

Programação Estruturada

Aula 13 - Recursão: Introdução, Números e Strings

Videoaula 03: Recursão e Strings



Videoaula 03: Recursão e Strings

Olá, tudo bem? Agora que você já entende o conceito de recursão e já sabe aplicá-lo aos números, nesta videoaula usaremos esse conceito para um outro tipo de dado, as *strings*.

Será que conseguimos inverter uma *string* usando recursão? A resposta é sim, apesar de não ser a maneira mais eficiente de inverter uma *string* em JavaScript. De qualquer maneira, veremos como fazer isso.

A nossa solução é que, recursivamente, peguemos a primeira letra do texto e concatenemos ela ao final da inversão do resto desse texto. Fazemos isso até que ele fique vazio. O **caso base** será a *string* vazia. Nesse caso, simplesmente, a retornaremos. No exemplo, não foi utilizada a **condição de terminação** porque o **caso base** já cumpre essa função. Não existem *strings* com caracteres "negativos". Portanto, desde que sejam passadas *strings* para a nossa função, está tudo OK.

Veja esse exemplo no slide (Figura 1), onde é calculada a inversão do texto "abcde". O resultado da função `inverter("abcde")` concatena o resultado de `inverter("bcde")` com a primeira letra, "a". Para calcular `inverter("bcde")`, precisamos concatenar `inverter("cde")` com a primeira letra, "b". Continuamos chamando recursivamente a função `inverter` até a chamada `inverter("")`, que está sublinhada. Nesse momento, chegamos ao **caso base** e simplesmente retornamos "". Cada chamada retorna corretamente o seu texto invertido. Você pode comprovar isso observando suas diferentes cores.

Figura 1 - Invertendo *strings*

```
inverter("abcde")
= inverter("bcde") + "a"
= (inverter("cde") + "b") + "a"
= ((inverter("de") + "c") + "b") + "a"
= (((inverter("e") + "d") + "c") + "b") + "a"
= ((((inverter("") + "e") + "d") + "c") + "b") + "a"
= (((("e" + "d") + "c") + "b") + "a"
= (("ed" + "c") + "b") + "a"
= ("edc" + "b") + "a"
= "edcb" + "a"
= "edcba"
```

Tudo bem até aqui? Se você entendeu a nossa solução, conheça agora a sua implementação. Fique à vontade para editar o Código 1 e realizar a quantidade de testes que desejar.

Código 1 - 13_6 Inverte Palavra.html e 13_6 Inverte Palavra.js

```
1 <html >
2   <head>
3     <meta charset="UTF-8" />
4     <title>Programação Estruturada - Aula 13</title>
5   </head>
6   <body>
7     <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9     <h1>Invertendo Palavras</h1>
10
11     Texto: <input type="text" id="texto" value="">
12     <button onclick="exibir()">Inverter</button>
13     <br>
14     <p id="resultado"></p>
15
16     <script src="script.js"></script>
17   </body>
18 </html>
19
```

```
1 function exibir() {
2   var texto = document.getElementById("texto").value;
3   var resultado = 'Inverso de "' + texto + '" = "' + inverter(texto) + '"';
4   document.getElementById("resultado").innerHTML = resultado;
5 }
6
```

```

7 function inverter(s) {
8   var retorno = "";
9
10  // CASO BASE
11  if (s === "") {
12    retorno = "";
13    // RECURSÃO
14  } else {
15    retorno = inverter(s.substr(1)) + s[0];
16  }
17
18  return retorno;
19 }
20
21

```

Nesse exemplo, o HTML é bastante simples e, de certa maneira, similar aos que já vimos. A diferença é que ao invés de termos um campo numérico, temos um de texto (entrada), e quando clicar no botão "Inverter" (linha 12) chamamos a função `exibir`, que vai calcular a *string* a ser exibida e vai escrevê-la nesse parágrafo de "resultado" (linha 14). O foco, novamente, é no JavaScript e a função `exibir` (linhas 1-5) preserva praticamente o mesmo raciocínio dos exemplos anteriores. Dessa vez, pegamos um texto ao invés do número, esse texto gera a variável `resultado` e altera o valor do parágrafo na página HTML, e esse resultado é o texto "Inverso de ", escrevemos o texto original e dizemos que é igual, e chamamos a função `inverter` desse texto.

Focaremos na função `inverter` (Código 1), uma função recursiva que vai inverter a palavra. Essa função tem um **caso base**, como já dito, que é a *string* vazia, e quando ela é vazia temos `retorno` vazio, caso contrário, chamamos a função `inverter` e passamos a *substring*, que tira o primeiro caractere. Então, tiramos o primeiro caractere e mandamos `inverter` essa *string* e jogamos o primeiro caractere para o final dela. Com isso, iremos, de fato, estar invertendo porque pegamos esses elementos e o primeiro caractere será jogado para o final, e vai inverter o resto. Assim, teremos o final com o resto da *string* sendo invertida e estaremos sempre realizando a mesma ação de pegar o primeiro e jogar para o final.

E, como resultado, o que temos é essa página HTML (Figura 2), e se escrevermos "abcde" e clicarmos em inverter ele, de fato, mostra que o inverso de "abcde" é "edcba". Então ele está invertendo a palavra. Qual foi a estratégia utilizada? Uma estratégia recursiva que estará sempre pegando o primeiro elemento e jogando-o para o final, e invertendo o resto, tudo bem?

Figura 2 - Saída da Página HTML



Concluimos neste instante a aula sobre Recursão. Na próxima aula, vamos aprofundar esses conhecimentos aplicando-os a Arrays e Matrizes, e faremos juntos alguns exercícios. Até lá!!!