

Número.min\_value/2 // => 0: subflow  
-Number.min\_value/2 // => -0: Zero negativo  
-1/infinito // -> -0: também negativo 0  
-0  
// As seguintes propriedades de número são definidas no ES6  
Número.parseInt () // O mesmo que o parseInt global ()  
função  
Número.parseFloat () // O mesmo que o parseFloat global ()  
função  
Número.isnan (x) // é x o valor da nan?  
Número.isfinite (x) // é x um número e finito?  
Número.isinteger (x) // é x um número inteiro?  
Número.isSafeInteger (x) // é x um número inteiro  $-(2^{53}) < x < 2^{53}$ ?  
Número.min\_safe\_integer // =>  $-(2^{53} - 1)$   
Número.max\_safe\_integer // =>  $2^{53} - 1$   
Número.epsilon // =>  $2^{-52}$ : menor diferença  
entre números

O valor não-número de número tem um recurso incomum no JavaScript: ele faz não comparar igual a nenhum outro valor, inclusive a si mesmo. Isso significa isso você não pode escrever `x === nan` para determinar se o valor de um variável x é nan. Em vez disso, você deve escrever `x !== X` ou `Number.isnan (x)`. Essas expressões serão verdadeiras se, e somente se, x tem o mesmo valor que a NAN constante global.

A função global `isNaN ()` é semelhante ao `numero.isnan ()`. Isto retorna verdadeiro se seu argumento for nan, ou se esse argumento é um não valor numérico que não pode ser convertido em um número. O relacionado função `numero.isfinite ()` retorna true se seu argumento for um Número diferente da NAN, Infinito ou -Infinitude. O global A função `isfinite ()` retorna true se seu argumento for ou pode ser convertido em um número finito.