

Programa o Estruturada

Aula 01 - Introdu o   Programa o Estruturada e JavaScript

Videoaula 04 - Instru es JavaScript



Videoaula 04 - Instruções JavaScript

Nesta videoaula, vamos apresentar as instruções JavaScript. Os programas JavaScript são definidos através de uma sequência de instruções separadas por ponto-e-vírgula (;). Na verdade, você pode encontrar exemplos sem o ponto-e-vírgula, isto porque terminar instruções com ponto-e-vírgula não é obrigatório, mas recomendamos sua utilização nesse caso.

As instruções JavaScript geralmente começam com uma palavra-chave para identificar a ação JavaScript a ser executada, por exemplo, você já aprendeu a palavra-chave *function* para definir nossas funções.

Ao longo desta disciplina, você aprenderá várias palavras-chave, que são palavras reservadas e não podem ser usadas como nome para variáveis. Observe alguns exemplos de instruções básicas JavaScript.

Se quiser executar um comando após o outro, é preciso colocar o ponto-e-vírgula. Ao longo do curso, você verá detalhes sobre a declaração de variáveis e os operadores que veremos nesse exemplo. Por enquanto, vamos apenas focar em escrever uma sequência de instruções.

No exemplo, tem três variáveis: *x*, *y*, *z*, atribuindo o valor 10 a *x*, *y* recebe o valor 20 e o nosso *z* vai receber aqui o valor *x* + *y*. Note que a linha foi quebrada no final, após o ponto-e-vírgula, mas isso não é obrigatório. Esses comandos poderiam aparecer todos em uma única linha, como demonstra a linha 18 no Código 1, mas mantivemos essa versão com quebra de linhas para melhor compreensão do código.

Código 1 - 01_7 Instruções JavaScript.html

```
1 <html>
2   <head>
3     <meta charset='UTF-8' />
4     <title>Programação Estruturada - Aula 01</title>
```

```

5 </head>
6 <body>
7   <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9   <h1>Instruções JavaScript</h1>
10
11  <h2>Sequência de Instruções</h2>
12  <p id='meu_texto_1'></p>
13  <script>
14    var x,y,z;
15    x = 10;
16    y = 20;
17    z = x + y;
18    <!-- var x,y,z; x = 10; y = 20; z = x + y; -->
19
20    document.getElementById("meu_texto_1").innerHTML = z;
21  </script>
22 </body>
23 </html>
24

```

Ao final, tem-se o último comando JavaScript. Ele faz algo que já fizemos anteriormente, que é mudar o conteúdo de um campo. Assim, o "meu_texto_1", agora vai receber exatamente o valor z, ou seja, o meu texto no HTML começa vazio, porém, o meu JavaScript será executado e, ao final, vai alterar o valor desse campo pra z. Com isso, esperamos que o valor 30 apareça, afinal $10 + 20$ é igual a 30, e é exatamente o que aparece na página HTML (Figura 1).

Figura 1 - Mudança de conteúdo de um campo



No segundo exemplo, você verá que os espaços em branco dentro de um comando são simplesmente ignorados. Temos na linha 14 a declaração da variável `nome`, que recebe o valor "Marcel", e na linha 15, a alteração do conteúdo do "meu_texto_2", que tem o valor "Conteúdo inicial", para "Marcel". Então, ao carregar essa página HTML, esse JavaScript vai ser executado e o que vai aparecer é o nome "Marcel" (Figura 2).

Figura 2 - Ignorando Espaços em Branco



Note que o comando independe da quantidade de espaços inseridos (Código 2) entre o nome e o igual (linha 14), e isso acontece porque na interpretação JavaScript não há alteração no comportamento, simplesmente porque nos comandos os espaços em branco são ignorados.

Código 2 - 01_8 Instruções JavaScript.html

```
1 <html>
2   <head>
3     <meta charset='UTF-8' />
4     <title>Programação Estruturada - Aula 01</title>
5   </head>
6   <body>
7     <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9     <h1>Instruções JavaScript</h1>
10
```

```

11 <h2>Ignorando espaços em branco</h2>
12 <p id='meu_texto_2'>Conteúdo inicial</p>
13 <script>
14     var nome =                "Marcel";
15     document.getElementById("meu_texto_2").innerHTML = nome;
16 </script>
17
18 </body>
19 </html>
20

```

Por fim, veja outro exemplo que é a inserção de várias instruções em um mesmo bloco. Neste exemplo, declaramos a função chamada `minhaFuncao`, delimitando o bloco de comandos dessa função com "abrindo chave" e "fechando chave" (`{ }`). Nesse HTML temos dois textos, "meu_texto_3" e "meu_texto_4", cada um em parágrafos diferentes (linhas 14 e 15). Dentro do bloco da função, temos as alterações desses dois textos para "Olá aluno" e "Está gostando da aula" (Código 3). Portanto, podemos observar que o bloco da função está sendo delimitado pelas chaves.

Código 3 - 01_9 Instruções JavaScript.html

```

1 <html>
2 <head>
3   <meta charset='UTF-8' />
4   <title>Programação Estruturada - Aula 01</title>
5 </head>
6 <body>
7   <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9   <h1>Instruções JavaScript</h1>
10
11  <h2>Bloco de Instruções</h2>
12  <button type="button" onclick="minhaFuncao()">Clique Aqui</button>
13
14  <p id='meu_texto_3'></p>
15  <p id='meu_texto_4'></p>
16
17  <script>
18  function minhaFuncao() {
19    document.getElementById("meu_texto_3").innerHTML = "Olá Aluno!";
20    document.getElementById("meu_texto_4").innerHTML = "Está gostando da aula?";
21  }
22  </script>

```

```
23 </body>
24 </html>
25
```

Ao clicar no botão, invocamos a função no `onclick`, que executou exatamente aquele bloco que corresponde a ela, conforme mostra a Figura 3.

Figura 3 - Inserção de várias instruções em um bloco



Nesse quarto exemplo, vou mostrar como inserir comentários no JavaScript, algo extremamente importante em um código-fonte, seja ele em JavaScript, seja ele HTML, ou em qualquer linguagem de programação. Isso porque os comentários ajudam no entendimento do trecho do código-fonte que está sendo comentado, tanto por outras pessoas que vão trabalhar com o código-fonte que você próprio programou ou você mesmo, algum tempo depois, se você voltar para aquele código e tiver dificuldade de entender alguns trechos de códigos mais complicados.

No Código 4, veja que os comentários podem ser feitos em uma linha ou em várias linhas. Se você quer fazer um comentário em apenas uma linha, basta usar o `//`, na linha onde se deseja fazer o comentário (linha 15).

Código 4 - 01_10 Instruções JavaScript.html

```
1 <html>
2   <head>
3     <meta charset='UTF-8' />
4     <title>Programação Estruturada - Aula 01</title>
5   </head>
6   <body>
7     <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9     <h1>Instruções JavaScript</h1>
10
11    <h2>Comentários</h2>
12
13    <p id='meu_texto_5'></p>
14    <script>
15      // Comentário de uma linha
16      /*
17      Comentário de várias linhas
18      */
19      //document.getElementById("meu_texto_5").innerHTML = 10;
20    </script>
21
22  </body>
23 </html>
24
```

Agora, se você quiser comentar mais de uma linha, o recomendado é usar `/*` e o `*/`. Tudo o que estiver entre o `/*` e o `*/`, vai ser interpretado como comentário, conforme demonstrado no Código 5.

Código 5 - 01_10 Instruções JavaScript.html

```
1 <html>
2   <head>
3     <meta charset='UTF-8' />
4     <title>Programação Estruturada - Aula 01</title>
5   </head>
6   <body>
7     <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9     <h1>Instruções JavaScript</h1>
10
11    <h2>Comentários</h2>
12
```

```

13 <p id='meu_texto_5'></p>
14 <script>
15     // Comentário de uma linha
16     /*
17     Comentário de várias linhas
18     ASdasds
19     asdasdasd
20     asdasdas
21     */
22     //document.getElementById("meu_texto_5").innerHTML = 10;
23 </script>
24
25 </body>
26 </html>
27

```

Percebeu que o resultado na página apareceu em branco? Isso aconteceu porque o comando que eu tinha dado está comentado, ou seja, você pode querer em algum momento do desenvolvimento fazer um comentário de um comando para ver se algum comportamento muda e qual é o efeito daquele comando.

Para isso, você pode comentar a linha que tem esse comando (linha 19). Veja que se for retirado o //, ao recarregar a página, o comando vai ser efetivamente executado e apresenta-se o resultado esperado (Figura 4).

Código 6 - 01_10 Instruções JavaScript.html

```

1 <html>
2   <head>
3     <meta charset='UTF-8' />
4     <title>Programação Estruturada - Aula 01</title>
5   </head>
6   <body>
7     <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9     <h1>Instruções JavaScript</h1>
10
11    <h2>Comentários</h2>
12
13    <p id='meu_texto_5'></p>
14    <script>
15      // Comentário de uma linha
16      /*
17      Comentário de várias linhas
18      */

```

```
19     document.getElementById("meu_texto_5").innerHTML = 10;
20     </script>
21
22     </body>
23 </html>
24
```

Figura 4 - Retirada de comentário

Programação Estruturada - Aula 01

Instruções JavaScript

Comentários

10

Agora, vou mostrar como você pode fornecer valores para o código JavaScript, prática que será muito comum durante esta disciplina. Os programas muitas vezes exigirão valores de entrada para que algum algoritmo seja executado gerando o seu resultado, ou seja, sua saída.

Nesse exemplo, temos os títulos e o campo de entrada. Esse campo usa a tag `input` e definiremos o tipo dessa entrada como `text`. Mas, temos outros tipos para essas entradas, e você conhecerá um deles na atividade que propus.

Vamos identificar essa entrada com um identificador chamado `"meu_texto"`, e temos aqui novamente o `button`, onde no `onclick` nós vamos chamar a função `minhaOutraFuncao` (Código 7). Essa `minhaOutraFuncao` está no trecho JavaScript que

eu coloquei no final do corpo HTML e ela declara uma variável `x` e atribui para essa variável o valor do elemento `meu_texto`.

Código 7 - 01_11 Instruções JavaScript.html

```
1 <html>
2   <head>
3     <meta charset='UTF-8' />
4     <title>Programação Estruturada - Aula 01</title>
5   </head>
6   <body>
7     <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9     <h1>Instruções JavaScript</h1>
10
11    <h2>Fornecendo Valores para o JavaScript</h2>
12
13    Entrada: <input type="text" id="meuTexto" value="">
14    <button onclick="minhaOutraFuncao()">Clique Aqui</button>
15    <p id="meu_texto_6"></p>
16
17    <script>
18    function minhaOutraFuncao() {
19      var x = document.getElementById("meuTexto").value;
20      document.getElementById("meu_texto_6").innerHTML = x;
21    }
22  </script>
23 </body>
24 </html>
25
```

O objetivo dela é pegar o valor do elemento `meu_texto`, declarado na linha 19, que é o campo de texto de entrada declarado na linha 13, e vai atribuir esse valor ao campo do texto que eu nomeei como `meu_texto_6` (linha 20).

Então, o comportamento é o valor que for colocado no campo de texto que aparecerá escrito na página. Vamos supor que eu escreva o famoso "Alô Mundo", ao clicar em , o valor será escrito na página. E se eu alterar, por exemplo, para "Marcel", esse valor será alterado, ou seja, pegamos um valor, uma entrada, fizemos um processamento JavaScript e fornecendo saídas (Figuras 5 e 6). Faremos isso muitas vezes ao longo desta disciplina!

Figura 5 - Demonstração do comportamento JavaScript



Figura 6 - Demonstração do comportamento JavaScript



E aí, gostou dos exemplos? Agora, você vai praticar um pouco do que foi visto nesta aula. Vamos lá!?

Chegamos ao final de nossa primeira aula! Nela, fizemos uma introdução à linguagem JavaScript, você conheceu o seu histórico e suas principais características, e também entendeu como o código JavaScript pode ser incorporado à página HTML e à estrutura de um programa básico em JavaScript.

Para isso, você viu vários exemplos de páginas HTML bem simples, mas que já permitem interação com o usuário através de códigos JavaScript. Esperamos que, com esta aula, você tenha construído o conhecimento necessário para iniciar o desenvolvimento de programas básicos em JavaScript.

Sugerimos que, sempre que possível, você pesquise códigos disponíveis na internet, analise-os e verifique os seus comportamentos ao serem executados. Complemente seus estudos através da leitura de outros materiais, sejam eles livros ou artigos, físicos ou digitais.

Na próxima aula, você conhecerá mais detalhes sobre a linguagem de programação JavaScript.

Um abraço e até lá!