos conjuntos.
 - **-**: diferencia de conjuntos.

```
[27]: bandas = {"AC/DC", "Metallica", "Greta Van Fleet", "Soda Stéreo", "Los Piojos"}
      bandas_nacionales = set(("Soda Stéreo", "La Renga", "Los Piojos"))
```

```
[29]: print("Los Piojos" in bandas)
```

```
True
```

```
[30]: todos = bandas | bandas_nacionales
print(f"Todas las bandas: {todos}")

en_ambos = bandas & bandas_nacionales
print(f"En ambos conjuntos: {en_ambos}")

internacionales = bandas - bandas_nacionales
print(f"No son nacionales: {internacionales}")
```

Todas las bandas: {'AC/DC', 'Los Piojos', 'Greta Van Fleet', 'Soda Stéreo', 'La Renga', 'Metallica'}

En ambos conjuntos: {'Los Piojos', 'Soda Stéreo'}

No son nacionales: {'AC/DC', 'Greta Van Fleet', 'Metallica'}

7 También hay métodos que se pueden utilizar

- Para copiar conjuntos podemos usar la sentencia de asignación o usando `copy()`.

```
[32]: bandas = {"AC/DC", "Metallica", "Greta Van Fleet", "Soda Stéreo", "Los Piojos"}
bandas_nacionales = set(("Soda Stéreo", "La Renga", "Los Piojos"))

#mis_bandas = bandas
mis_bandas = bandas.copy()
print(id(mis_bandas))
print(id(bandas))
```

140099003817920

140099003819936

- Se puede adicionar un elemento con `add()`.

```
[34]: bandas.add("Foo Fighters")
print(bandas)
```

{'AC/DC', 'Los Piojos', 'Greta Van Fleet', 'Soda Stéreo', 'Foo Fighters', 'Metallica'}

8 Probar en casa otros métodos

- `issubset()`, `isdisjoint()`, `issuperset()`, `update()`, `discard()`, `remove()`
- + [Info en Real Python](#)
- + [Info en w3schools](#)
- + [Info en el sitio oficial](#)