

operando, mas a expressão "3" \* "5" é legal porque JavaScript pode converter os operando em números. O valor dessa expressão é O número 15, não a corda "15", é claro. Lembre -se também disso Todo valor de JavaScript é "verdade" ou "falsidade", então os operadores que Espere que os operando booleanos funcionem com um operando de qualquer tipo. Alguns operadores se comportam de maneira diferente, dependendo do tipo de operando usados ??com eles. Mais notavelmente, o operador + adiciona numérico operando, mas concatena operandos de string. Da mesma forma, a comparação operadores como <executar comparação em numéricos ou alfabéticos Encomende, dependendo do tipo de operandos. As descrições de operadores individuais explicam suas dependências de tipo e especificam o que Digite conversões que eles executam.

Observe que os operadores de atribuição e alguns dos outros operadores Listado na Tabela 4-1 Espere um operando do tipo LVAL. lvalue é a termo histórico que significa ?uma expressão que pode aparecer legalmente no Lado esquerdo de uma expressão de atribuição. ?Em JavaScript, variáveis, Propriedades dos objetos e elementos das matrizes são lvalues.

#### 4.7.3 Efeitos colaterais do operador

Avaliar uma expressão simples como 2 \* 3 nunca afeta o estado de Seu programa e qualquer computação futura que seu programa executa será não seja afetado por essa avaliação. Algumas expressões, no entanto, têm efeitos colaterais e sua avaliação podem afetar o resultado do futuro Avaliações. Os operadores de atribuição são o exemplo mais óbvio: se Você atribui um valor a uma variável ou propriedade, que altera o valor de Qualquer expressão que use essa variável ou propriedade. O ++ e -