

O JavaScript possui dois operadores que testam se dois valores são iguais. O "operador estrito da igualdade", `===`, não considera seus operandos serem iguais se eles não forem do mesmo tipo, e este é quase sempre o operador correto para usar ao codificar. Mas porque JavaScript é tão flexível com conversões de tipo, ele também define o operador `==` com um flexível Definição de igualdade. Todas as seguintes comparações são verdadeiras, para exemplo:

`null == indefinido // => true`: esses dois valores são tratados como iguais.

`"0" == 0 // => true`: string se converte em um número antes de comparar.

`0 == false // => true`: boolean converte em número antes de comparar.

`"0" == false // => true`: ambos os operandos se convertem para 0 antes de comparar!

§4.9.1 explica exatamente quais conversões são executadas pelo `==` operador para determinar se dois valores devem ser considerado iguais.

Lembre-se de que a conversibilidade de um valor para outro não implica igualdade desses dois valores. Se `indefinido` é usado onde um booleano

O valor é esperado, por exemplo, será convertido para `false`. Mas isso faz não significa que `indefinido == false`. Operadores JavaScript e

As declarações esperam valores de vários tipos e realizam conversões para Esses tipos. A declaração se converte indefinida a falsa, mas

O operador `==` nunca tenta converter seus operandos em booleanos.

### 3.9.2 Conversões explícitas

Embora o JavaScript execute muitas conversões de tipo automaticamente,