

O valor zero negativo também é um tanto incomum. Compara igual (mesmo usando o teste estrito de igualdade de JavaScript) para zero positivo, que significa que os dois valores são quase indistinguíveis, exceto quando usado como divisor:

Seja zero = 0; // zero regular

deixe negz = -0; // Zero negativo

zero === negz // => true: zero e zero negativo são

igual

1/zero === 1/negz // => false: infinito e -infinity são

não é igual

3.2.4 Ponto flutuante binário e erros de arredondamento

Existem infinitamente muitos números reais, mas apenas um número finito de

eles (18.437.736.874.454.810.627, para ser exato) podem ser representados

Exatamente pelo formato de ponto flutuante JavaScript. Isso significa que quando

Você está trabalhando com números reais em JavaScript, a representação de

O número geralmente será uma aproximação do número real.

A representação do ponto flutuante IEEE-754 usado por JavaScript (e

praticamente qualquer outra linguagem de programação moderna) é um binário

Representação, que pode representar exatamente frações como 1/2, 1/8,

e 1/1024. Infelizmente, as frações que usamos mais comumente

(especialmente quando realizar cálculos financeiros) são decimais

Frações: 1/10, 1/100 e assim por diante. Ponto flutuante binário

As representações não podem representar exatamente os números tão simples quanto 0,1.

Os números de JavaScript têm muita precisão e podem aproximar 0,1

muito de perto. Mas o fato de esse número não pode ser representado

Exatamente pode levar a problemas. Considere este código: