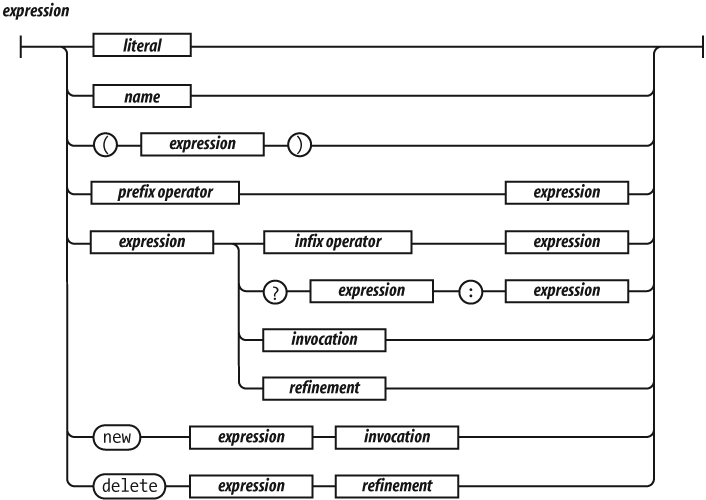
Expressions

# Expresiones



Las expresiones más sencillas son un valor literal (como una cadena o un número), una variable, un valor incorporado (true, false, null, undefined, NaN o Infinity), una expresión de invocación precedida de new, una expresión de refinamiento precedida de delete, una expresión entre paréntesis, una expresión precedida de un operador prefijo o una expresión seguida de:

* Un operador infijo y otra expresión
* El operador ternario ? seguido de otra expresión, luego por :, y luego por otra expresión más
* Una invocación
* Un refinamiento

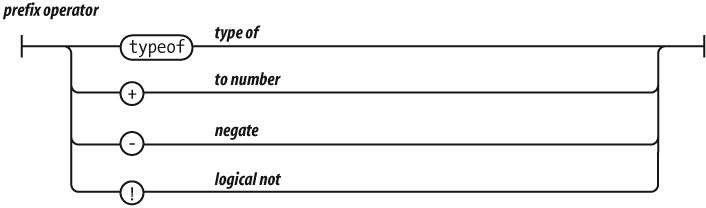
El operador ternario ? toma tres operandos. Si el primer operando es verdadero, produce el valor del segundo operando. Pero si el primer operando es falso, produce el valor del tercer operando.

Los operadores situados en la parte superior de la lista de precedencia de operadores de [la Tabla 2-1](ch02s06.html#operator_precedence) tienen mayor precedencia y se vinculan más estrechamente. Los operadores de la parte inferior tienen la precedencia más baja. Se pueden utilizar paréntesis para alterar la precedencia normal, así:

2 + 3 \* 5 === 17  
(2 + 3) \* 5 === 25

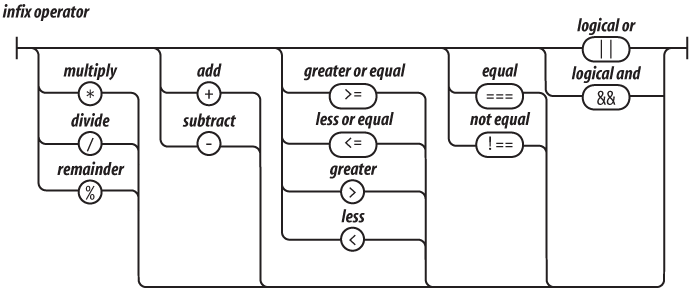
Tabla 2-1. Precedencia de los operadores

|  |  |
| --- | --- |
| . [] ( ) | Refinamiento e invocación |
| delete new typeof + - ! | Operadores unarios |
| \* / % | Multiplicación, división, resto |
| + - | Suma/concatenación, resta |
| >= <= > < | Desigualdad |
| === !== | Igualdad |
| && | Lógica y |
| || | Lógica o |
| ?: | Ternario |



Los valores producidos por typeof son 'number', 'string', 'boolean', 'undefined', 'function', y 'object'. Si el operando es una matriz o null, el resultado es 'object', que es incorrecto. Encontrarás más información sobre typeof en [el Capítulo 6](ch06.html) y en el [Apéndice A](apa.html).

Si el operando de ! es verdadero, produce false. En caso contrario, produce true.

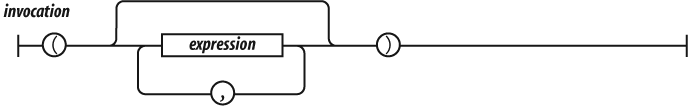


El operador + suma o concatena. Si quieres que sume, asegúrate de que ambos operandos son números.

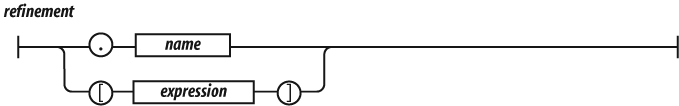
El operador / puede producir un resultado no entero aunque ambos operandos sean enteros.

El operador && produce el valor de su primer operando si éste es falso. En caso contrario, produce el valor del segundo operando.

El operador || produce el valor de su primer operando si el primer operando es verdadero. En caso contrario, produce el valor del segundo operando.



La invocación provoca la ejecución del valor de una función. El operador de invocación es un par de paréntesis que siguen al valor de la función. Los paréntesis pueden contener argumentos que se entregarán a la función. En el [Capítulo 4](ch04.html) se tratará mucho más sobre las funciones.



Un refinamiento se utiliza para especificar una propiedad o elemento de un objeto o matriz. Esto se describirá con detalle en el capítulo siguiente.