

# 😊 Conjuntos (set)

Un **conjunto** es una colección de objetos únicos. Los conjuntos son ampliamente utilizados en **lógica y matemática**.

La **teoría de conjuntos** es una parte de las matemáticas que tiene un objeto de estudio propio; con métodos propios, con ciertas relaciones con otras teorías matemáticas, en particular, con todas las teorías matemáticas tradicionales y a partir de sus principios se mantiene la existencia, estructura y relaciones mutuas entre ellos.

Ver: [Teoría de conjuntos \(Noción y determinación de conjuntos\)](https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:nocion-y-determinacion-de-conjuntos/a/141-nocion-y-determinacion-de-conjuntos) (<https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:nocion-y-determinacion-de-conjuntos/a/141-nocion-y-determinacion-de-conjuntos>)

En **Python** un conjunto es una estructura mutable. Los conjuntos no pueden tener elementos duplicados.

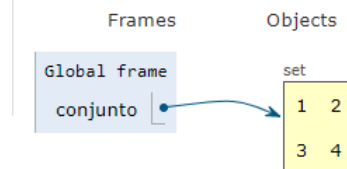
## Creación de un conjunto en Python

Para crear un conjunto especificamos sus elementos entre llaves.

```
In [1]: conjunto = {1,2,3,4}
conjunto
```

```
Out[1]: {1, 2, 3, 4}
```

```
1 conjunto = {1,2,3,4}
→ 2 conjunto
```



```
In [2]: type(conjunto)
```

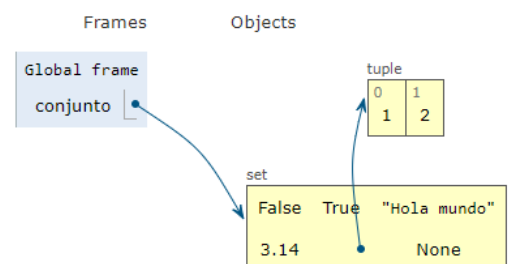
```
Out[2]: set
```

Al igual que otras colecciones, sus miembros pueden ser de diversos tipos.

```
In [3]: conjunto = {True, 3.14, None, False, "Hola mundo", (1, 2)}
conjunto
```

```
Out[3]: {(1, 2), 3.14, False, 'Hola mundo', None, True}
```

```
1 conjunto = {True, 3.14, None, False, "Hola mundo", (1, 2)}
→ 2 conjunto
```



No obstante, un conjunto no puede incluir objetos mutables como listas, diccionarios, e incluso otros conjuntos.

```
In [4]: conjunto = {[1,2,3]}
```

```
-----
TypeError                                Traceback (most recent call last)
Input In [4], in <cell line: 1>()
----> 1 conjunto = {[1,2,3]}

TypeError: unhashable type: 'list'
```

**Python no puede dirimir el siguiente caso.** Por defecto, la asignación siguiente crea un diccionario.

```
In [5]: conjunto = {}
conjunto
```

```
Out[5]: {}
```

```
In [6]: type(conjunto)
```

```
Out[6]: dict
```

## set

Para generar un conjunto vacío, directamente creamos una instancia de la clase **set**:

```
In [7]: conjunto = set()
```

```
In [8]: type(conjunto)
```

```
Out[8]: set
```

De la misma forma podemos obtener un conjunto a partir de cualquier objeto iterable:

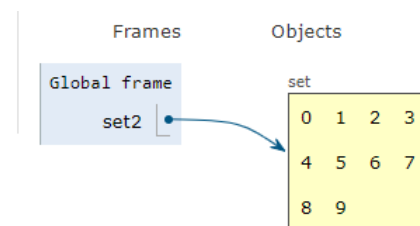
```
In [9]: set1 = set([1, 2, 3, 4])
        set1
```

```
Out[9]: {1, 2, 3, 4}
```

```
In [10]: set2 = set(range(10))
        set2
```

```
Out[10]: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
1 set2 = set(range(10))
→ 2 set2
```



Un set puede ser convertido a una lista y viceversa. En este último caso, los elementos duplicados son unificados.

```
In [11]: lista = list({1, 2, 3, 4})
        lista
```

```
Out[11]: [1, 2, 3, 4]
```

```
In [12]: set_ = set([1, 2, 2, 3, 4])
        set_
```

```
Out[12]: {1, 2, 3, 4}
```

## Elementos

Los conjuntos son objetos mutables. Podemos agregar o eliminar elementos.

```
In [13]: conjunto = {1,2,3,4}
        conjunto
```

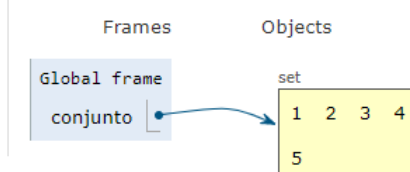
```
Out[13]: {1, 2, 3, 4}
```

## add

```
In [14]: conjunto.add(5)
        conjunto
```

```
Out[14]: {1, 2, 3, 4, 5}
```

```
1 conjunto = {1,2,3,4}
2
→ 3 conjunto.add(5)
```

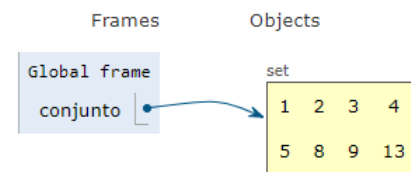


## update

```
In [15]: conjunto.update({2,3,9} ,{4,3,5,8,13})
conjunto
```

```
Out[15]: {1, 2, 3, 4, 5, 8, 9, 13}
```

```
1 conjunto = {1,2,3,4}
2
3 conjunto.add(5)
4
→ 5 conjunto.update({2,3,9} ,{4,3,5,8,13})
```

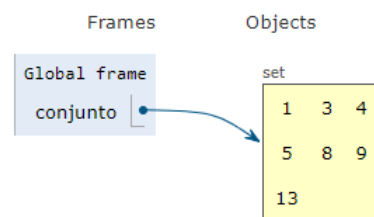


### discard

```
In [16]: conjunto.discard(2)
conjunto
```

```
Out[16]: {1, 3, 4, 5, 8, 9, 13}
```

```
1 conjunto = {1,2,3,4}
2
3 conjunto.add(5)
4
5 conjunto.update({2,3,9} ,{4,3,5,8,13})
6
→ 7 conjunto.discard(2)
```



### remove

```
In [17]: conjunto.remove(5)
conjunto
```

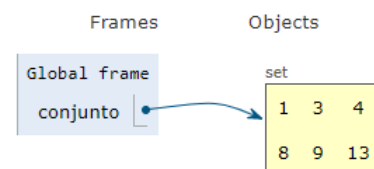
```
Out[17]: {1, 3, 4, 8, 9, 13}
```

Si el elemento pasado como argumento a `discard()` no está dentro del conjunto es simplemente ignorado. En cambio, el método `remove()` opera de forma similar pero en dicho caso da la excepción `KeyError`.

```
In [18]: conjunto.discard(15)
conjunto
```

```
Out[18]: {1, 3, 4, 8, 9, 13}
```

```
1 conjunto = {1,2,3,4}
2
3 conjunto.add(5)
4
5 conjunto.update({2,3,9} ,{4,3,5,8,13})
6
7 conjunto.discard(2)
8
→ 9 conjunto.remove(5)
```



```
In [19]: conjunto.remove(15)
conjunto
```

```
-----
KeyError                                Traceback (most recent call last)
Input In [19], in <cell line: 1>()
----> 1 conjunto.remove(15)
      2 conjunto

KeyError: 15
```

La función **clear()** elimina todos los elementos.

```
In [20]: conjunto = {1,2,3,4,5}
          conjunto
```

```
Out[20]: {1, 2, 3, 4, 5}
```

```
In [21]: conjunto.clear()
          conjunto
```

```
Out[21]: set()
```

El método **pop()** remueve los miembros de un conjunto:

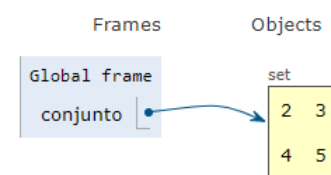
```
In [22]: conjunto = {1, 2, 3, 4, 5}
          conjunto
```

```
Out[22]: {1, 2, 3, 4, 5}
```

```
In [23]: conjunto.pop()
          conjunto
```

```
Out[23]: {2, 3, 4, 5}
```

```
1 conjunto = {1, 2, 3, 4, 5}
2
→ 3 conjunto.pop()
```



```
In [24]: conjunto.pop()
```

```
Out[24]: 2
```

```
In [25]: conjunto
```

```
Out[25]: {3, 4, 5}
```

Para obtener el número de elementos aplicamos función **len()**

```
In [26]: len(conjunto)
```

```
Out[26]: 3
```

**remove() y pop() lanzan la excepción KeyError cuando un elemento no se encuentra en el conjunto o bien éste está vacío, respectivamente.**

### Pertenencia

Ver: [Relación de pertenencia \(https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:nocion-y-determinacion-de-conjuntos/a/141-nocin-y-determinacin-de-conjuntos\)](https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:nocion-y-determinacion-de-conjuntos/a/141-nocin-y-determinacin-de-conjuntos)

Para determinar si un elemento **pertenece** a un conjunto, utilizamos la palabra reservada **in**

```
In [27]: conjunto = {1, 3, 4, 5}
          3 in conjunto
```

```
Out[27]: True
```

```
In [28]: 2 in conjunto
```

```
Out[28]: False
```

### Subconjuntos y superconjuntos

Ver: [Subconjunto, subconjunto propio y superconjunto \(https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/v/subset-strict-subset-and-superset\)](https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/v/subset-strict-subset-and-superset)

Se dice que conjunto2 es un subconjunto de conjunto1 cuando todos los elementos de aquél pertenecen también a éste. Python puede determinar esta relación con el método **issubset()**

```
In [29]: conjunto1 = {1,2,3,4,5}
conjunto2 = {1,2,5}
conjunto2.issubset(conjunto1)
```

Out[29]: True

```
In [30]: conjunto2 < conjunto1
```

Out[30]: True

```
In [31]: conjunto1 = {1,2,3,4,5}
conjunto2 = {10,20,50}
conjunto2.issubset(conjunto1)
```

Out[31]: False

```
In [32]: conjunto2 < conjunto1
```

Out[32]: False

Inversamente, se dice que conjunto1 es un superconjunto de conjunto2 con el método `issuperset()`

```
In [33]: conjunto1 = {1,2,3,4,5}
conjunto2 = {1,2,5}
conjunto1.issuperset(conjunto2)
```

Out[33]: True

La definición de estas dos relaciones nos lleva a concluir que todo conjunto es al mismo tiempo un subconjunto y un superconjunto de sí mismo.

```
In [34]: conjunto1.issubset(conjunto1)
```

Out[34]: True

```
In [35]: conjunto1.issuperset(conjunto1)
```

Out[35]: True

```
In [36]: conjunto2.issubset(conjunto2)
```

Out[36]: True

```
In [37]: conjunto2.issuperset(conjunto2)
```

Out[37]: True

```
In [38]: conjunto1.issuperset(conjunto2)
```

Out[38]: True

```
In [39]: conjunto1.issubset(conjunto2)
```

Out[39]: False

## Relaciones entre conjuntos

### Igualdad

Ver: [Relaciones y operaciones entre conjuntos \(Igualdad de conjuntos\)](https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/a/152-relaciones-y-operaciones-entre-conjuntos-igualdad-de-conjuntos) (<https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/a/152-relaciones-y-operaciones-entre-conjuntos-igualdad-de-conjuntos>)

Dos conjuntos son iguales si y solo si contienen los mismos elementos (a esto se lo conoce como principio de extensionalidad):

```
In [40]: conjunto1 = {1,2,3,4,5}
conjunto2 = {5,4,3,2,1}
```

```
In [41]: conjunto1 == conjunto2
```

Out[41]: True

```
In [42]: conjunto1 = {1,2,3,4,5}
conjunto2 = {0,1,2,5,6,7}
conjunto1 == conjunto2
```

Out[42]: False

Ver: [Intersección y unión de conjuntos \(https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/v/intersection-and-union-of-sets\)](https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/v/intersection-and-union-of-sets)

### Intersección

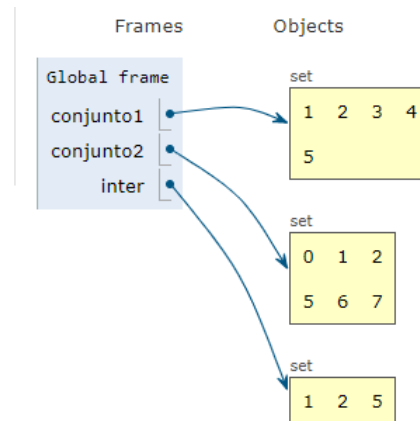
La intersección se realiza con el caracter ( & ) y retorna un nuevo conjunto con los elementos que se encuentran en ambos. **Es simétrica**

```
In [43]: conjunto1 = {1, 2, 3, 4, 5}
conjunto2 = {0,1,2,5,6,7}
```

```
In [44]: conjunto1 & conjunto2
```

Out[44]: {1, 2, 5}

```
1 conjunto1 = {1,2,3,4,5}
2 conjunto2 = {0,1,2,5,6,7}
3
→ 4 inter = conjunto1 & conjunto2
```



```
In [45]: conjunto1.intersection(conjunto2)
```

Out[45]: {1, 2, 5}

```
In [46]: conjunto2.intersection(conjunto1)
```

Out[46]: {1, 2, 5}

```
In [47]: conjunto1.intersection(conjunto2) == conjunto2.intersection(conjunto1)
```

Out[47]: True

### Unión

La unión se realiza con el caracter ( | ) y retorna un conjunto que contiene los elementos que se encuentran en al menos uno de los dos conjuntos involucrados en la operación. **Es simétrica.**

```
In [48]: conjunto1 = {1,2,3,4,5}
conjunto2 = {0,1,2,5,6,7}
```

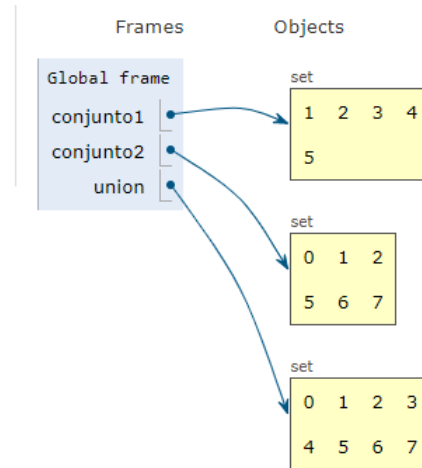
```
In [49]: conjunto1 | conjunto2
```

Out[49]: {0, 1, 2, 3, 4, 5, 6, 7}

```

1 conjunto1 = {1,2,3,4,5}
2 conjunto2 = {0,1,2,5,6,7}
3
→ 4 union = conjunto1 | conjunto2

```



```
In [50]: conjunto1.union(conjunto2)
```

```
Out[50]: {0, 1, 2, 3, 4, 5, 6, 7}
```

```
In [51]: conjunto2.union(conjunto1)
```

```
Out[51]: {0, 1, 2, 3, 4, 5, 6, 7}
```

```
In [52]: conjunto1.union(conjunto2) == conjunto2.union(conjunto1)
```

```
Out[52]: True
```

### Diferencia

Ver: [Complemento relativo o diferencia entre conjuntos](https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/v/relative-complement-or-difference-between-sets) (<https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/v/relative-complement-or-difference-between-sets>)

La diferencia, por último, retorna un nuevo conjunto que contiene los elementos de conjunto1 que no están en conjunto2. **No es simétrica.**

```
In [53]: conjunto1 = {1,2,3,4,5}
         conjunto2 = {0,1,2,5,6,7}
```

```
In [54]: conjunto1 - conjunto2
```

```
Out[54]: {3, 4}
```

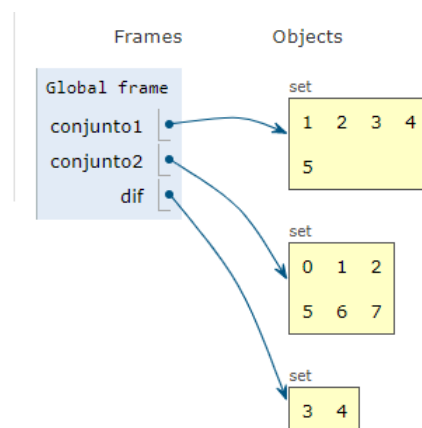
```
In [55]: conjunto1.difference(conjunto2)
```

```
Out[55]: {3, 4}
```

```

1 conjunto1 = {1,2,3,4,5}
2 conjunto2 = {0,1,2,5,6,7}
3
→ 4 dif = conjunto1 - conjunto2

```



```
In [56]: conjunto2 - conjunto1
```

```
Out[56]: {0, 6, 7}
```

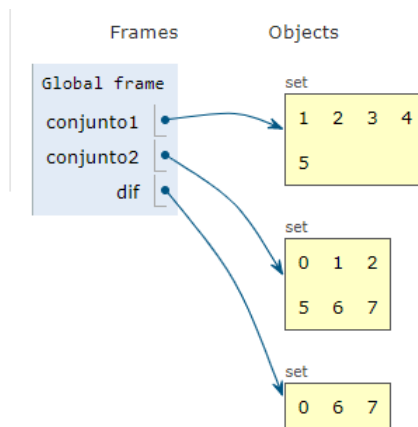
```
In [57]: conjunto2.difference(conjunto1)
```

```
Out[57]: {0, 6, 7}
```

```

1 conjunto1 = {1,2,3,4,5}
2 conjunto2 = {0,1,2,5,6,7}
3
→ 4 dif = conjunto2 - conjunto1

```



```
In [58]: conjunto1.difference(conjunto2) == conjunto2.difference(conjunto1)
```

```
Out[58]: False
```

### Conjunto universal

Ver: [Conjunto universal y complemento absoluto \(https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/v/universal-set-and-absolute-complement\)](https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/v/universal-set-and-absolute-complement).

### Diferencia simétrica

Ver: [Relaciones y operaciones entre conjuntos \(Diferencia simétrica\) \(https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/a/158-relaciones-y-operaciones-entre-conjuntos-diferencia-simtrica\)](https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/a/158-relaciones-y-operaciones-entre-conjuntos-diferencia-simtrica).

La diferencia simétrica (^) retorna un nuevo conjunto el cual contiene los elementos que pertenecen a alguno de los dos conjuntos que participan en la operación pero no a ambos. Podría entenderse como una unión exclusiva.

```
In [59]: conjunto1 = {1,2,3,4,5}
         conjunto2 = {1,2,5,6,7,8}
```

```
In [60]: conjunto1.symmetric_difference(conjunto2)
```

```
Out[60]: {3, 4, 6, 7, 8}
```

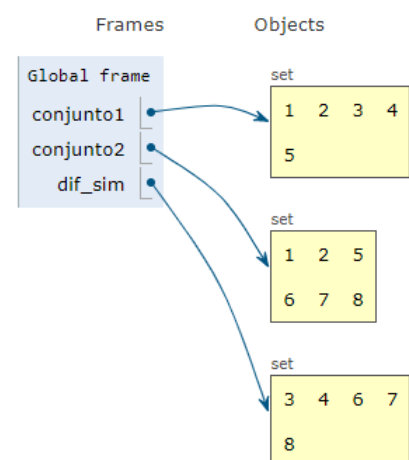
```
In [61]: conjunto1 ^ conjunto2
```

```
Out[61]: {3, 4, 6, 7, 8}
```

```

1 conjunto1 = {1,2,3,4,5}
2 conjunto2 = {1,2,5,6,7,8}
3
→ 4 dif_sim = conjunto1.symmetric_difference(conjunto2)

```



```
In [62]: conjunto2.symmetric_difference(conjunto1)
```

```
Out[62]: {3, 4, 6, 7, 8}
```

```
In [63]: conjunto2 ^ conjunto1
```

```
Out[63]: {3, 4, 6, 7, 8}
```



Por último, se dice que un conjunto es **disconexo** o **disyunto** respecto de otro si no comparten elementos entre sí.

```
In [64]: conjunto1 = {1,2,3,4,5}
         conjunto2 = {1,2,5,6}
         conjunto3 = {7,8,9}
```

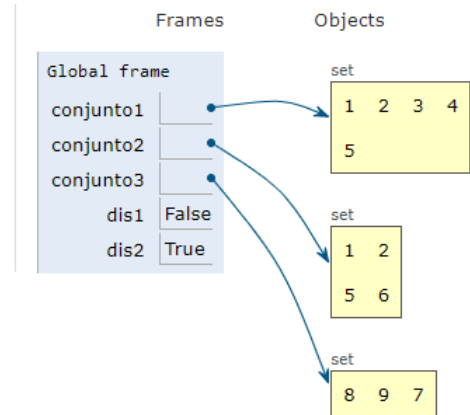
```
In [65]: conjunto1.isdisjoint(conjunto2)
```

```
Out[65]: False
```

```
In [66]: conjunto1.isdisjoint(conjunto3)
```

```
Out[66]: True
```

```
1 conjunto1 = {1,2,3,4,5}
2 conjunto2 = {1,2,5,6}
3 conjunto3 = {7,8,9}
4
5 dis1 = conjunto1.isdisjoint(conjunto2)
6
→ 7 dis2 = conjunto1.isdisjoint(conjunto3)
```



En otras palabras, dos conjuntos son **disconexos** si su intersección es el conjunto vacío.

Ver: [Relaciones y Operaciones entre conjuntos](https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/v/bringing-the-set-operations-together) (<https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/v/bringing-the-set-operations-together>).

ver: [Relaciones y operaciones entre conjuntos \(Leyes del álgebra de conjuntos\)](https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/a/1512-relaciones-y-operaciones-entre-conjuntos-leyes-del-lgebra-de-conjuntos) (<https://es.khanacademy.org/math/aritmetica-pe-pre-u/xce51e392da300f11:relaciones-logicas-y-conjuntos/xce51e392da300f11:relaciones-y-operaciones-entre-conjuntos/a/1512-relaciones-y-operaciones-entre-conjuntos-leyes-del-lgebra-de-conjuntos>).

## Conjuntos inmutables

**frozenset** es una implementación similar a set pero **inmutable**. Es decir, comparte todas las operaciones de conjuntos provistas anteriormente a excepción de aquellas que implican alterar sus elementos (add(), discard(), etc.). La diferencia es análoga a la existente entre una lista y una tupla.

```
In [67]: conjunto1 = frozenset({1,2,3,4,5})
         conjunto1
```

```
Out[67]: frozenset({1, 2, 3, 4, 5})
```

```
In [68]: conjunto2 = frozenset({1,2,5,6})
         conjunto2
```

```
Out[68]: frozenset({1, 2, 5, 6})
```

```
In [69]: conjunto1 & conjunto2
```

```
Out[69]: frozenset({1, 2, 5})
```

```
In [70]: conjunto1 | conjunto2
```

```
Out[70]: frozenset({1, 2, 3, 4, 5, 6})
```

```
In [71]: conjunto1.isdisjoint(conjunto2)
```

```
Out[71]: False
```

```

1 conjunto1 = frozenset({1,2,3,4,5})
2
3 conjunto2 = frozenset({1,2,5,6})
4
5 inter = conjunto1 & conjunto2
6
7 union = conjunto1 | conjunto2
8
→ 9 dis = conjunto1.isdisjoint(conjunto2)

```

