

# Operadores

## CI-0202 Principios de informática

Sivana Hamer - [sivana.hamer@ucr.ac.cr](mailto:sivana.hamer@ucr.ac.cr)

**Escuela de Ciencias de la Computación e Informática, Universidad de  
Costa Rica**

Licencia: CC BY-NC-SA 4.0



UNIVERSIDAD DE COSTA RICA

## Operador (1)

Son **símbolos** que representan **transformaciones**. Los valores que utiliza el operador se llaman **operandos**.

$$\begin{array}{ccc} & \text{Operador} & \\ & + & \\ \underbrace{22} & & \underbrace{13} \\ \text{Operando} & & \text{Operando} \end{array}$$

## Operador (2)

### Nota

Existen operadores **binarios** (utilizan dos datos) y **unitarios** (utilizan un dato).

### Nota

Los operadores pueden ser agrupados por los operandos y el tipo de dato resultado.

## Operadores aritméticos (1)

Son aquellos que los operandos son valores **numéricos o strings** y el resultado es un valor **numérico o strings**.

- **Número con número:**  $3 + 10 \rightarrow 13$
- **String con string:** "Hello" + "Hola"  $\rightarrow$  "HelloHola"
- **String con número:** "Hola" \* 2  $\rightarrow$  "HolaHola"

## Operadores aritméticos (2)

### Operador de negación

**Descripción:** Convierte un número en negativo.

**Tipo:** Unitario

```
>>> x = -13  
>>> -x  
13
```

### Operador de adición

**Descripción:** Suma dos números.

**Tipo:** Binario

```
>>> 4 + 5  
9
```

## Operadores aritméticos (3)

### Operador de sustracción

**Descripción:** Subtrae el segundo número del primero.

**Tipo:** Binario

```
>>> 10 - 24  
-14
```

### Operador de multiplicación

**Descripción:** Multiplica dos números.

**Tipo:** Binario

```
>>> 2 * 5  
10
```

## Operadores aritméticos (4)

### Operador de división

**Descripción:** Divide el primer número entre el segundo. Realiza la división real (con decimales). El segundo no puede ser 0.

**Tipo:** Binario

```
>>> 8 / 5  
1.6
```

### Operador de división entera

**Descripción:** Divide el primer número entre el segundo. Realiza la división entera (sin decimales).

**Tipo:** Binario

```
>>> 8 // 5  
1
```

## Operadores aritméticos (5)

### Operador de módulo

**Descripción:** Residuo de la división entera del primer número entre el segundo.

**Tipo:** Binario

```
>>> 8 % 5  
3
```

### Operador de potencia

**Descripción:** Eleva el primer número a la potencia del segundo.

**Tipo:** Binario

```
>>> 2 ** 5  
32
```



## Operadores aritméticos (6)

### Operador de concatenación

**Descripción:** Une dos hileras de texto en una tercera hilera.

**Tipo:** Binario

```
>>> "Hola" + " amigos"  
'Hola amigos'
```

### Operador de repetición

**Descripción:** Repite una hilera de texto una cantidad(entera) de veces.

**Tipo:** Binario

```
>>> 5 * "Repeat"  
'RepeatRepeatRepeatRepeatRepeat '
```

## Operadores relacionales (1)

Son aquellos que los operandos son valores **numéricos o hileras** y el resultado es un valor **booleano**.

- **Número con número:**  $3 == 3 \rightarrow \text{True}$
- **String con string:**  $\text{"Hello"}! = \text{"Hola"} \rightarrow \text{False}$

## Operadores relacionales (2)

### Operador de igualdad

**Descripción:** Igual a. No confundir con asignación (=).

**Tipo:** Binario

```
>>> x = 5
>>> print(x == 5)
True
>>> print(x == 2)
False
```

## Operadores relacionales (3)

### Operador de desigualdad

**Descripción:** No igual a.

**Tipo:** Binario

```
>>> x = 5
>>> print(x != 5)
False
>>> print(x != 2)
True
```

## Operadores relacionales (4)

### Operador de mayor/menor (o igual)

**Descripción:** Si el primer número es mayor/menor/mayor o igual/menor o igual que el segundo.

**Tipo:** Binario

```
>>> x = 5
>>> print(5 < 4) # menor
False
>>> print(5 > 4) # mayor
True
>>> print(5 <= 4) # menor o igual
False
>>> print(5 >= 4) # mayor o igual
True
```

## Operadores lógicos (1)

Son aquellos que los operandos son valores **booleanos** y el resultado es un valor **booleano**.

- **Booleano con booleano:** *True and True*  $\rightarrow$  *True*
- **Booleano:** *not True*  $\rightarrow$  *False*

## Operadores lógicos (2)

### Operador de negación

**Descripción:** Invierte el valor del booleano.

**Tipo:** Unitario

```
>>> print(not(True))  
False  
>>> print(not(False))  
True
```

## Operadores lógicos (3)

### Operador de conjunción

**Descripción:** El resultado es verdadero si ambos valores booleanos son verdaderos y es falso de lo contrario. También llamado y.

**Tipo:** Binario

```
>>> print(True and True)
True
>>> print(True and False)
False
>>> print(False and True)
False
>>> print(False and False)
False
```



## Operadores lógicos (4)

### Operador de disjunción

**Disyunción:** El resultado es verdadero cuando alguno de los booleanos son verdaderos y es falso de lo contrario. También llamado o.

**Tipo:** Binario

```
>>> print(True or True)
True
>>> print(True or False)
True
>>> print(False or True)
True
>>> print(False or False)
False
```

## Operadores asignación (1)

Son aquellos que el operando de la izquierda es una **variable**, y el de la derecha **cualquier cosa**. Además **no tiene resultado**.

- **Variable con número:**  $x = 13$
- **Variable con string:**  $x = \text{"Hola"}$
- **Variable con booleano:**  $x = \text{True}$
- **Variable con variable:**  $x = y$

## Operadores asignación (2)

### Operador de asignación

**Descripción:** Asigna a la variable de la izquierda el valor a la derecha.

```
>>> x = 13 # NOTA: No devuelve nada la operacion
>>> print(13)
13
```

## Operadores asignación (3)

### Operador de asignación mas operadores aritméticos

**Descripción:** Se aplica el operador aritmético al valor de la variable con el dato de la derecha, y se guarda el resultado en la variable.

```
>>> x = 5
>>> x+= 5
>>> print(x)
10
>>> x-=3
>>> print(x)
7
```

```
>>> y = "Hola"
>>> y+= " mundo!"
>>> print(y)
Hola mundo!
>>> y *=3
>>> print(y)
Hola mundo!Hola mundo!Hola mundo!
```

# Expresiones

**Combinación de operadores, variables, y valores** que retornan un resultado.


## Ejemplos de expresiones

- $x = 1$
- $y = (x + y) * 4/z > \sin(2 * \pi * t)$

## Nota

Se evalúan las expresiones por **orden de precedencia**.

# Precedencia de operadores (1)

Precedencia	Operadores	Tipo operador
Alta (Primero) 	Potencia (**)	Aritméticos
	Negación ( $-x$ )	
	Multiplicativos ( $*$ / $//$ %)	
	Aditivos ( $+$ $-$ )	
	Comparación ( $<$ $>$ $<=$ $>=$ $==$ $!=$ )	Relacionales
	Negación ( <i>not</i> )	Lógicos
	Y ( <i>and</i> )	
	O ( <i>or</i> )	
Baja (Último)	Asignación ( $=$ )	Asignación

## Precedencia de operadores (2)

### Nota

Los operadores del mismo nivel de precedencia estan asociados de **izquierda a derecha**.

### Nota

Se puede cambiar el orden de precedencia con los **paréntesis**.

```
>>> print(4 / 1 + 1)
5.0
>>> print(4 / (1 + 1))
2.0
```

## Referencias I

L. Villalobos, "Operadores," Material del curso CI-0202, Universidad de Costa Rica, 2019.

C. Swaroop, *A Byte of Python*. Independent, 2020.

J. Elkner, A. B. Downey, and C. Meyers, "How to think like a computer scientist: Learning with python," 2012.

Python. Expressions. [Online]. Available:  
<https://docs.python.org/3/reference/expressions.html>