

Operadores lógicos

Objetivos

- Hacer uso de los operadores lógicos para evaluar y simplificar expresiones.
- Invertir una condición.

¿Cómo evaluarías la siguiente expresión?

```
a = 24  
a > 20 y a < 30
```

Para probarla aprenderemos que la expresión `y` se puede escribir como `&&`

```
a = 24  
a > 20 && a < 30 # true
```

¿Cómo evaluarías la siguiente expresión?

```
a = 32  
a > 20 && a < 30
```

```
a = 32  
a > 20 && a < 30 # false
```

Es falsa porque solo cumple uno de los criterios, no ambos.

Motivación

Los operadores lógicos nos ayudan a simplificar los flujos y a evaluar condiciones más complejas. En este capítulo aprenderemos a utilizarlos.

Operadores lógicos

Operador	Nombre	Ejemplo	Resultado
&&	y (and)	false && true	Devuelve true si ambos operandos son true , en este ejemplo se devuelve false.
||	o (or)	false || true	Devuelve true si al menos una de los operando es true , en este ejemplo devuelve true.
!	no (not)	!false	Devuelve lo opuesto al resultado de la evaluación, en este ejemplo devuelve true.

Observemos los siguiente ejemplos:

```
nombre = 'Carlos'
apellido = 'Santana'

nombre == 'Carlos' && apellido == 'Santana'
# true

nombre == 'Carlos' && apellido == 'Vives'
# false

nombre == 'Carlos' || apellido == 'Vives'
# true
```

Identities

Hay varias formas de expresar una afirmación en español, de la misma forma sucede en la lógica y en la programación. Por lo mismo hablamos de identidades.

Veamos ejemplos de esto

'Igual' es lo mismo que 'no distinto'

Negar algo dos veces es afirmarlo (en español, no siempre es así; en programación, sí).

Por lo mismo estas dos afirmaciones son equivalentes:

```
a = 18
puts a == 18 # true
puts !(a != 18) # true
```

Son identidades porque para cualquier valor de `a` ambas expresiones siempre se evaluarán igual. Prueba cambiando el valor asignado:

```
a = 17
puts a == 18 # false
puts !(a != 18) # false
```

Mayor y no menor igual

Un caso similar es la comparación `a > 18`. Decir que `a` **no es mayor a 18**, es decir que es **menor o igual a 18**, (debemos incluir el 18 al negar)

```
a = 18
puts a > 18 # false
puts !(a <= 18) # false
```

```
a = 19
puts a > 18 # true
puts !(a <= 18) # true
```

Unless

Para ayudarnos a escribir las condiciones siempre en positivo, existe una instrucción que es el antónimo del `if`, esta se llama `unless`: Se lee **a menos que...**

```
unless a <= 18
end

if a > 18
end
```

Resumen del capítulo

- Operadores lógicos: Son importantes porque nos ayudan a determinar si una expresión es cierta o falsa (la base de la programación).
- Los operadores lógicos nos pueden ayudar a simplificar expresiones.
- Trataremos de escribir siempre las condiciones en positivo.
- no ($a > b$) es lo mismo que ($a \leq b$).
- no ($a == b$) es lo mismo que ($a != b$).
- if ($a > 18$) es lo mismo que unless ($a \leq 18$).