



códigofacilito

---

# Matemáticas Discretas

Fundamentos matemáticos para la programación

Cyn - Matemática





# > Teoría de Conjuntos



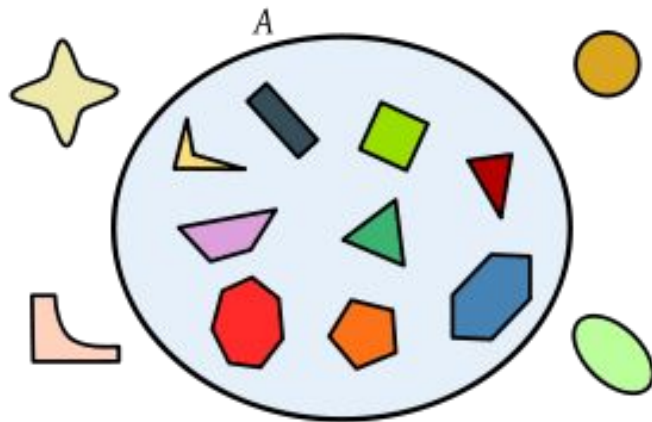


Oh, la redundancia...

# Conjuntos y Elementos

Un **conjunto** es una agrupación de **elementos**.

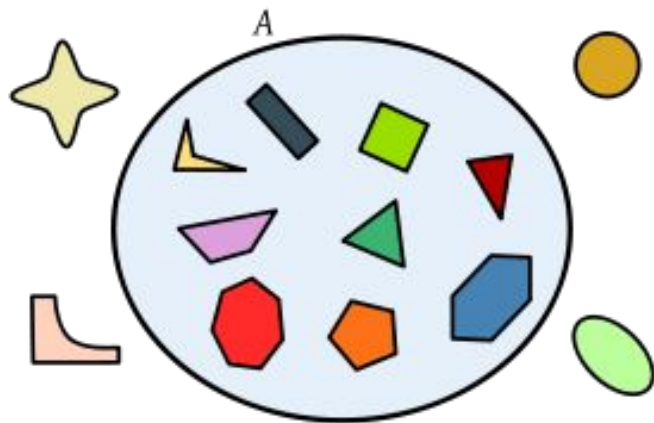
Un **elemento** es un objeto que forma parte de un **conjunto**.





Oh, la redundancia...

# Conjuntos y Elementos



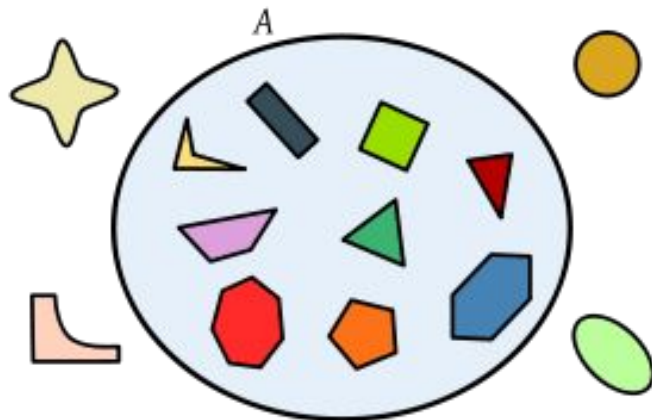


Oh, la redundancia...

# Conjuntos y Elementos



Está en A.





Oh, la redundancia...

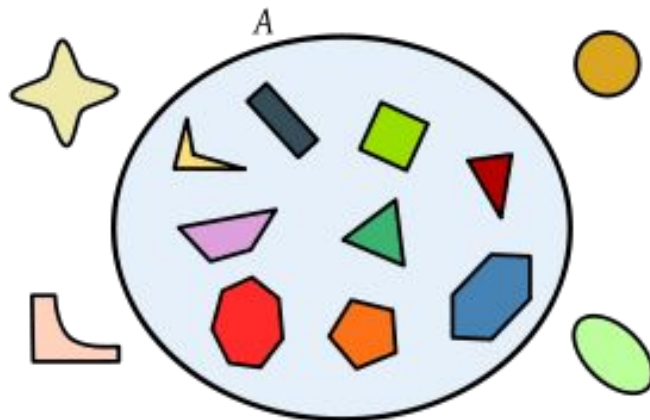
# Conjuntos y Elementos



Está en A.



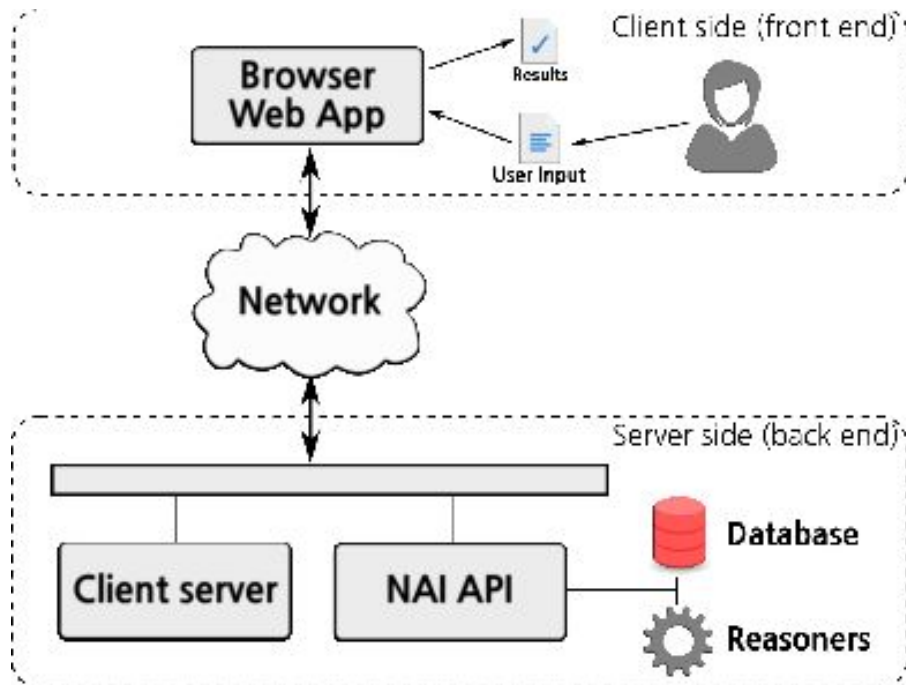
No está en A.





Suena muy matemático...

# ¿Por qué me deben de importar los conjuntos?

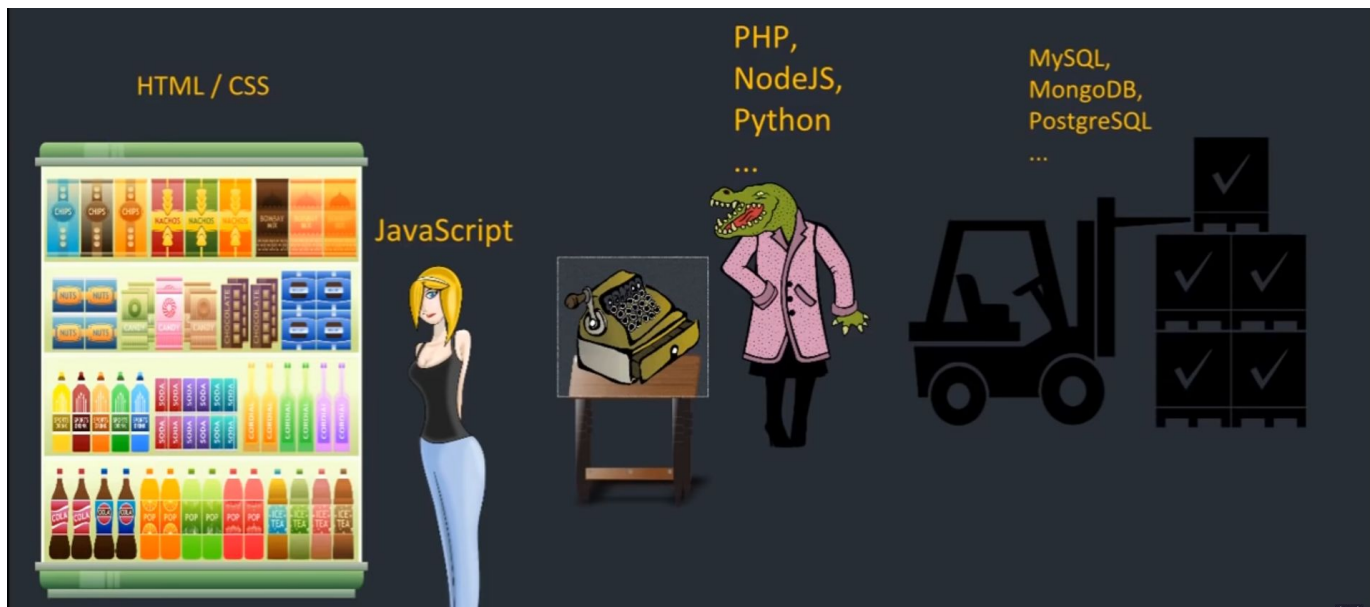






Suena muy matemático...

# ¿Por qué me deben de importar los conjuntos?



Fuente: YouTube – “Logic for programmers”





La **Teoría de Conjuntos** es muy importante para el diseño de base de datos.



## ☰ ¿Por qué me deben de importar los conjuntos?

Lo usual es guardar los datos en objetos, con lo que podremos guardar conjuntos de información.

**Ojo:** Normalmente estos objetos ya están definidos por nuestra base de datos.



Recordando...

# Conjuntos y Elementos



Está en A.



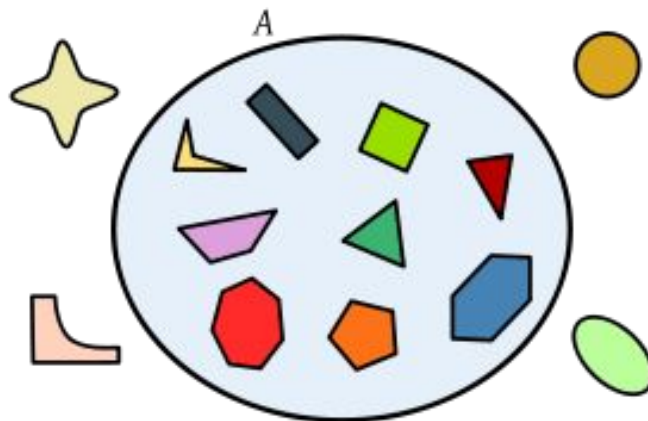
No está en A.



$\in A$



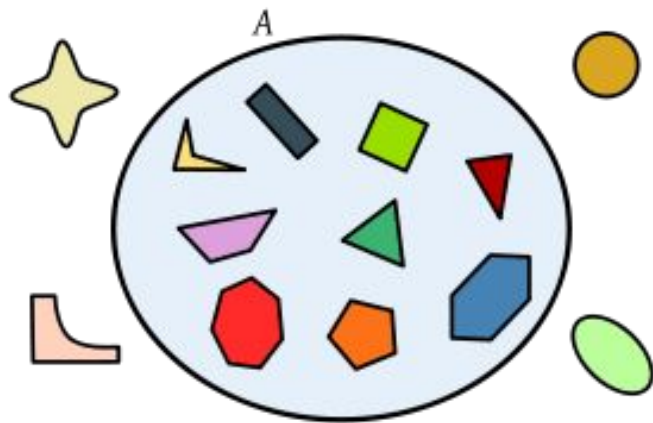
$\notin A$





Recordando...

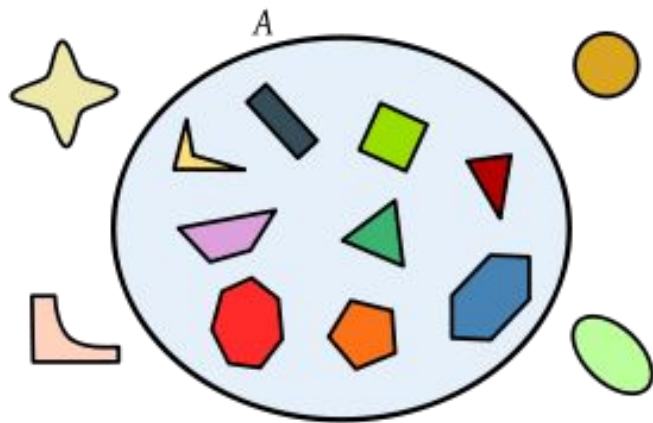
# Conjuntos y Elementos





Recordando...

# Conjuntos y Elementos



TRUE

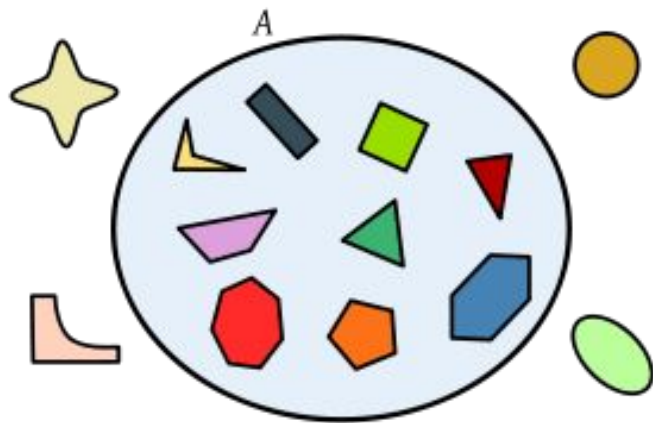






Recordando...

# Conjuntos y Elementos



$\in A$

TRUE



$\in A$

FALSE

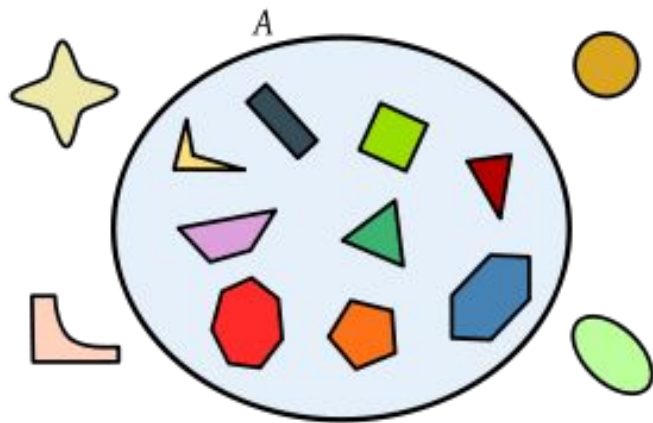


$\notin A$



Recordando...

# Conjuntos y Elementos



$\in A$

TRUE



$\in A$

FALSE



$\notin A$

TRUE



# **Definiendo Conjuntos**

**Por extensión**

**Por intención**

# Definiendo Conjuntos

## Por extensión

$A = \{\text{"Lógica"}, \text{"Conjuntos"}, \text{"Grafos"}, \text{"Combinatoria"}, \text{"Teoría de juegos"}\}$

## Por intención

# Definiendo Conjuntos

## Por extensión

$A = \{\text{"Lógica", "Conjuntos", "Grafos", "Combinatoria", "Teoría de juegos"}\}$

## Por intención

$B =$  Todos los estudiantes del Bootcamp de Introducción a la Programación



Porque hay conjuntos, y hay CONJUNTOS...

# Conjuntos Especiales

## Conjunto Vacío

$\{\}$

$\emptyset$

## Conjunto Universo

$U$



# 📋 Operadores de Conjuntos

Los operadores de conjuntos son los siguientes:

Cardinalidad	$ A $
Complemento	$A^c$
Contención	$A \subset B$
Igualdad	$A = B$
Unión	$A \cup B$
Intersección	$A \cap B$
Diferencia	$A \setminus B$
Diferencia simétrica	$A \Delta B$
Producto Cartesiano	$A \times B$



# Cardinalidad

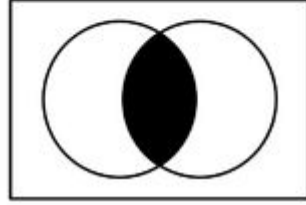
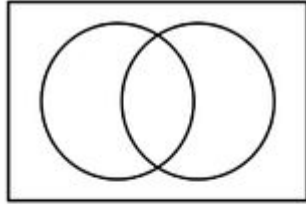
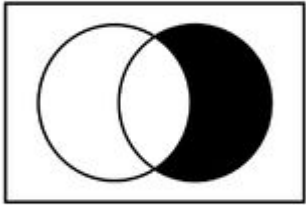
Cuando hablamos de la cardinalidad de  $A$  ( $|A|$ ) hablamos de la cantidad de elementos que tiene el conjunto  $A$ . Siempre será un número natural.



$$|A| = 8$$

# Diagramas de Venn

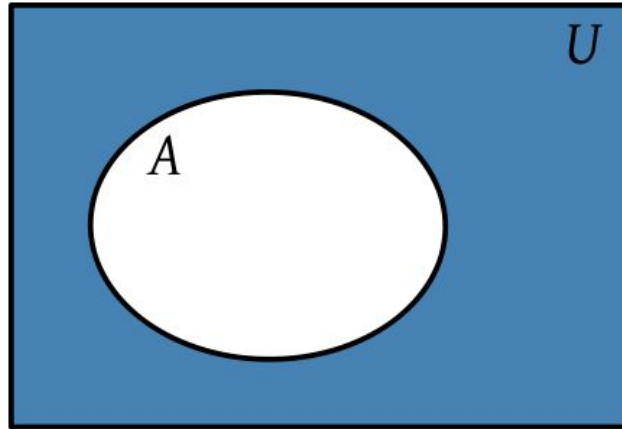
Nos ayudan a representar a los conjuntos y las relaciones entre sí.





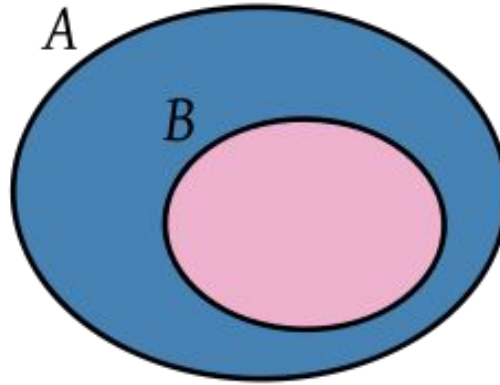
# Complemento

Decir que queremos el complemento de  $A$  ( $A^c$ ) quiere decir que nos quedaremos con los elementos del Universo que NO están en  $A$ .



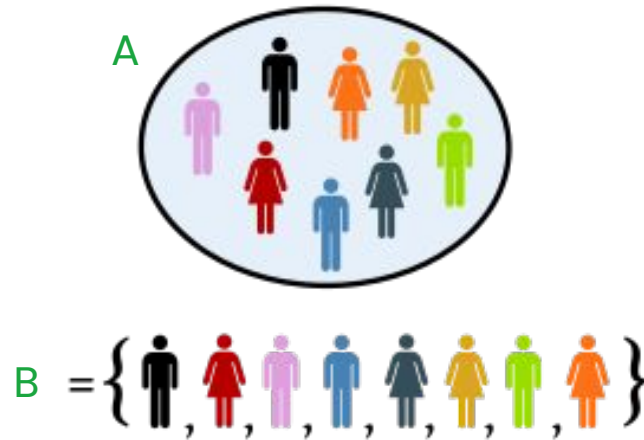
# Contención

Decir que un conjunto está contenido en otro ( $B \subset A$ ) quiere decir que todos los elementos del conjunto B también están en el conjunto A.



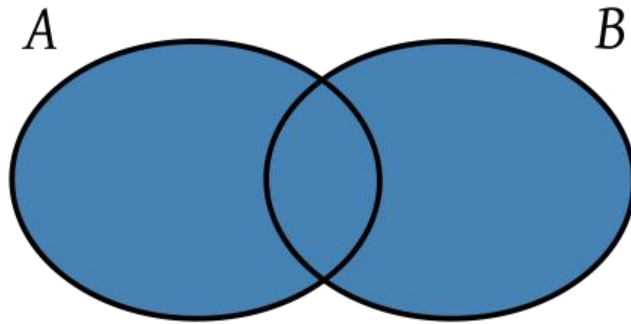
# Igualdad

Decir que un conjunto es igual a otro ( $B = A$ ) quiere decir que todos los elementos del conjunto B también están en el conjunto A y viceversa.



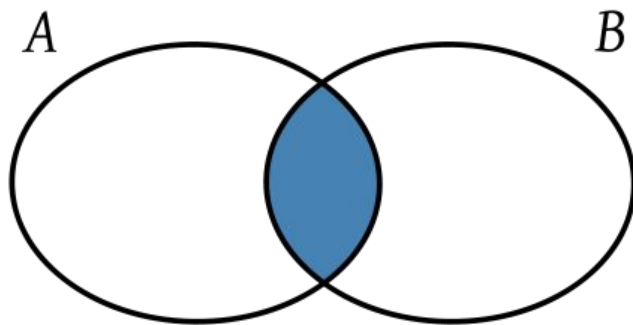
# Unión

Hacer una unión de conjuntos ( $A \cup B$ ) quiere decir que juntaremos todos los elementos de los conjuntos A y B.



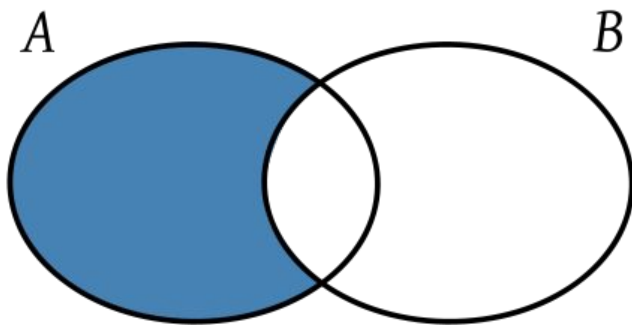
# Intersección

Hacer una intersección de conjuntos ( $A \cap B$ ) quiere decir que sólo nos quedaremos con los elementos que sean tanto parte de A como de B.



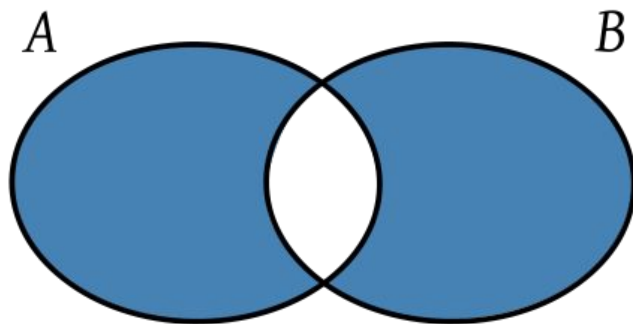
# Diferencia

Hacer una diferencia de conjuntos ( $A \setminus B$ ) quiere decir que sólo nos quedaremos con los elementos que sean estén en A pero que no estén en B.



## Diferencia Simétrica

Hacer una diferencia simétrica de conjuntos ( $A \Delta B$ ) quiere decir que sólo nos quedaremos con los elementos que sólo estén en A o que sólo estén en B.





# Producto Cartesiano

Al hacer un producto cartesiano entre dos conjuntos ( $A \times B$ ) obtendremos los pares ordenados del conjunto A y del conjunto B (a,b) tales que a será elemento de A y b será elemento de B.

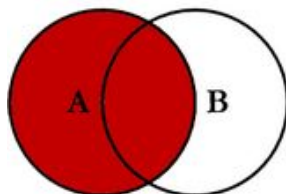
	EmployeeID	EmployeeName
1	1000	Mary Alice
2	1001	Billy Bob
3	1002	Rdger Dodger

	DutyID	Duty
1	1000	Burger Flipper
2	1001	French Fryer
3	1002	Counter

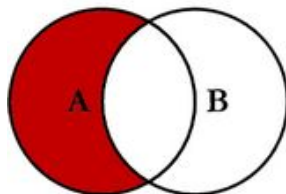
	EmployeeName	Duty
1	Mary Alice	Burger Flipper
2	Billy Bob	Burger Flipper
3	Rdger Dodger	Burger Flipper
4	Mary Alice	French Fryer
5	Billy Bob	French Fryer
6	Rdger Dodger	French Fryer
7	Mary Alice	Counter
8	Billy Bob	Counter
9	Rdger Dodger	Counter

# 📊 Usos en SQL

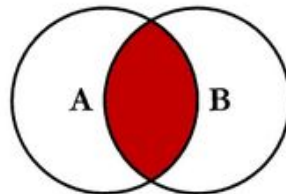
## SQL JOINS



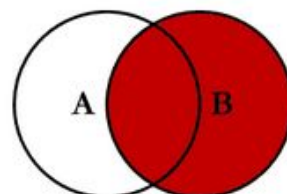
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



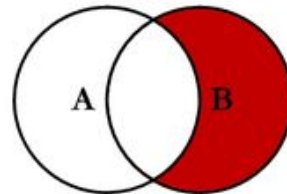
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



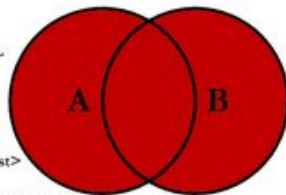
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



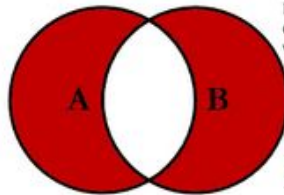
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

# Ejercicios con conjuntos

**Código Facilito** tiene un grupo de 191 estudiantes, de los cuales 10 toman Python, Javascript y PHP; 36 toman Python y Javascript; 20 están en Python y PHP; 18 en Javascript y PHP; 65 en Python; 76 en Javascript y 63 toman PHP.



Apoyarnos en la **Teoría de Conjuntos**  
nos ayudará a obtener los datos que  
necesitemos.





códigofacilito

---

# Matemáticas Discretas

Fundamentos matemáticos para la programación

Cyn - Matemática

