

Conceptos básicos y sintaxis de Python

Operadores de Python



Índice

Introducción	3
Operadores	4
Ordenar los operadores por precedencia (de menor a mayor)	5
Combinación de operadores y precedencia	5

Introducción

Ahora que ya hemos visto los primeros tipos básicos de Python, vamos a hacer un interludio para introducir los operadores básicos de Python que nos permitirán operar tanto con números como con el resto de tipos que vayamos conociendo.

Como veremos en unidades posteriores, un programa en Python se compone de módulos, éstos de sentencias que, a su vez, se componen de expresiones.

Una expresión es una combinación de números (u otros objetos de Python) y operadores que, cuando es ejecutada, produce un valor. En esta sección vamos a conocer los operadores que permiten combinar objetos dentro de las expresiones.

Operadores

Operador	Descripción
yield x	Protocolo de generadores “send”
lambda args: expresión	Función anónima
x if y else z	Selección ternaria (retorna x si y es cierta)
x or y	OR lógico (y es evaluada sólo si x es falsa)
x and y	AND lógico (y es evaluada sólo si x es verdadera)
not x	Negación Lógica
x in y, x not in y	Operadores de membresía
x is y, y is not y	Operadores de identidad
x < y, x <= y, x > y, x >= y	Comparación de magnitudes. Set, subset y superset
x == y, x != y	Operadores de igualdad
x y	Bit a bit (bitwise) OR. Unión de sets (conjuntos)
x ^ y	Bitwise XOR. Diferencia simétrica de sets
x & y	Bit a bit AND. Intersección de sets (conjuntos)
x << y, x >> y	Desplazamiento de x en y bits a izquierda y derecha
x + y	Adición. Concatenación
x - y	Substracción. Diferencia entre sets (conjuntos)
x * y	Multiplicación. Repetición
x % y	Resto de la división
x / y	División real (verdadera)
x // y	División truncada
-x, +x	Negación. Identidad
~x	Negación Bit a bit
x ** y	Potencia (exponenciación)
x[i]	Indexado (secuencias, mapeados, otros)
x[i:j:k]	Troceado (slicing)
x(...)	Llamada (función, método, clase, etc.)
x.attr	Referencia a un atributo
(...)	Tupla, expresión, expresión de generador
[...]	Lista, lista por comprensión
{...}	Diccionario, Set, comprensión de diccionarios y sets

Muchos de estos operadores no los utilizaremos en este curso, pero conservad la tabla para futura referencia.

Combinación de operadores y precedencia

Cuando en una misma expresión se combinan varios operadores, hay que tener en cuenta qué operadores preceden (se evalúan antes) a otros. La [tabla](#) nos muestra todos los operadores de Python ordenados por su precedencia. Es decir, los operadores que están más arriba tienen menor precedencia sobre los que se encuentran más abajo en la tabla. Para aquellos operadores que se encuentran en la misma fila, la evaluación se hace de izquierda a derecha excepto para la exponenciación, que se agrupa de derecha a izquierda y las comparaciones, que se encadenan de izquierda a derecha.

De esta manera, en una operación donde se encadenan multiplicaciones y sumas, por ejemplo, $A * B + C * D$ se realizan primero las multiplicaciones y luego la suma.

Una manera de saltarnos la precedencia es utilizando paréntesis para aislar expresiones. En Python, las expresiones entre paréntesis se evalúan siempre antes que el resto. De esta manera, en la expresión anterior, si quisiéramos evaluar antes la suma, haríamos lo siguiente: $A * (B + C) * D$.

```
>>> 2 * 3 + 2 * 4           # Multiplicación precede a suma
14

>>> 2 * (3 + 2) * 4         # Expresiones entre paréntesis se evalúan primero
40
```