

Observe que a expressão `x ++` nem sempre é a mesma que `x = x+1`.

O operador `++` nunca executa concatenação de string: sempre converte seu operando em um número e o incrementa. Se `x` é a corda `"1"`, `++ x` é o número 2, mas `x+1` é a string `"11"`.

Observe também que, devido ao semicolon automático do JavaScript Inserção, você não pode inserir uma quebra de linha entre o pós-incremento operador e o operando que o precede. Se você fizer isso, JavaScript tratará o operando como uma declaração completa por si só e inserirá um Semicolon antes dele.

Este operador, tanto em suas formas de pré e pós-incremento, é a maioria comumente usado para incrementar um contador que controla um loop (§5.4.4).

Decremento (-)

O operador espera um operando LValue. Converte o valor de o operando a um número, subtrai 1 e atribui o decrementado Valor de volta ao operando. Como o operador `++`, o valor de retorno de - depende de sua posição em relação ao operando. Quando usado Antes do operando, ele diminui e retorna o decrementado valor. Quando usado após o operando, diminui o operando, mas Retorna o valor indecrementado. Quando usado após seu operando, não A quebra de linha é permitida entre o operando e o operador.

4.8.3 Operadores bitwise

Os operadores bitwee

Representação binária de números. Embora eles não realizem

Operações aritméticas tradicionais, elas são classificadas como aritmética operadores aqui porque operam em operandos numéricos e retornam um valor numérico. Quatro desses operadores realizam álgebra booleana no

Bits individuais dos operandos, se comportando como se cada um pouco em cada operando