

Conjuntos

Son colecciones desordenadas de elementos únicos utilizados para hacer pruebas de pertenencia a grupos y eliminación de elementos duplicados.

Para definir un conjunto vacío hay que llamar a su clase set (conjunto en inglés):

```
conjunto = set()  
conjunto
```

```
set()
```

Sin embargo si lo creamos con algunos datos se definen entre llaves:

```
conjunto = {1,2,3}  
conjunto
```

```
{1, 2, 3}
```

Método add()

Sirve para añadir elementos al conjunto, pero si un elemento ya se encuentra no se añadirá de nuevo:

```
conjunto.add(4)  
conjunto
```

```
{1, 2, 3, 4}
```

```
conjunto.add(0)  
conjunto
```

```
{0, 1, 2, 3, 4}
```

Colecciones desordenadas

Se dice que son desordenados porque gestionan automáticamente la posición de sus elementos, en lugar de conservarlos en la posición que nosotros los añadimos:

```
conjunto.add('H')
conjunto.add('A')
conjunto.add('Z')
conjunto
```

```
{0, 1, 2, 3, 4, 'A', 'Z', 'H'}
```

Pertenencia a grupos

```
grupo = {'Hector', 'Juan', 'Mario'}
```

Es fácil saber si un elemento se encuentra en un conjunto utilizando la sintaxis **in**. Se utiliza mucho para trabajar con grupos:

```
'Hector' in grupo
```

```
True
```

También se puede hacer la comprobación inversa con **not in**:

```
'Hector' not in grupo
```

```
False
```

Elementos únicos

Los conjuntos no pueden tener el mismo elemento más de una vez, se borran los duplicados automáticamente:

```
test = {'Hector', 'Hector', 'Hector'}
test
```

```
{'Hector'}
```

Conversiones con listas

Es muy útil transformar listas a conjuntos para borrar los elementos duplicados automáticamente y viceversa:

```
lista = [1,2,3,3,2,1]

print(lista)

conjunto = set(lista)
lista = list(conjunto)

print(lista)
```

```
[1, 2, 3, 3, 2, 1]
[1, 2, 3]
```

La conversión se puede hacer en una línea:

```
lista = [1,2,3,3,2,1]
print(lista)

lista = list(set(lista))
print(lista)
```

```
[1, 2, 3, 3, 2, 1]
[1, 2, 3]
```

Conversiones con cadenas

Hacer esta transformación sirve para crear un conjunto con todos los caracteres de la cadena, pero sin duplicados:

```
cadena = "Al pan pan y al vino vino"
set(cadena)
```

```
{' ', 'A', 'a', 'i', 'l', 'n', 'o', 'p', 'v', 'y'}
```