

Programa o Estruturada

Aula 06 - Operadores JavaScript: Aritm tica

Videoaula 04 - Convertendo strings em n meros



Videoaula 04 - Convertendo strings em números

Diferentemente de como ocorre com o símbolo `+`, o JavaScript tenta converter as *strings* em números quando usamos outras operações como divisão, multiplicação, subtração, módulo e exponenciação.

Caso essa conversão não seja possível, ou seja, caso um dos operandos seja uma *string* não numérica, o resultado na operação será o número especial `NaN`, sigla da expressão em inglês *Not a Number*, que indica que um número não é um número válido. Veja agora, na prática, vários exemplos dessa conversão de *string* em números e os erros que podem ser cometidos.

Temos vários exemplos de conversão de *string* para números quando usamos operadores aritméticos. Teremos sempre um parágrafo e no final (linha 28) você pode ver que eu estou sempre escrevendo nesse parágrafo o valor de `x`.

Código 1 - 06_5 Strings para Numeros.html

```
1 <html>
2 <head>
3   <meta charset="UTF-8" />
4   <title>Programação Estruturada - Aula 06</title>
5 </head>
6 <body>
7   <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9   <h1>Convertendo strings em números</h1>
10  <p id="texto"></p>
11
12  <script>
13    x = "100" / "20";
14    //x = "100" * "20";
15    //x = "100" - "20";
16    //x = "100" % "20";
17
18    //x = 100 / "Javascript";
19    //isNaN(x);
```

```

20
21 //x = NaN + 10 ;
22 //x = NaN + "10" ;
23
24 //x = typeof NaN;
25
26 document.getElementById("texto").innerHTML = x;
27 </script>
28 </body>
29 </html>
30

```

Início com `x` recebendo o valor da divisão da *string* "100" pela *string* "20". Nesse momento, JavaScript tenta converter essas *strings* em números e, como elas são *strings* numéricas, ele vai fazer essa conversão e vai escrever na tela o resultado 5, que é o resultado da divisão de 100 por 20.

E o mesmo acontece se eu fizer "100" multiplicado por "20", essa conversão também acontece resultando na escrita do valor 2000 na tela. O mesmo acontece se eu fizer a subtração da *string* "100" pela *string* "20", resultando aí no no valor 80 sendo escrito na tela, e também se eu usar a operação de módulo que dá o resto da divisão da *string* "100" pela *string* "20", a conversão também acontece, e o resultado é 0; afinal de contas, 100 dividido por 20 é 5, e o resto é 0.

Um outro teste que eu queria mostrar para você é este: o que acontece se eu dividir o valor 100 pelo texto "JavaScript", a conversão não vai dar certo porque o texto "JavaScript" não é uma *string* numérica, e o que será escrito na tela é o valor `Not a Number`, então o valor `NaN` é escrito na tela porque nós não teremos um número em mãos.

Código 2 - 06_5 Strings para Numeros.html

```

1 <html>
2 <head>
3 <meta charset="UTF-8" />
4 <title>Programação Estruturada - Aula 06</title>
5 </head>
6 <body>
7 <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9 <h1>Convertendo strings em números</h1>

```

```
10 <p id="texto"></p>
11
12 <script>
13 //x = "100" / "20";
14 //x = "100" * "20";
15 //x = "100" - "20";
16 //x = "100" % "20";
17
18 x = 100 / "Javascript";
19 //isNaN(x);
20
21 //x = NaN + 10 ;
22 //x = NaN + "10" ;
23
24 //x = typeof NaN;
25
26 document.getElementById("texto").innerHTML = x;
27 </script>
28 </body>
29 </html>
30
```

Figura 1 - Erro na Conversão

Convertendo strings em números

NaN

E temos um operador chamado `isNaN` que simplesmente retorna `true` se o valor passado é um `NaN` e retorna `false`, caso contrário. E como `x` é `NaN` (linha 15), se a gente perguntar se ele é um `NaN`, eu vou de novo atribuir à variável `x`, e posso fazer isso, lembre lá das primeiras aulas que nós dissemos que JavaScript não é fortemente tipada, a mesma variável pode receber valores de tipos diferentes, então ela estará recebendo o valor booleano, que é a resposta dessa pergunta, se `x` da linha anterior é um `NaN`. E a resposta, como vimos, deveria ser `true` que é exatamente o que será escrito na tela.

Um outro exemplo acontece se eu somar o NaN, que representa que algo não é um número, sabendo-se que esse valor existe, o valor NaN existe, eu posso usá-lo conforme estou mostrando na linha 15.

Código 3 - 06_5 Strings para Numeros.html

```
1 <html>
2 <head>
3   <meta charset="UTF-8" />
4   <title>Programação Estruturada - Aula 06</title>
5 </head>
6 <body>
7   <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9   <h1>Convertendo strings em números</h1>
10  <p id="texto"></p>
11
12  <script>
13    //x = "100" / "20";
14    //x = "100" * "20";
15    //x = "100" - "20";
16    //x = "100" % "20";
17
18    //x = 100 / "Javascript";
19    //isNaN(x);
20
21    //x = NaN + 10 ;
22    x = NaN + "10" ;
23
24    //x = typeof NaN;
25
26    document.getElementById("texto").innerHTML = x;
27  </script>
28 </body>
29 </html>
30
```

Estou somando um `Not a Number` com 10, obviamente se nós somamos algo que não é um número com 10, a nossa expectativa é que a resposta continue sendo um `Not a Number`, então qualquer operação aritmética que você fizer com `Not a Number`, a resposta vai ser o NaN. Seja ela o +, também se eu fizer a soma com a *string* "10", como nós temos um `Not a Number` do lado esquerdo. A conversão da *string* "10",

não acontece, e o que ele faz é concatenar o valor, ele transforma o `Not a Number` numa *string* que é um `NaN`, e concatena com essa *string* "10", resultando na tela que você pode ver a seguir, a *string* "NaN10" sendo impressa na tela.

Figura 2 - Convertendo `NaN` em *String*

Convertendo strings em números

NaN10

Por fim, apenas para que você possa ver qual de fato é o tipo desse valor numérico `Not a Number`, atribuí a `x` um `typeof` de `Not a Number` e pedi para imprimir isso na tela. Bem, pelo que você pode ver, de fato, `Not a Number` é um número, ok?

Figura 3 - Tipo de `NaN`

Convertendo strings em números

number