

ES6 define mais funções no objeto de matemática:

`Math.cbrt (27) // => 3`: raiz de cubo

`Math.hypot (3, 4) // => 5`: raiz quadrada da soma dos quadrados de todos os argumentos

`Math.Log10 (100) // => 2`: Logaritmo Base-10

`Math.log2 (1024) // => 10`: LOGARITHM BASE-2

`Math.log1p (x) // log natural de (1+x)`; preciso para muito pequeno x

`Math.expm1 (x) // math.exp (x) -1`; o inverso de

`Math.log1p ()`

`Math.sign (x) // -1, 0 ou 1` para argumentos `<`, `==`, ou `>` 0

`Math.imul (2,3) // => 6`: multiplicação otimizada de 32 bits

Inteiros

`Math.clz32 (0xf) // => 28`: Número de zero bits líderes em um

Inteiro de 32 bits

`Math.trunc (3.9) // => 3`: converter em um número inteiro truncando parte fracionária

`Math.Fround (X) // Round to mais próximo do número de flutuação de 32 bits`

`Math.sinh (x) // seno hiperbólico`. Também `Math.Cosh ()`,

`Math.tanh ()`

`Math.asinh (x) // arcsina hiperbólica`. Também `math.acosh ()`,

`Math.atanh ()`

A aritmética em JavaScript não levanta erros em casos de transbordamento, subfluxo, ou divisão por zero. Quando o resultado de uma operação numérica é maior que o maior número representável (estouro), o resultado é

Um valor especial do infinito, infinito. Da mesma forma, quando o absoluto o valor de um valor negativo se torna maior que o valor absoluto do

O maior número negativo representável, o resultado é o infinito negativo, -

Infinidade. Os valores infinitos se comportam como seria de esperar: acrescentando, subtrair, multiplicar ou dividi -los por qualquer coisa resulta em um valor infinito (possivelmente com o sinal revertido).

O fluxo ocorre quando o resultado de uma operação numérica está mais próxima de