



Somos un **ecosistema** de desarrolladores de software

Operadores



Lógicos

```
respuesta = a and b
```

```
respuesta = a or b
```

```
respuesta = !a
```



Comparación

```
Compara_igualdad = a == b
```

```
Compara_si_a_mayor_que_b = a > b
```



Aritméticos

```
suma = numeroA + numeroB
```

```
resta = numeroA - numeroB
```

```
multiplicacion = numeroA * numeroB
```

Operadores lógicos

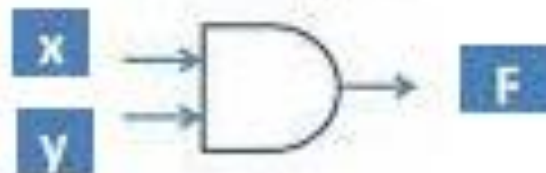
NOT

x	F
0	1
1	0



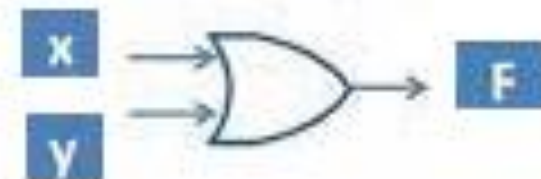
AND

x	y	F
0	0	0
0	1	0
1	0	0
1	1	1



OR

x	y	F
0	0	0
0	1	1
1	0	1
1	1	1



Operador lógico (NOT)



ocupado = not disponible

ocupado	disponible
True	False
False	True

Operador lógico (AND)



`hijos = conejo_fertil and coneja_fertil`

hijos	Conejo_fertil	Coneja_fertil
False	False	False
False	False	True
False	True	False
True	True	True



Operador lógico (OR)



colegio_entrega_la_niña = va_el_papa or va_la_mama

colegio_entrega_la_niña	va_el_papa	va_la_mama
False	False	False
True	False	True
True	True	False
True	True	True

Operadores

Operador	Uso	Descripción
+	a + b	Suma: suma a y b
-	a - b	Resta: resta b de a
*	a * b	Multiplicación: multiplica a por b
/	a / b	División: divide a entre b
//	a // b	División entera: divide a entre b y redondea al número entero más cercano
%	a % b	Módulo: devuelve el residuo de la división de a entre b
**	a ** b	Exponenciación: eleva a a la potencia de b
==	a == b	Igual a: devuelve True si a es igual a b, de lo contrario, False
!=	a != b	No igual a: devuelve True si a no es igual a b, de lo contrario, False
<	a < b	Menor que: devuelve True si a es menor que b, de lo contrario, False
>	a > b	Mayor que: devuelve True si a es mayor que b, de lo contrario, False
<=	a <= b	Menor o igual que: devuelve True si a es menor o igual que b, de lo contrario, False
>=	a >= b	Mayor o igual que: devuelve True si a es mayor o igual que b, de lo contrario, False
and	expr1 and expr2	Operador lógico AND: devuelve True si ambas expresiones son verdaderas
or	expr1 or expr2	Operador lógico OR: devuelve True si al menos una de las expresiones es verdadera
not	not expr	Operador lógico NOT: devuelve True si la expresión es falsa, y viceversa

Operadores de comparación

Operator	Name	Example	Result
==	Equal	5 == 5	True
!=	Not equal	26 != 3	True
>	Greater than	100 > 67	True
<	Less than	89 < 216	True
>=	Greater than or equal to	90 >= 54	True
<=	Less than or equal to	23 <= 77	True

proyectoRentable = gananciaMensual > inversiones*0,15


Operadores de aritméticos

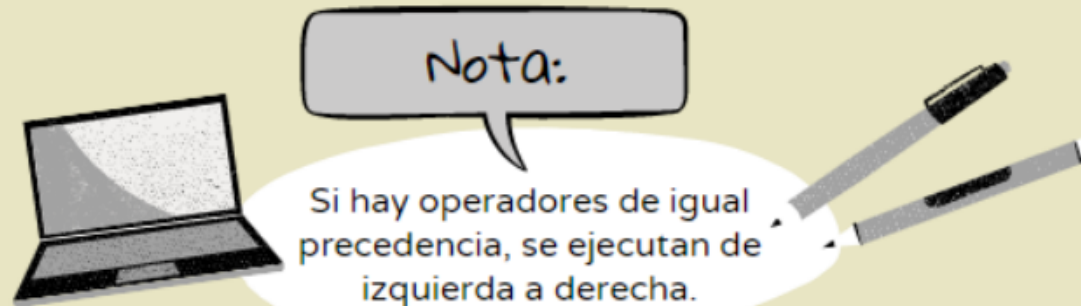
</Riwi>

Operador	Nombre	Descripción	Ejemplo
+	Suma	Suma dos valores	3 + 4
-	Resta	Resta dos valores	5 - 2
*	Multiplicación	Multiplica dos valores	8 * 7
**	Exponenciación	Eleva un valor a otro valor	2 ** 3
/	División	Divide dos valores, retorna real	16 / 2
//	División Entera	Divide dos valores, retorna entero	15 // 2
%	División Modular	Divide dos valores, retorna residuo	5 % 3

Jerarquía de operadores

</Riwi>

OPERADOR	REPRESENTACIÓN	PRIORIDAD
Operaciones entre paréntesis	()	Mayor
Potencia	**	
Multiplicación y División, módulo o residuo, División entera	*, /, %, //	
Suma y resta	+, -	
Operadores relacionales	<, <=, >, >=, !=, ==	
Operador lógico AND	And	
Operador lógico OR	Or	Menor



Jerarquía de operadores

</Riwi>



Expresiones

Las expresiones son combinaciones de constantes, variables, símbolos y paréntesis, que arrojan un valor tal y como se conocen en la notación tradicional de las matemáticas.



`ganancia = precio*unidadesVendidas – costos – impuestos`

Jerarquía de operadores

</Riwi>

Expresiones

Orden de prioridades de operadores

$$R=4 + 6 - 2 = 8$$

Suma y resta tienen misma prioridad, por tanto se evalúan de izquierda a derecha

$$R=4 + 6 * 2 = 16$$

Multiplicación mayor prioridad que la suma

$$R=10 * 5 + 4 ** 3 = 114$$

Potenciación mayor prioridad a Multiplicación

$$R=(10 * (5 + 4)) ** 3 = 729000$$

Se puede hacer una sobre escritura de las prioridades con el uso de paréntesis



1

Suma 5 más 4

2

Multiplica el resultado de la suma por 10

3

Eleva a 3, el resultado de la multiplicación