

# Expresiones & Operadores

Las **expresiones** son construcciones que combinan variables y operadores para crear un valor. En general, las expresiones en Python pueden contener cualquier combinación de los siguientes:

- Constantes literales como números ( `5` , `3.14` ) y cadenas ( `"hola mundo"` ).
- Variables que representan valores.
- Operadores que manipulan valores y variables.
- Llamadas a funciones que devuelven valores.
- Objetos y métodos de objeto que pueden retornar valores.

Aquí hay algunos ejemplos de expresiones:

```
# expresión aritmética
a = 5 + 2

# expresión de comparación
b = (10 < 20)

# expresión de cadena
c = "Hola " + "Mundo"

# expresión con llamada a función
d = len("Hola Mundo")
```

En cada una de estas expresiones, Python evalúa la expresión para producir un solo valor.

Los **operadores** son símbolos especiales o palabras clave que son capaces de manipular o combinar los valores de las variables y constantes en las expresiones. Los operadores en Python se pueden clasificar en varios tipos:

- **Operadores aritméticos:** Realizan operaciones matemáticas como la suma (+), la resta (-), la multiplicación (\*), la división (/), etc.
- **Operadores de comparación:** Realizan comparaciones entre valores, como igual a (==), menor que (<), mayor que (>), no igual a (!=), etc.
- **Operadores lógicos:** Combinan declaraciones booleanas. Los operadores lógicos en Python incluyen `and`, `or` y `not`.
- **Operadores de asignación:** Se utilizan para asignar valores a las variables. Esto incluye el operador de asignación simple (=) y los operadores de asignación compuestos (+=, -=, \*=, /=, etc).
- **Operadores de identidad:** Prueban la identidad de dos objetos. Los operadores de identidad en Python son `is` e `is not`.
- **Operadores de pertenencia:** Prueban si un objeto se encuentra en una secuencia, como una lista o una tupla. Los operadores de pertenencia en Python son `in` y `not in`.

Vamos a revisar algunos operadores comunes en Python y cómo se usan en las expresiones.

**Operadores aritméticos:** Se utilizan para realizar operaciones matemáticas.

```
a = 10
b = 3

suma = a + b      # Suma, resultado: 13
resta = a - b     # Resta, resultado: 7
multiplicacion = a * b # Multiplicación, resultado: 30
division = a / b   # División, resultado: 3.3333333333333335
modulo = a % b    # Módulo, resultado: 1 (resto de la división)
potencia = a ** b  # Potencia, resultado: 1000 (10 al cubo)
division_entera = a // b # División entera, resultado: 3 (cociente entero de la división)
```

**Operadores de asignación:** Se utilizan para asignar valores a las variables. Además del operador de asignación básico `=`, también hay operadores de asignación compuestos que realizan una operación y una asignación en un solo paso.

```
c = 5 # Asigna 5 a c
c += 3 # Equivalente a c = c + 3, c es ahora 8
c -= 2 # Equivalente a c = c - 2, c es ahora 6
c *= 4 # Equivalente a c = c * 4, c es ahora 24
c /= 8 # Equivalente a c = c / 8, c es ahora 3.0
```

**Operadores de comparación:** Se utilizan para comparar valores. El resultado de una comparación es un booleano (`True` o `False`).

```
d = 10
e = 20

es_igual = d == e # Igualdad, resultado: False
no_es_igual = d != e # Desigualdad, resultado: True
menor_que = d < e # Menor que, resultado: True
mayor_que = d > e # Mayor que, resultado: False
menor_o_igual_que = d <= e # Menor o igual que, resultado: True
mayor_o_igual_que = d >= e # Mayor o igual que, resultado: False
```

**Operadores lógicos:** Se utilizan para combinar declaraciones condicionales.

```
f = True
g = False

y_logico = f and g # Y lógico, resultado: False
o_logico = f or g # O lógico, resultado: True
no_logico = not f # No lógico, resultado: False
```

Estos son los operadores más comunes que encontrarás en Python. Existen otros, pero estos te darán un buen comienzo.