Programação Estruturada

Aula 07 - Operadores Javascript: Strings

Videoaula 03 - Transformando *Strings*

Videoaula 03 - Transformando *Strings*

Além de podermos transformar uma *string* usando os métodos `toUpperCase()` e `toLowerCase()`, como vimos em videoaulas anteriores, JavaScript também oferece métodos para substituir partes de uma *string*. O método `replace()` substitui a primeira ocorrência da *string* passada como primeiro parâmetro pela *string* passada como segundo parâmetro. Assim como nos métodos de extração de *strings* que vimos, o método `replace()` não altera a *string* à qual ele está sendo aplicado, mas retorna uma nova *string* como resultado.

Por exemplo, no slide podemos ver que, se aplicarmos esse método ao texto "ALÔ MUNDO LINDO", passando "LINDO" e "BELO" como parâmetros, faremos a substituição de "LINDO" por "BELO", resultando na *string* "ALÔ MUNDO BELO". Esse método diferencia letras maiúsculas e letras minúsculas. No entanto, é possível usá-lo sem que ele faça essa distinção. Além disso, também é possível substituir todas as ocorrências da primeira *string* pela segunda *string*. Vamos ver como podemos fazer isso?

Figura 1 - Alterando *Strings*

Neste exemplo, você verá várias formas possíveis para substituir ou transformar uma *string*. Tenho o texto "ALÔ MUNDO LINDO" e, novamente, a variável `resultado`, cuja definição sempre estará na linha 16, e o valor dessa variável será escrito continuamente na página HTML.

Código 1 - 07_6 Substituindo Strings.html

```
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Programação Estruturada - Aula 07</title>
  </head>
  <body>
    <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>

    <h1>Substituindo Strings</h1>

    <p id="saida"></p>

    <script>

    var texto = "ALÔ MUNDO LINDO";
    var resultado = texto.replace("LINDO","BELO");
    // Não altera a string original
    //var resultado = texto;

    // Substituição apenas da primeira ocorrência
    //var texto = "ALÔ MUNDO LINDO LINDO";
    //var resultado = texto.replace("LINDO","BELO");

    // Método é case sensitive
    //var texto = "ALÔ MUNDO LINDO";
    //var resultado = texto.replace("lindo","belo");

    // Expressões regulares para case insensitive
    //var texto = "ALÔ MUNDO LINDO";
    //var resultado = texto.replace(/lindo/i,"belo");

    // Expressões regulares para substituir todas as ocorrências
    //var texto = "ALÔ MUNDO LINDO LINDO";
    //var resultado = texto.replace(/LINDO/g,"BELO");
    //var resultado = texto.replace(/lindo/ig,"belo");

    document.getElementById("saida").innerHTML = resultado;

    </script>
  </body>
</html>

```

No primeiro exemplo, vou escrever na variável `resultado` a substituição, então eu vou aplicar o método `replace()` substituindo o texto "LINDO" pelo texto "BELO". Veja o que acontece página HTML. Observe que aparece "ALÔ MUNDO BELO", ou seja, aquele texto "LINDO" foi substituído pelo texto "BELO". Agora, note que ao fazer isso eu não vou substituir a string original, certo?

Se eu acessar a linha 18 e disser que o meu `resultado` é igual ao texto, você vai notar que, ao fazer isso, o que se mostra na tela é o texto "ALÔ MUNDO LINDO", que é o resultado original do texto. Por que isso acontece? Como eu falei, ao aplicar na linha 16 o `replace()` na variável `texto` eu não o alterei texto, essa variável é imutável, ela não muda o seu valor. Então, se eu a escrever novamente, na variável `resultado` terei o valor original dela, que é o texto "ALÔ MUNDO LINDO". Ok?

Vamos ver agora um outro exemplo, onde a gente vai substituir apenas a primeira ocorrência do texto. Tenho a variável `texto` "ALÔ MUNDO LINDO LINDO" e o meu resultado vai ser substituir o texto "LINDO" pela palavra "BELO", e o resultado ficará "ALÔ MUNDO BELO LINDO". Note que apenas a primeira ocorrência do texto "LINDO" foi substituída, a segunda ficou intocada.

Figura 2 - Alterando *Strings*

Além disso, analisando outro exemplo, veremos que esse método é sensível a letras maiúsculas e minúsculas, o que chamamos de *case sensitive*, em inglês. Utilizando o meu texto "ALÔ MUNDO LINDO" vou dar um `replace` do texto "lindo" pelo texto "belo", mas vamos supor que eu escrevi nesse trecho como eu fiz na linha 17, ou seja, usei letras minúsculas para indicar essa substituição.

Daí, ao carregar a página, veja que nada mudou no texto original, uma vez que esse método é sensível à diferença entre letras maiúsculas e minúsculas, logo, ele não encontrou a palavra "lindo" no texto "ALÔ MUNDO LINDO", onde todas as letras são maiúsculas. E, na verdade, qualquer letra que fique minúscula, vou alterar apenas a letra "o" como exemplo, continua também sem alteração, simplesmente porque ele não encontrou o texto "LINDO", onde todas as letras são maiúsculas (linhas 15-17).

Figura 3 - Alterando *Strings* (Minúsculas e Maiúsculas)

```
15      // Método é case sensitive
16      var texto = "ALÔ MUNDO LINDO";
17      var resultado = texto.replace("LINDo","belo");
```

Vejamos um outro exemplo com relação a expressões regulares: tenho o texto "ALÔ MUNDO LINDO", vou fazer a substituição do padrão `/lindo/i` por "BELO". Esse padrão mostra o texto "lindo", porém, com essa letra i que vem depois dessa definição, o padrão se tornou insensível à diferença entre maiúsculas e minúsculas, ou seja, se ele encontrar o texto "LINDO", independentemente dessas letras estarem maiúsculas ou minúsculas, ele vai substituir pelo texto "belo", e esse texto aparecerá com letras minúsculas. Ao recarregar a página, veja que o texto "LINDO" foi substituido pelo texto "belo" com todas as letras minúsculas, mesmo estando no padrão escrito com letras minúsculas, porque eu indiquei com essa letra i que esse padrão não é sensível à diferença entre maiúsculas e minúsculas, ok?

Figura 4 - Resultado da Substituição do Padrão `/lindo/i` por "`belo`"

Novamente, vou deixar uma referência para você, caso queira estudar mais sobre expressões regulares. Por fim, vou deixar uma dica para substituição: aqui eu deixei "ALÔ MUNDO LINDO LINDO" e agora, no resultado, estou substituindo "LINDO" por "BELO". Vou inserir outra letra, o g de global. Então, ele vai procurar o "LINDO" em todo

o texto que eu fizer o `replace`, mas como eu não estou usando a letra i, o "LINDO" é maiúsculo, e ele vai substituir pelo texto "BELO", o que vai acontecer em todas as ocorrências porque eu estou usando a letra g. Eu tenho duas ocorrências de "BELO", que é exatamente o resultado de substituir essas duas ocorrências de "LINDO" na linha 16.

Figura 5 - Padrão para substituir todas as ocorrências

```
15    // Expressões regulares para substituir todas as ocorrências
16    var texto = "ALÔ MUNDO LINDO LINDO";
17    var resultado = texto.replace(/LINDO/g,"BELO");
18    //var resultado = texto.replace(/lindo/ig,"belo");
```

Figura 6 - Resultado da Substituição do Padrão `/LINDO/g` por `"BELO"`

Substituindo Strings

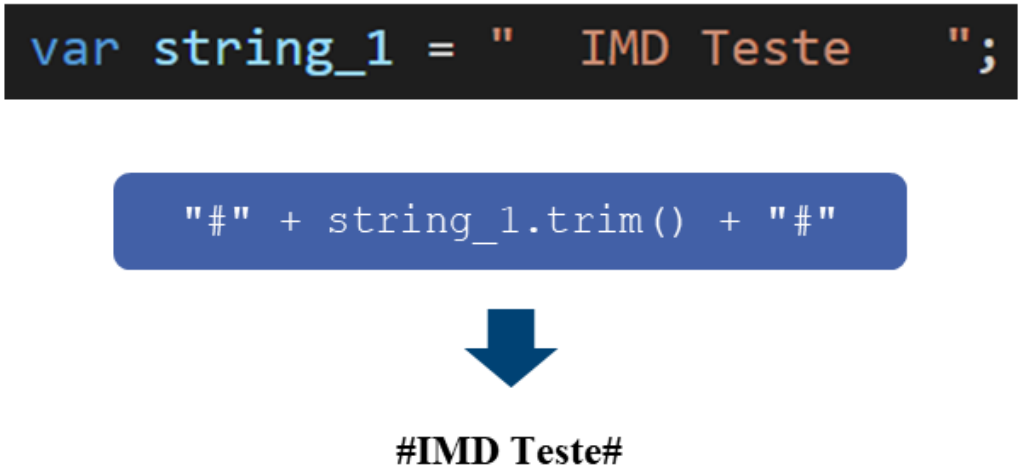
ALÔ MUNDO BELO BELO

Mas, eu posso fazer além disso, eu posso dizer: "Não, eu não quero diferenciar entre maiúsculas e minúsculas", então vou usar aquele padrão que acabei de mostrar, usando o "lindo" todo minúsculo. Coloco a letra i pra dizer que é insensível à diferença entre maiúsculas e minúsculas, e coloco a letra g pra dizer que eu vou substituir todas as ocorrências. E aí, se eu fizer isso, o esperado é que essas duas ocorrências do texto "LINDO", mesmo que sejam uma ocorrência em maiúsculo e outra em minúsculo, sejam substituídas pelo texto "belo", o que vemos como resultado na tela. Ok?

Figura 7 - Resultado da Substituição do Padrão `/lindo/ig` por `"belo"`

Outra transformação comum que podemos fazer em *strings* é a remoção dos espaços em branco no início e no final da *string*. O método `trim()` faz exatamente isso. No slide, colocamos o jogo da velha no início e no final do resultado para que você possa ver que, de fato, o método `trim()` removeu todos os espaços em branco do início e do final da `string_1`. Note, porém, que o espaço em branco do meio da *string*, entre as palavras "IMD" e "Teste", não foi removido.

Figura 8 - Removendo espaços em branco



Infelizmente, o método `trim()` não é suportado no Internet Explorer 8 ou em versões anteriores. Se você precisar oferecer suporte ao Internet Explorer 8 e versões anteriores, poderá usar `replace()` com uma expressão regular. Veja como no exemplo a seguir.

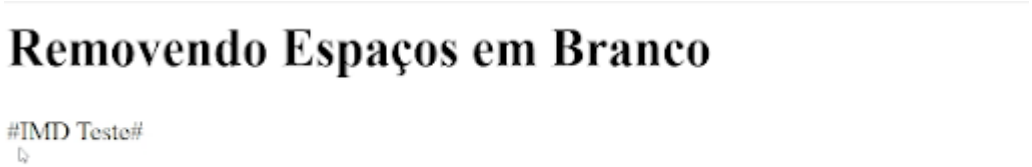
Neste exemplo, temos uma página HTML simples, com um campo de saída (linha 11), e vou usar um JavaScript externo, o que já vínhamos fazendo em outras aulas, refaço esse procedimento e você irá entender o porquê daqui a pouco. Nesse JavaScript, tenho a definição de uma variável `string_1 = "  IMD Teste  "`, mas, note que no começo e no final, eu deixei vários espaços em branco. E aí o meu resultado, que é usado para escrever o resultado no HTML, é simplesmente eu aplicar aqui o método `trim()` nessa *string*. E note que eu vou colocar o jogo da velha antes e depois do valor dessa variável `resultado`.

Figura 9 - Exemplo de remoção de espaços em branco

```
1  var string_1 = "  IMD Teste   ";
2
3  var resultado = string_1.trim();
4  document.getElementById("saida").innerHTML = "#"+ resultado+"#";
5
```

Veja que, de fato, todos os espaços em branco, tanto no começo como no final da *string*, foram removidos, ok?

Figura 10 - Resultado da remoção dos espaços em branco



Na verdade, essa é a maneira mais comum de se fazer isso, porém, o método `trim` não funciona no Internet Explorer 8 ou em versões anteriores, então eu vou deixar uma dica aqui pra você: apresento um método que redefine o método `trim` e usa o método `replace` (linha 6), passando um padrão, e você pode, simplesmente, copiar e colar esse trecho das linhas 4 a 8. Com esse código adicional, o que vai acontecer do ponto de vista de quem está usando o Google Chrome, é que não haverá diferença no resultado, vamos remover os espaços em branco do mesmo jeito, porém, esse código também vai funcionar na versão Internet Explorer 8 ou versões anteriores.

Código 2 - 07_7 Removendo Espaços.html

```html
<html>
   <head>
      <meta charset="UTF-8" />
      <title>Programação Estruturada - Aula 07</title>
   </head>
   <body>
      <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>

      <h1>Removendo Espaços em Branco</h1>

      <p id="saida"></p>
      <script src='script.js'></script>
   </body>
</html>

```

```javascript
var string_1 = "  IMD Teste   ";

// O código abaixo só é necessário para IE 8 ou anteriores
if (!String.prototype.trim) {
   String.prototype.trim = function () {
     return this.replace(/^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g, '');
   };
}

var resultado = string_1.trim();
document.getElementById("saida").innerHTML = "#"+ resultado+"#";


```

Com esse exemplo, concluímos esta aula sobre *strings* em Javacript, onde você aprendeu a definir, procurar e transformar *strings* em JavaScript. Não esqueça de fazer as atividades propostas para que você possa colocar, sozinho, esse conhecimento em prática. Até a próxima aula. Tchau, tchau!!!