

Operadores

¿De qué nos ocuparemos?

Qué podemos hacer con los datos, específicamente el rango de operaciones aritméticas y de cadena, y los operadores lógicos y de listas.

Ahora que sabemos cómo crear datos de diferentes tipos y asignarlos a diferentes variables para su uso posterior, es lícito preguntarnos ¿qué podemos hacer con todo esto? ¿Cómo se manipulan estos datos? Para todo esto sirven los *operadores*: operan sobre los datos. No voy a cubrir todos los operadores de Python, sino que me concentraré en aquellos que están presentes en prácticamente todos los lenguajes de programación.

Operadores aritméticos

Ya hemos visto la mayor parte de los operadores aritméticos en el capítulo dedicado a las "Secuencias", sin embargo, no viene mal una recapitulación:

Operadores Aritméticos

Operador	Descripción	Efecto
M+N	Adición	(Matemático)
M-N	Sustracción	(Matemático)
M*N	Multiplicación	(Matemático)
M/N	División	(Matemático)
M**N	M elevado a la N	(Matemático)
M>>N	Mover M, N lugares a la derecha	(Binario)
M«N	Mover M, N lugares a la izquierda	(Bitwise)
~M	Complemento de M	(Binario)
M & N	AND binario de M,N	(Binario)
M N	OR binario de M,N	(Binario)

Operadores de Cadena

Hay varias operaciones que se pueden realizar con las cadenas, varias de las cuales vienen predeterminadas en Python o son provistas en módulos que debemos *importar* (de la misma manera que hicimos con sys en el capítulo acerca de las secuencias).

Operadores de Cadena

Operador	Descripción	Ubicación
S+T	Concatenación	Predeterminado
S*N	Repetición	Predeterminado
upper(S)	Convierte S en mayúsculas	módulo string
split(S)	Separa S en "palabras"	módulo string

Operadores Lógicos (o Booleanos)

Operadores Lógicos

Operador	Descripción	Resultado
A and B	AND	"true" si A y B son ambos verdaderos
A or B	OR	"true" si ambos o al menos uno de A y B son verdaderos
A == B	Igualdad	"true" si A y B son iguales
A <> B	Desigualdad	"true" si A y B NO son iguales
not B	Desigualdad	"true" si B no es verdadero (siendo B un valor o una expresión booleana)

Nota: el último opera sobre un solo valor, los restantes requieren siempre dos valores.

Operadores de Listas

Operadores de Listas

Operador	Descripción	Locación
L+J	Concatenación	Predeterminado
L*N	Repetición	Predeterminado
L[I]	Índice	Predeterminado

Notas Generales

Muchos lenguajes de programación implementan operaciones de las que carecen otros lenguajes. Muchas veces estas operaciones "únicas" son la razón de ser de la creación de muchos lenguajes, y ciertamente son factores importantes en la "popularidad" que el lenguaje pueda adquirir entre los programadores.

Por ejemplo Python presenta operaciones poco comunes como el recorte de listas (`Spam[X:Y]`), asignación de Tuples (`X,Y = 12,34`), etc.

Algunos lenguajes de programación (en los cuales se incluye Python), soportan lo que se conoce como *superposición de operadores*, que es la opción que tiene el programador de definir su propia versión de los operadores predeterminados para los tipos de datos recién creados. Otros lenguajes (p.ej. Lisp) van más allá y permiten cambiar el comportamiento de los operadores predeterminados, pero esto no es general una buena idea, ya que vuelve imposible para otra persona el entender la mecánica interna del programa (imagínate el peligro de alguien que redefine la adición para que formatee el disco rígido!).

Tcl va más lejos aún, permitiendo al programador no sólo definir sus propios operadores sino también las estructuras de control del lenguaje (algo que veremos más adelante). Esta es una característica muy poderosa de Tcl, pero es algo que puede confundir a los demás programadores si se la usa sin cuidado.

Para recordar

- Existen muchos operadores en cada uno de los lenguajes de programación, y parte del aprendizaje de un nuevo lenguaje implica familiarizarse con los tipos de datos que este maneja y con los operadores disponibles para cada uno de estos tipos.
- Un mismo operador (p.ej. la adición) puede estar disponible para tipos diferentes, pero los resultados pueden no ser idénticos o incluso no estar relacionados.

[Anterior](#) [Próxima](#) [Contenido](#)

Si tenés preguntas o sugerencias, enviame un e-mail a alan.gauld@btinternet.com