



# Operadores

Los operadores nos permiten comparar datos o expresiones.

Podemos tener operadores de:

- ★ Asignación.
- ★ Comparación
  - × Igualdad estricta y abstracta.
  - × Desigualdad.
  - × Mayor, mayor igual, menor y menor igual.
- ★ Lógicos.

## ASIGNACIÓN

El operador de asignación es el signo “ = ”. Lo utilizamos para asignar (guardar) datos en una variable.

```
let universidad = 'UdeSA';    //El valor 'UdeSA' está guardado en la variable universidad.
```

## IGUALDAD ESTRICTA

El operador de comparación estricta se representa con "**===**". Permite comparar el tipo y el valor de ambos operandos.

```
1 === 1           //true: el tipo de dato y el valor coinciden.
```

```
"1" === 1         //false: el valor coincide pero no así el tipo de dato.
```

## IGUALDAD ABSTRACTA

El operador de comparación abstracta se representa con "**==**". Permite comparar únicamente el valor de ambos operandos.

```
1 == 1            //true: el tipo de dato y el valor coinciden.
```

```
"1" == 1          //true: el valor coincide pero no así el tipo de dato.
```

El operador de desigualdad combina el símbolo de “=” o “==” con el signo de negación representado por “!”.

## DESIGUALDAD ESTRUCTA

```
1  !== 2      //true: el tipo de dato es igual pero el valor es diferente.  
"1" !== 1     //true: el valor coincide pero no así el tipo de dato.  
1  !== 1     //false: el tipo de dato y el valor son iguales.
```

## DESIGUALDAD ABSTRACTA

```
1  != 2       //true: el valor es diferente.  
1  != "2"     //true: el valor es diferente.  
"1" != 1      //false: el valor coincide.  
1  != 1       //false: el valor coincide.
```

# COMPARACIÓN - MAYOR y MENOR

Operador mayor que (>)

**x > y**

Operador mayor o igual (>=)

**x >= y**

Operador menor que (<)

**x < y**

Operador menor o igual (<=)

**x <= y**

## OPERADORES LÓGICOS

Los operadores lógicos nos permiten combinar el resultado de 2 o más comparaciones. Existen tres operadores lógicos: **&&** (and), **||** (or), **negación** (not).

### 1. **AND** (&&)

Permite evaluar si las condiciones de cada término se cumplen (retornan **TRUE**) simultáneamente.

```
{
```

```
(10 > 15) && (10 != 20) // false
```



```
{
```

```
(12 > 4) && (12 != 24) // true
```



# OPERADORES LÓGICOS

## 2. **OR** ( || )

Permite evaluar si **al menos una de las condiciones de los términos** se cumple (retorna **TRUE**).

```
{ }
```

```
(10 > 15) || (10 != 20) // true
```



```
{ }
```

```
(12 > 4) || (12 === 24) // true
```



3. **NOT** ( ! ) niega una el resultado de una condición. Si es **true**, retornará **false** y viceversa.

```
{ }
```

```
!false // true
```

```
!(10 > 15) // true
```





# Gracias