Seja x = 0,3 - .2;// trinta centavos menos 20 centavos
Seja y = .2 - .1;// vinte centavos menos 10 centavos
x === y // => false: os dois valores não são o
mesmo!
x === .1 // => false: .3-.2 não é igual a .1
y === .1 // => true: .2-.1 é igual a .1
Devido ao erro de arredondamento, a diferença entre as aproximações
de .3 e .2 não é exatamente o mesmo que a diferença entre o
aproximações de .2 e .1.É importante entender que isso
O problema não é específico para o JavaScript: afeta qualquer programação
idioma que usa números de ponto flutuante binário.Além disso, observe que o
Os valores x e y no código mostrado aqui estão muito próximos um do outro e
para o valor correto.Os valores calculados são adequados para quase qualquer
propósito;O problema só surge quando tentamos comparar valores
para igualdade.
Se essas aproximações de ponto flutuante forem problemáticas para o seu
Programas, considere o uso de números inteiros em escala.Por exemplo, você pode
manipular valores monetários como centavos inteiros em vez de fracionários
dólares.
3.2.5 números inteiros de precisão arbitrária com bigint
Uma das características mais recentes do JavaScript, definida no ES2020, é um novo
Tipo numérico conhecido como bigint.No início de 2020, tem sido
implementado em Chrome, Firefox, Edge e Nó, e há um
Implementação em andamento no Safari.Como o nome indica, Bigint é um
Tipo numérico cujos valores são inteiros.O tipo foi adicionado a
Javascript principalmente para permitir a representação de números inteiros de 64 bits, que
são necessários para a compatibilidade com muitas outras linguagens de programação