



Manipulando Elementos do DOM

Nesta aula, vamos escrever programas em JavaScript com entrada, processamento e saída de resultados com uma estrutura sequencial. Em seguida, vamos entender as formas de seleção simples, composta e encadeada para desenvolver a estrutura de seleção em JavaScript, usando o elemento window do DOM.

Entrada, Processamento e Saída de Dados em JavaScript

O objetivo de um programa é escrever um código no computador para resolver problemas. Para isso, é preciso que o programa tenha a entrada de dados, para que sejam processados e então os resultados apresentados para o usuário. A estrutura de um programa em JavaScript é como segue. Utilizamos `//` para comentar uma linha e `/* ... */` para comentar várias linhas.

Por exemplo, a declaração e comentários num programa JavaScript segue o seguinte código:

```
1 <script>
2 //comentário da linha
3 /*comentário em bloco*/
4
5 // declaração no momento da atribuição
6 a = 5;           //inteiro int
7 b = 1.7;         //real double
8 x = 'x';         //caracter char
9 frase = "frase"; //conjunto de caracter String
```

Fonte: Autoral

```
10: x = true;           //booleano boolean
11
12 //comandos
13 </script>
```

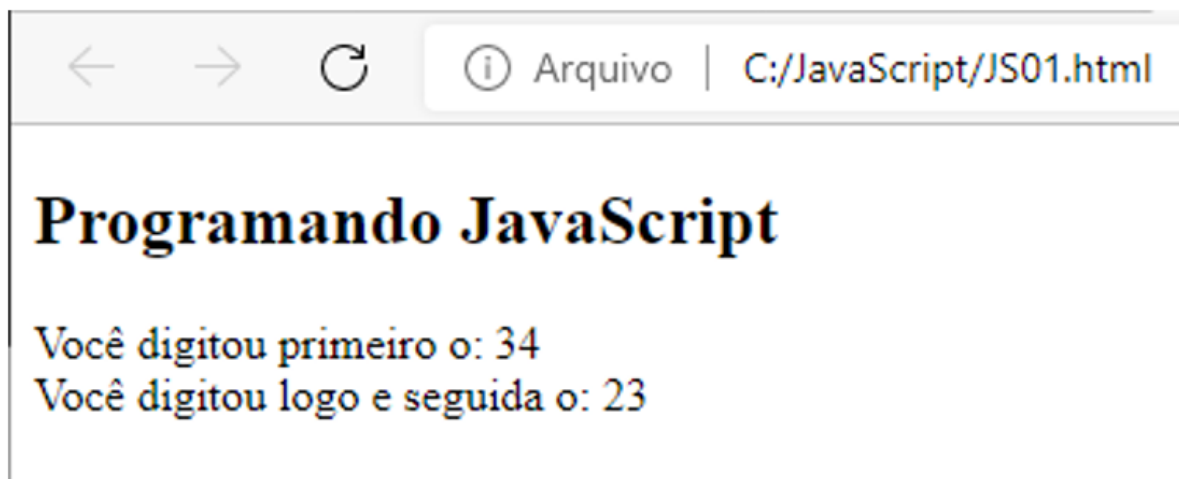


Para exemplificar a entrada de dados e a saída de informações em JavaScript, vamos desenvolver um programa JavaScript que recebe dois inteiros e mostra ambos os números.

```
1 <script>
2
3 a = window.prompt("digite um número inteiro: ");
4 b = window.prompt("digite outro número inteiro: ");
5
6 document.write("Você digitou primeiro o: " + a + "<br>");
7 document.write("Você digitou logo e seguida o: " + b);
8
9 </script>
```

Fonte: Autoral

Após a compilação e execução do programa JavaScript, temos:



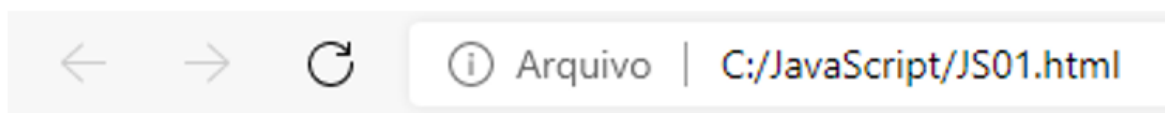
Fonte: Autoral

Estrutura Sequencial em JavaScript

A estrutura sequencial é um programa JavaScript que possui entrada de dados, processamento desses dados em resultado e saída desses resultados. Por exemplo, desenvolva um programa JavaScript que recebe dois valores inteiros e mostra a soma destes dois números.

```
1 <script>
2
3 n1 = parseInt(window.prompt("digite um número inteiro: "));
4 n2 = parseInt(window.prompt("digite outro número inteiro: "));
5
6 soma = n1 + n2;
7
8 document.write("A soma dos dois valores digitados é: " + soma + "<br>");
9
10 </script>
```


O resultado da execução deste programa JavaScript ficará:



Programando JavaScript

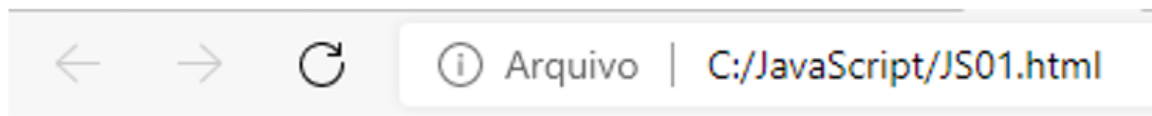
A soma dos dois valores digitados é: 46

Outro exemplo de estrutura sequencial para desenvolver um programa JavaScript que recebe duas notas bimestrais, calcula e mostra a média aritmética destas duas notas. Lembre-se que sempre antes de sair escrevendo o programa JavaScript, é importante entender qual o problema que precisa ser resolvido,

identificar a entrada de dados e seus tipos, além da saída dos resultados. 

```
1 <script>
2
3 nota1 = parseInt(window.prompt("digite uma nota bimestral: "));
4 nota2 = parseInt(window.prompt("digite outra nota bimestral: "));
5
6 media = (nota1 + nota2) / 2;
7
8 document.write("A média é: " + media + "<br>");
9
10 </script>
```

O resultado da execução deste programa JavaScript ficará:



Programando JavaScript

A média é: 6.5

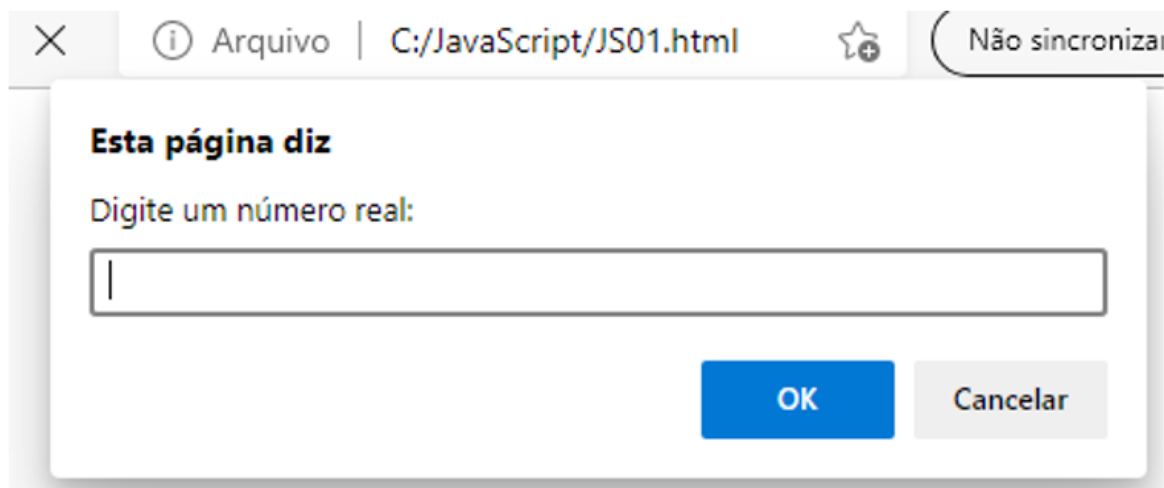
Estrutura de Seleção em JavaScript

Para entendermos a aplicação da estrutura de decisão em um programa JavaScript, vamos praticar desenvolvendo um algoritmo que recebe um valor real, verifica e mostra se esse número é negativo.

Entendemos que precisamos receber como entrada de dados um float, um processamento para verificar se o número é negativo e apresentar esta informação. Neste programa, vamos utilizar uma estrutura de decisão simples.

```
1 <script>
2
3 x = parseFloat(window.prompt("Digite um número real: "));
4
5 if (x < 0)
6 {
7     document.write(x + " é negativo");
8 }
9
10 </script>
```

Veja o resultado da execução do programa JavaScript.





Programando JavaScript

-3.5 é negativo

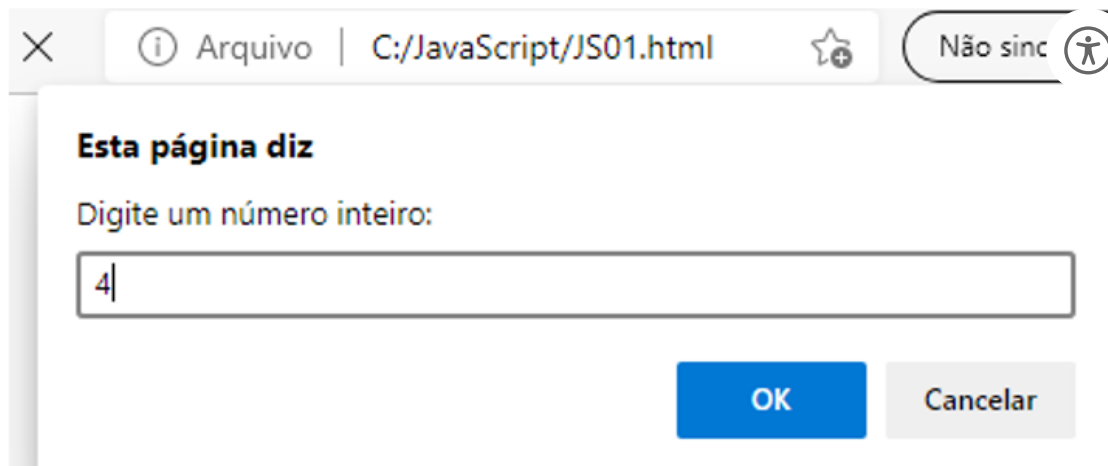
Fonte: Autoral

Vamos praticar desenvolvendo um programa JavaScript que recebe um valor inteiro, verifica e mostra se o número é par ou ímpar.

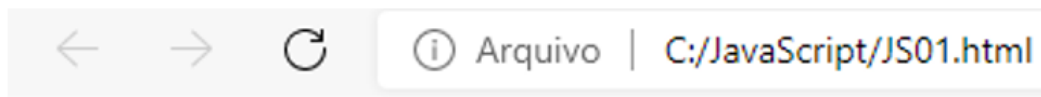
Neste programa JavaScript, temos como entrada de dados um número inteiro, processando esse número para verificar se é par ou ímpar e apresentando o resultado.

```
1 <script>
2
3 x = parseInt(window.prompt("Digite um número inteiro: "));
4
5 if ((x % 2) == 0)
6 {
7     document.write(x + " é par");
8 }
9 else
10 {
11     document.write(x + " é ímpar");
12 }
13
14 </script>
```

O resultado da execução deste programa JavaScript:

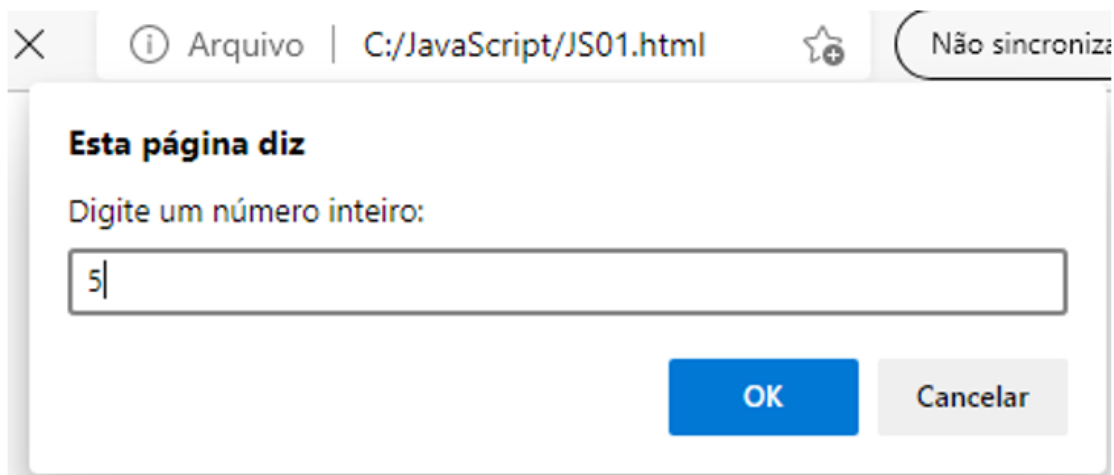


Fonte: Autoral



Programando JavaScript

4 é par





Atividade Extra

Sugiro que faça a leitura dos textos abaixo:

Link: [Saiba mais sobre instrução while_](#) (Acesso em 03/08/2022)

Link: [Veja mais exemplos de uso da instrução for_](#) (Acesso em 03/08/2022)


Referência Bibliográfica

Deitel, Paul J. e Deitel, Harvey M. Ajax, Rich **Internet Applications e Desenvolvimento Web para programadores**. São Paulo. Pearson Prentice Hall. 2008.

Morrison, Michael. **Use a Cabeça! JavaScript**. 1a Edição. Rio de Janeiro. Alta Books. 2009.

Atividade Prática 16 - Manipulando Elementos do DOM

Título da Prática: Manipulando Elementos do DOM.

Objetivos: Compreendendo o processo de manipulação dos elementos do DOM. 

Materiais, Métodos e Ferramentas: Visual Studio Code ou Bloco de notas do Windows.

Atividade Prática

1º Passo:

Leia atentamente o texto.

O comando **window.prompt()** instrui o navegador a exibir uma caixa de diálogo com uma mensagem opcional solicitando que o usuário insira algum texto e aguarde até que o usuário envie o texto ou cancele a caixa de diálogo.

Sob algumas condições — por exemplo, quando o usuário alterna entre as guias — o navegador pode não exibir uma caixa de diálogo ou pode não esperar que o usuário envie o texto ou cancele a caixa de diálogo.

Sintaxe

```
prompt()
```

```
prompt(mensagem)
```

```
prompt(mensagem, variavel)
```

Parâmetros

➤ **mensagem**



Uma string de texto a ser exibida. Observe que essa mensagem poderá ser omitida se não houver nada para mostrar na janela do prompt.

➤ **variavel**

Uma string contendo o valor padrão exibido no campo de entrada de texto.

Valor de retorno

Uma string contendo o texto digitado pelo usuário ou null.

2º Passo:

O método **prompt()** apresenta uma caixa de diálogo que solicita uma entrada do usuário. Além disso, esse método retorna o valor de entrada se o usuário clicar em “OK”, caso contrário, ele retorna null.

Observação

Uma caixa de prompt é usada se você deseja que o usuário insira um valor.

Quando uma caixa de prompt aparecer, o usuário terá que clicar em “OK” ou “Cancelar” para prosseguir.

Não abuse deste método. Impede que o usuário acesse outras partes da página até que a caixa seja fechada.

Veja a utilização desse recurso no código abaixo:

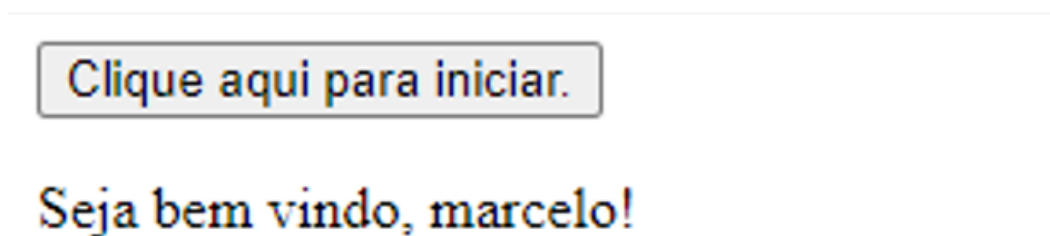
Figura 1 – Reprodução do código fonte.



```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <button onclick="cadastroUsuario()">Clique aqui para iniciar.</button>
5 <p id="mensagemInicial"></p>
6 <script>
7 function cadastroUsuario() {
8     nomeUsuario = prompt("Cadastro", "Digite o seu nome");
9     if (nomeUsuario != null) {
10         document.getElementById("mensagemInicial").innerHTML =
11             "Seja bem vindo, " + nomeUsuario + "!";
12     }
13 }
14 </script>
15 </body>
16 </html>
17
```

Fonte: o próprio autor (2023).

Figura 2 – Reprodução da execução do código fonte.



3º Passo:

A partir do código de referência abaixo, implemente a validação dos demais meses restantes.

Figura 3 – Reprodução do código fonte.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5   </head>
6   <body>
7
8   <button onclick="validaMes()">Clique aqui</button>
9
10  <p id="mensagemFuncao"></p>
11
12  <script>
13  function validaMes() {
14    let text;
15    let mes = prompt("Digite o mês atual (valor numérico)");
16    switch(mes) {
17      case "1":
18        text = "Janeiro.";
19        break;
20      default:
21        text = "Mês inválido..";
22    }
23    document.getElementById("mensagemFuncao").innerHTML = text;
24  }
25 </script>
26
27 </body>
28 </html>
```



O usuário deverá digitar o número do mês, e a lógica do seu programa deverá exibir o mês por extenso, como por exemplo: “Janeiro”, “Fevereiro”, “Março”, “Abril”, etc.

Comentários e conclusões:


1. O que aprendeu com esta atividade?
2. Tudo funcionou como o esperado?
3. Quais foram as principais dificuldades?

Gabarito Atividade Prática

Segue o código fonte da atividade proposta:



Figura 3 – Reprodução do código fonte.



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5   </head>
6   <body>
7
8   <button onclick="validaMes()">Clique aqui</button>
9
10  <p id="mensagemFuncao"></p>
11
12  <script>
13  function validaMes() {
14    let text;
15    let mes = prompt("Digite o mês atual (valor numérico)");
16    switch(mes) {
17      case "1":
18        text = "Janeiro.";
19        break;
20      case "2":
21        text = "Fevereiro.";
22        break;
23      case "3":
24        text = "Março.";
25        break;
26      case "4":
27        text = "Abril.";
28        break;
29      case "5":
30        text = "Maio.";
31        break;
32      case "6":
33        text = "Junho.";
34        break;
35      case "7":
36        text = "Julho.";
37        break;
38      case "8":
39        text = "Agosto.";
40        break;
41      case "9":
42        text = "Setembro.";
43        break;
44      case "10":
45        text = "Outubro.";
46        break;
47      case "11":
48        text = "Novembro.";
49        break;
50      case "12":
51        text = "Dezembro.";
52        break;
53      default:
54        text = "Mês inválido..";
55    }
56    document.getElementById("mensagemFuncao").innerHTML = text;
57  }
58  </script>
59
60 </body>
61 </html>
```

Fonte: o próprio autor (2023)

Ir para exercício

