você especifica).toprecision () converte um número em uma string com o
Número de dígitos significativos que você especificar.Usa notação exponencial se
O número de dígitos significativos não é grande o suficiente para exibir o inteiro
parte inteira do número.Observe que todos os três métodos ao redor do
dígitos ou almofada à direita com zeros, conforme apropriado.Considere o seguinte
Exemplos:

```
Seja n = 123456.789;
n.tofixado (0) // => "123457"
n.tofixado (2) // => "123456.79"
n.tofixado (5) // => "123456.78900"
n.toexponencial (1) // => "1.2e+5"
n.toexponencial (3) // => "1.235e+5"
N.Toprecision (4) // => "1.235e+5"
N.Toprecision (7) // => "123456.8"
N.Toprecision (10) // => "123456.7890"
```

Além dos métodos de formatação de números mostrados aqui, o
Classe Intl.NumberFormat define uma mais geral, internacionalizada
Método de formatação por número.Veja §11.7.1 para obter detalhes.
Se você passar uma string para a função de conversão número (), ela tenta
Para analisar essa corda como um número inteiro ou literal de ponto flutuante.Essa função
Funciona apenas para os números inteiros da Base-10 e não permite caracteres à direita
que não fazem parte do literal.O parseint () e
funções parsefloat () (essas são funções globais, não métodos de
qualquer classe) são mais flexíveis.parseint () passa apenas números inteiros, enquanto
Parsefloat () analisa números inteiros e pontos flutuantes.Se a
String começa com "0x" ou "0x", parseint () o interpreta como um
Número hexadecimal.Parseint () e Parsefloat () Skip
liderando espaço em branco, analisam o maior número possível de personagens numéricos, e