

Programação Estruturada

Aula 03 - Variáveis e Constantes

Videoaula 04 - Atribuições

Videoaula 04 - Atribuições

Outra construção importante que também já usamos até aqui é a atribuição. Lembre-se que o sinal `=` é uma atribuição e não um operador de comparação, como em algumas outras linguagens. O operador de comparação em JavaScript é o `==`. Dessa forma, o comando `x = 10` atribui o valor `10` à variável `x` e o comando `y = (10 == 20)`, atribui o valor `false` à variável `y`, visto que `10` é diferente de `20`.

Além do igual, temos alguns operadores de atribuição especiais que visam reduzir o tamanho das expressões. Por exemplo, para os operadores aritméticos, temos:

- `a += b` para a atribuição de `a + b` à variável `a`
- `a -= b` para a atribuição de `a - b` à variável `a`
- `a *= b` para a atribuição de `a * b` à variável `a`
- `a /= b` para a atribuição de `a / b` à variável `a`
- `a %= b` para a atribuição de `a % b` à variável `a`

Vamos ver essas expressões sendo usadas na prática?

Neste exemplo, vamos fazer algumas operações usando as atribuições especiais. Nas linhas 16 e 17, temos a declaração de duas variáveis – `a` e `b` – a variável `a` recebendo o valor `20` e `b` recebendo o valor `10`, e vamos atribuir à variável `a` o valor de `a + b`. Faremos isso usando a atribuição especial `+=`.

Código 1 - 03_7 Assignments.html

```
1 <html>
2
3 <head>
4   <meta charset="UTF-8" />
5   <title>Programação Estruturada - Aula 03</title>
```

```
6 </head>
7
8 <body>
9   <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
10
11   <h1>Atribuições Especiais</h1>
12
13   <p id="texto"></p>
14
15   <script>
16     var a = 20;
17     var b = 10;
18
19     a += b;
20     //a -= b;
21     //a *= b;
22     //a /= b;
23     //a %= b;
24
25     document.getElementById("texto").innerHTML = a;
26   </script>
27
28 </body>
29
30 </html>
31
```

Então, na linha 25, vamos novamente alterar o valor do campo texto usando exatamente o valor dessa variável `a`. Veja como estou atribuindo à variável `a` o valor de `a + b`, usando uma atribuição especial. O valor que esperamos que seja exibido é 30, resultado de 20 mais 10, exatamente o valor exibido (Figura 1).

Atribuições Especiais

30

Vamos agora fazer a atribuição na linha 19, `a -= b` (Figura 2), que vai atribuir à variável `a` o valor de `a-b`, ou seja, 20 menos 10, que resultará na exibição do valor 10 (Figura 3).

Figura 2 - Conteúdo do arquivo 03_7 Assignments.html

```
1 a -= b;  
2 //a *= b;  
3 //a /= b;  
4 //a %= b;  
5
```

Figura 3 - Atribuições especiais (subtração)

Atribuições Especiais

10

Vamos agora fazer atribuição à variável a do valor de a vezes b , usando a atribuição $*=$ e teremos a expectativa de que o valor exibido seja 200, que é 20 vezes 10, cujo resultado está sendo exibido na tela (Figura 4).

Figura 4 - Atribuições especiais (multiplicação)

Atribuições Especiais

200

Vamos usar a atribuição especial de divisão, que vai atribuir à variável a o valor de a dividido por b ($a /= b$), ou seja, vamos atribuir à variável a o valor de 20 dividido por 10 e, como vemos (Figura 5), o valor exibido na página HTML é exatamente o 2, que é o

resultado de 20 dividido por 10.

Figura 5 - Atribuições especiais (divisão)

Atribuições Especiais

2

Por fim, vamos usar o operador de porcento, que vai resultar no resto da divisão de 20 por 10, ou seja, estamos atribuindo à variável a o valor do resto da divisão de a dividido por b , ou seja, a vai receber o valor de $a \% b$. E o resto da divisão de 20 por 10, como sabemos, é 0 que é exatamente o valor que está sendo exibido na tela (Figura 6).

Figura 6 - Atribuições especiais (porcentagem)

Atribuições Especiais

0

Então, podemos usar essas atribuições especiais ao longo do código normalmente.

Existem outras formas desse tipo de atribuição com outros tipos de dados, como, por exemplo, os booleanos. Mas vamos deixar para apresentá-los em outras aulas ainda nesta disciplina. Vamos agora fazer alguns exercícios?



Atenção

A partir desta aula, vamos estruturar melhor os nossos arquivos de exercícios. Todos os exercícios serão baseados em dois arquivos: um arquivo HTML e um arquivo JavaScript. Para respondê-los, será necessário apenas alterar o arquivo JavaScript. Todos os nomes dos campos de entrada e saída serão fornecidos no enunciado da questão ou dentro do arquivo JavaScript fornecido na questão.

Para verificar se a sua resposta está correta, você precisará carregar o arquivo HTML em um navegador. Vamos fazer um exemplo juntos para que você possa entender melhor?



Atividade

1. Complete o corpo da função `soma_e_escreve` do arquivo `Atividade_03_1.js`, de forma que essa função escreva no parágrafo de identificador "saida" a soma dos dois valores dados como entrada.

- HTML: [Atividade 03 1.html](#)
- JS: [Atividade 03 1.js](#)

Bom, só pra ilustrar como é que vai funcionar a estruturação dos arquivos. Foram disponibilizados para você dois arquivos, o `Atividade_03_1.html` e o `Atividade_03_1.js`, que é onde vai ficar o código JavaScript.

Nesta atividade, você irá escrever uma função JavaScript, o corpo da função JavaScript que vai receber dois valores, x e y , que vêm do HTML, e, ao clicar no botão OK, você irá escrever na tela o valor da soma de x com y (Figura 6).

Figura 7 - Atividade 1

Exercício

Atividade 03-1

x: y:

Então, podemos observar como está a estruturação da página (Código 2). Você vai receber esse HTML, que tem dois campos, "entrada1" e "entrada2" (linha 13 e 14), um botão, que o `onclick` dele chama a função `soma_e_escreve` (linha 15), e um parágrafo ou campo de "saída" (linha 16).

Código 2 - Atividade_03_1.html

```
1 <html>
2
3 <head>
4   <meta charset="UTF-8" />
5   <title>Programação Estruturada - Aula 03</title>
6 </head>
7
8 <body>
9   <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
10
11   <h1>Exercício</h1>
12
13   <h2>Atividade 03-1</h2>
14
15   x: <input type="number" id="entrada1" value="">
16   y: <input type="number" id="entrada2" value="">
17   <button onclick="soma_e_escreve()">OK</button>
18   <p id="saida"></p>
19
20   <script src="Atividade_03_1.js"></script>
```

```
21
22 </body>
23
24 </html>
25
```

Temos a `entrada1`, `entrada2`, `saida`, e, no final (linha 18), estamos usando aquela estruturação de um código JavaScript em um arquivo externo que precisa estar exatamente na mesma pasta da página HTML (Figura 7).

Figura 8 - Estruturação dos arquivos

Nome	Data de modificação	Tipo	Tamanho
Atividade_03_1	15/05/2019 18:54	Arquivo HTML	1 KB
Atividade_03_1	28/05/2019 11:02	Arquivo JavaScript	1 KB

Se voltarmos para o código JavaScript, vemos o arquivo `js` (Código 3), onde temos a declaração da função `soma_e_escreve` (linha 1). O arquivo que você irá receber já estará com essa estrutura, com a leitura dos valores de `x` e de `y`, da `entrada1` e da `entrada2` (linhas 2 e 3), e com a escrita, usando o valor de uma variável `z`, no campo com o identificador `saida` (linha 7). O seu trabalho é simplesmente escrever um código que declare uma variável `z` (linha 5) e que atribua a essa variável `z` a soma de `x` mais `y`.

Código 3 - Atividade_03_1.js

```
1 function soma_e_escreve() {
2   var x = Number(document.getElementById("entrada1").value);
3   var y = Number(document.getElementById("entrada2").value);
4
5   //Complete seu código aqui
6
7   document.getElementById("saida").innerHTML = z;
8 }
9
```

Então, a partir de agora, vamos sempre ter essa estrutura de páginas HTML usando arquivos JavaScripts externos à página HTML. Ok?

Boa tarefa!

Agora que você entendeu melhor a estrutura dos arquivos dos exercícios, chegou a hora de praticar o que aprendeu nesta aula. Além da Atividade 1 que apresentei, há mais seis atividades a fazer. Bom exercício. Tchau!!!