

Desenvolvimento Web

Aula – Javascript

Prof. Dr. Fernando Parra dos Anjos Lima
fernando.lima@tga.ifmt.edu.br

Conteúdo da Aula

- Introdução;
- Definição;
- Histórico;
- Sobre a linguagem;
- Requisitos mínimos;
- Aplicando JavaScript ao HTML;
- Carregamento;
- Console;
- Variáveis;
- Operadores;
- Funções;

Introdução

- Até agora estudamos as linguagens HTML e CSS e vimos como elas são fundamentais para a criação de páginas web.
- O foco do HTML é o conteúdo enquanto o foco da CSS é a formatação das páginas;
- As linguagens HTML e CSS não são linguagens de programação, sendo que para resolver determinados problemas as linguagens de programação são necessárias.
- Usando a linguagem JavaScript é possível construir páginas mais dinâmicas e interativas. O foco do JS é implementar o comportamento ou a inteligência das páginas Web.

Definição

- JavaScript é uma linguagem de programação que roda no **lado cliente** (no navegador do usuário) e é capaz de realizar algumas ações em uma página web tornando-a mais interativa.
- O propósito original da linguagem JavaScript era client side porém hoje há soluções como o Node.js que rodam JS no servidor para diversos propósitos.

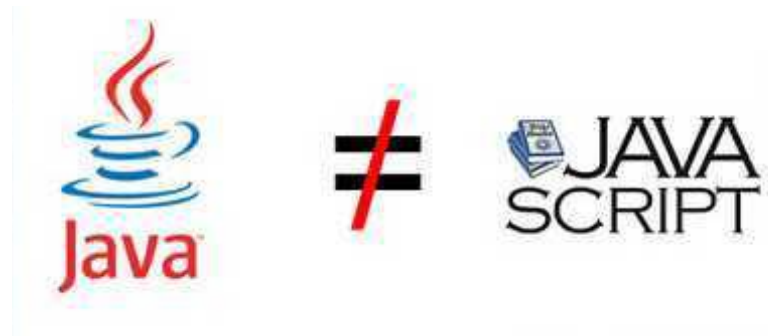
Histórico

- Criada pela Netscape em 1995 inicialmente se chamava LiveScript e visava atender necessidades de maior interação da página com o usuário e prover um mecanismo de validação de formulários.



Sobre a Linguagem

- JavaScript é uma linguagem que possui uma sintaxe semelhante ao Java, porém ela não tem nenhuma relação com o mesmo.



- A grande vantagem do JavaScript é a capacidade de interagir com uma página HTML permitindo que conteúdos sejam adicionados e manipulados dinamicamente.

Sobre a Linguagem

- Algumas das interações possíveis de serem implementadas com JS:
- Interações com mouse, teclado;
- Animações, sliders;
- Requisições ajax;
- Validações de formulários dinâmicos;

Sobre a Linguagem

- É uma linguagem interpretada.
- Aliada a outras tecnologias pode se tornar muitíssimo poderosa.



Requisitos mínimos

- Para começar a programar usando JavaScript são necessários poucos recursos...



Hello World em Javascript

- Vamos criar um documento HTML com a estrutura básica e fazer o “Hello World” em JavaScript...
- Para começar vamos colocar o código dentro do cabeçalho da página.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Exempo 1 JavaScript</title>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scal
7
8    </head>
9    <body>
10     <h1>Exempo 1 JavaScript</h1>
11  </body>
12 </html>
```



Hello World em Javascript

- E agora dentro de uma TAG script colocaremos uma instrução em JavaScript para mostrar uma mensagem na tela;

```
<script>  
    alert("Olá Mundo!");  
</script>
```

Hello World em Javascript

- Código completo:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exempo 1 JavaScript</title>
    <meta charset="UTF-8">
    <script>
      alert("Olá Mundo!");
    </script>
  </head>
  <body>
    <h1>Exempo 1 JavaScript</h1>
  </body>
</html>
```

Aplicando Javascript ao HTML

- De acordo com a documentação do HTML a tag `SCRIPT` serve para “definir um script client-side (JavaScript)”. Ela pode conter um código JS ou apontar para um arquivo externo usando o atributo **src**.
- Há basicamente duas formas de se associar código JS aos documentos HTML.
- JavaScript interno;
- JavaScript externo;

Aplicando Javascript ao HTML

- JavaScript interno:

O código JS pode ser definido dentro de um documento HTML o corpo de uma tag script


Assim:

```
<script>  
    alert("Olá Mundo!");  
</script>
```

Aplicando Javascript ao HTML

- JavaScript interno:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Exemplo 1 - JavaScript</title>
5          <meta charset="UTF-8">
6
7      </head>
8      <body>
9          <h1>Exemplo 1 - JavaScript</h1>
10     </body>
11 </html>
12
```

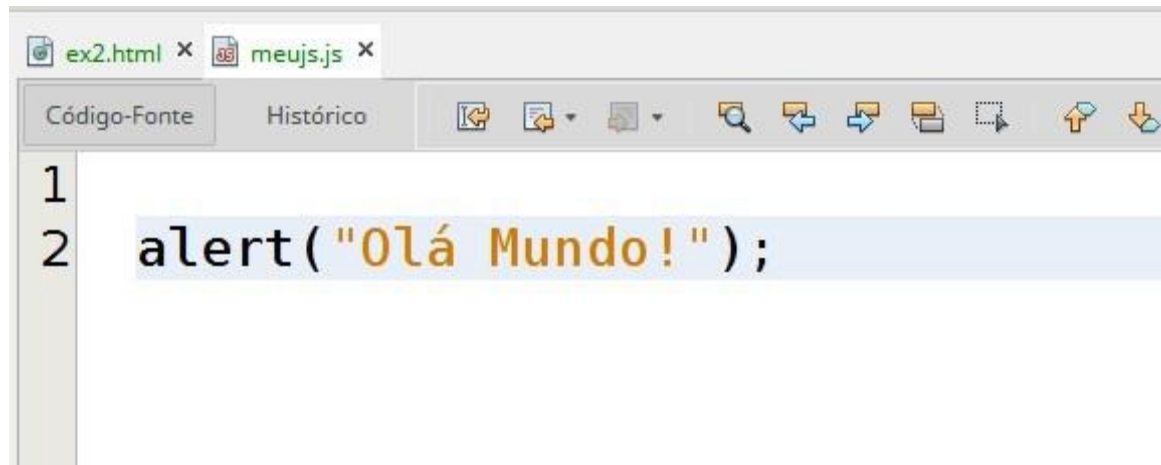


Aplicando Javascript ao HTML

- JavaScript externo:

O código JS também pode ser definido em arquivos separados e depois associado ao documento HTML através da tag script.

Criar um arquivo JavaScript (meujs.js)



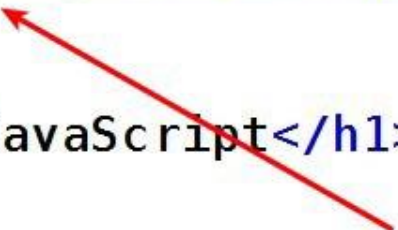
The screenshot shows a code editor with two tabs: 'ex2.html' and 'meujs.js'. The 'meujs.js' tab is active, showing a single line of JavaScript code: `alert("Olá Mundo!");`. The code is highlighted in blue. The editor has a toolbar with various icons for file operations and a sidebar with a line number indicator showing '1' and '2'.

Aplicando Javascript ao HTML

- JavaScript externo:

Arquivo.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Exemplo 2 - JavaScript</title>
5          <meta charset="UTF-8">
6          <script src="meujs.js"></script>
7      </head>
8      <body>
9          <h1>Exemplo 2 - JavaScript</h1>
10     </body>
11 </html>
```



Aplicando Javascript ao HTML

- A grande diferença entre JS interno e externo é que na primeira abordagem o código fica “amarrado” a um único documento HTML. Já na segunda é possível usar o mesmo código para outros documentos.

Carregamento

- Como acabamos de ver a tag script serve para associar o código JavaScript (interno ou externo) aos documentos HTML.
- Essa tag pode ser adicionada dentro do corpo da página (body) ou do cabeçalho (head).
- No entanto a localização desta declaração afeta o momento no qual o código JavaScript será carregado pelos navegadores.

Carregamento

- Situação 1: tag **script** dentro do **head**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exempo 1 JavaScript</title>
    <meta charset="UTF-8">
    <script src="meujs.js"></script>
  </head>
  <body>
    <h1>Exempo 1 JavaScript</h1>
  </body>
</html>
```

Carregamento

- Situação 1: tag script dentro do head
- Dessa forma, o código JavaScript será carregado antes do body ser processado.
- Consequentemente, a página só será exibida depois do carregamento do JavaScript.

Carregamento

- Situação 2: tag **script** no início do **body**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo 2 JavaScript</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <script src="meujs.js"></script>
    <h1>Exemplo 2 JavaScript</h1>
  </body>
</html>
```

Carregamento

- Situação 2: tag script no início do body
- Desta forma o JavaScript será carregado juntamente com o corpo da página.
- No caso ele ainda será carregado antes de todo o conteúdo HTML.

Carregamento

- Situação 3: tag **script** no final do **body**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo 2 JavaScript</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>Exemplo 2 JavaScript</h1>
    <script src="meujs.js"></script>
  </body>
</html>
```


Carregamento

- Situação 3: tag script no final do body
- Nesta última situação há uma grande diferença, como a tag script foi colocada no final do body o JavaScript será carregado apenas após a completa renderização do corpo da página.
- Essa abordagem é bastante interessante pois caso o código JS seja “pesado” a pagina não ficará vazia até o completo carregamento do JS.

Funções

- Em programação uma função (também chamada de método) é uma rotina usada para resolver problemas de modo estruturado, ou seja, um conjunto de linhas de código que podem ser usadas diversas vezes no programa.
- Sintaxe de uma função:

```
<script>  
    function nome_funcao(){  
        /* Código */  
    }  
</script>
```

Funções

- Uma função pode ou não ter argumentos. Argumentos de uma função são as variáveis que serão passadas na chamada da função para que sejam usadas em seu código, é por meio delas que a função interage com o código externo a ela.
- Sintaxe de uma função com argumentos:

```
<script>  
function nome_funcao(arg1, arg2){  
    /* Código */  
}  
</script>
```



Praticando

- Vamos praticar mais um pouco, agora usando uma função.
- Crie uma página como esta:

Exemplo 3 JavaScript



Praticando

- Código completo:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Exemplo 3 Javascript</title>
5      <meta charset="UTF-8">
6      <meta name="viewport" , content="width=device-width, init
7  <style>
8      div{
9          margin: 20px;
10         padding: 20px;
11         border: solid 2px #000;
12     }
13 </style>
14 </head>
15 <body>
16     <h1>Exemplo 3 Javascript</h1>
17     <div style="background-color: red;">
18
19     </div>
20     <div style="background-color: green;">
21
22     </div>
23     <div style="background-color: blue;">
24
25     </div>
26 </body>
27 </html>
```

Praticando

- O objetivo do exemplo é fazer com que ao clicar em cada DIV colorido o plano de fundo da página seja alterado para a cor do DIV clicado.
- Para isso vamos criar uma função que faz essa alteração e colocar uma chamada a essa função em cada um dos DIVs.

Praticando

- Primeiro crie o código JS da função:

```
<script>  
    function mudacor(color){  
        document.getElementsByTagName("BODY")[0].  
            style.backgroundColor = color;  
    }  
</script>
```

* não é preciso colocar em duas linhas, fiz isso para ficar melhor no slide.

Praticando

- Em seguida vamos colocar as “chamadas” a função em cada DIV, utilizaremos o atributo “onclick”, que é disparado quando clicamos sobre os DIVs;
- Veja:

```
<div class="box verm" onclick="mudacor( 'red' );"></div>  
<div class="box verd" onclick="mudacor( 'green' );"></div>  
<div class="box azul" onclick="mudacor( 'blue' );"></div>
```


Praticando

- Como ficou:

Exemplo 3 Javascript

Variáveis

- Em JS as variáveis devem iniciar com uma letra ou um _ (underline).
- A linguagem diferencia maiúsculas de minúsculas (case sensitive).
- Para se criar uma variável deve-se usar antes do nome da variável a instrução var.
- Exemplos:

```
<script>  
    var nome;  
    var cidade, estado;  
    var sit = true;  
    var str = "Qual é seu nome?";  
    var sal = 678.00;  
</script>
```

Operadores

- Para manipular os valores de variáveis de um programa, são usados os operadores.
- Em JavaScript assim como na maioria das linguagens existem operadores básicos, matemáticos e lógicos.

Operador	Operação
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da Divisão
++	Incremento
--	Decremento
=	Atribuição

Operadores

- Exemplo de uso de operadores.
- O que será mostrado no “alert”?

```
<script>
  var a = 3;      // variável a recebe 3
  var b = "3";    // variável b recebe "3"

  var res;

  res = a + b;
  alert(res);
</script>
```

- O operador + tem duas funções. Caso uma das variáveis seja do tipo string (texto) ele será utilizado como operador de concatenação (juntar). Desta forma é importante prestar atenção aos tipos das variáveis.

Operadores

- Relacionais e Lógicos:

Operador	Operação
==	Igualdade
!=	Diferença
<	Menor
<=	Menor ou igual
>	Maior
>=	Maior ou Igual
&&	Op. Lógico AND
	Op. Lógico OR

Operadores

- Exemplo de uso de operadores

```
<script>
    var val, nota;

    if(val == "a"){
        // Código qualquer
    }

    if(nota < 6 && nota >= 4){
        // Código qualquer
    }
</script>
```

Caixas de Dialogo

- Existem diferentes tipos de caixa de diálogo, são elas:
 - alerta (alert)
 - confirmação (confirm)
 - prompt de entrada (prompt).

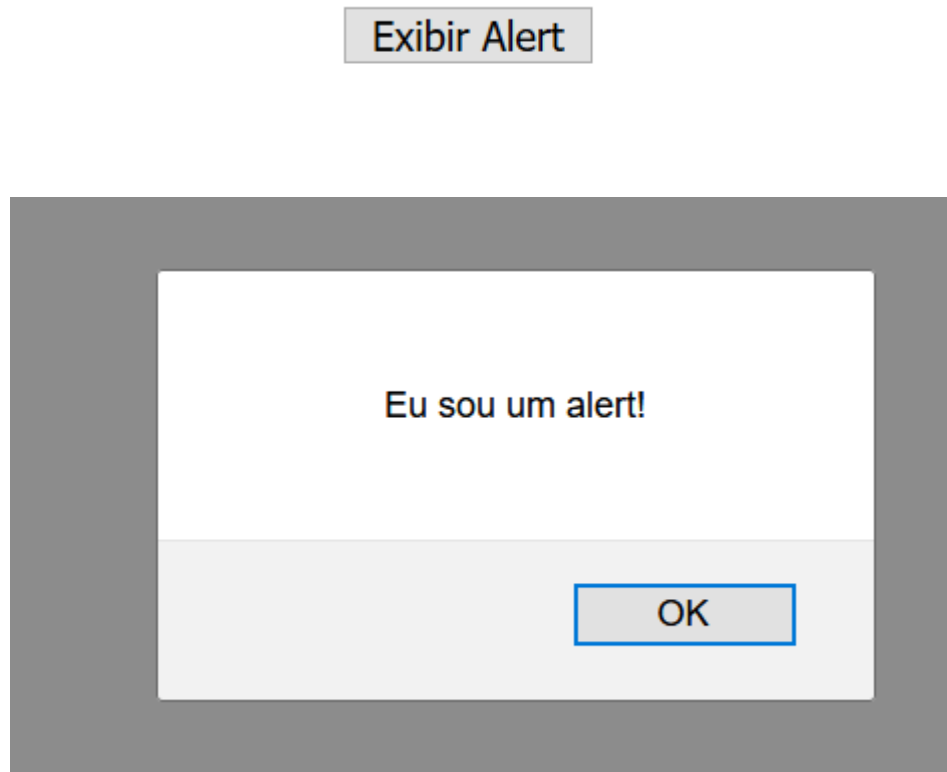
Caixas de Dialogo - Alerta

- Código:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Tutorial de Alert em JavaScript - Linha de Código</title>
    <script>
      function funcao1() {
        alert("Eu sou um alert!");
      }
    </script>
  </head>
  <body>
    <input type="button" onclick="funcao1()" value="Exibir Alert" />
  </body>
</html>
```


Caixas de Dialogo - Alerta

- Como Ficou:



Caixas de Dialogo - Confirmação

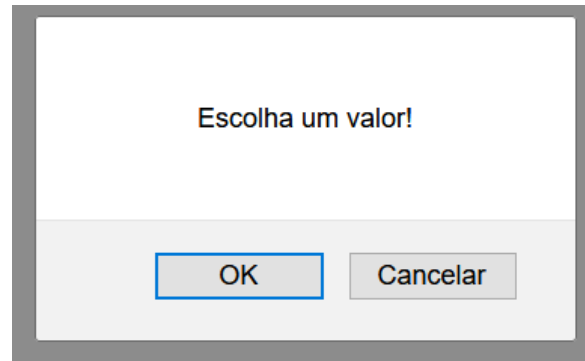
```
<!DOCTYPE html>
<html>
  <head>
    <title>Tutorial de Alert em JavaScript - Linha de Código</title>
  </head>
  <body>
    <p>Clique no botão para exibir a caixa de confirmação.</p>
    <button onclick="funcao1()">Clique aqui</button>
    <p id="demo"></p>
    <script>
      function funcao1() {
        var x;
        var r=confirm("Escolha um valor!");
        if (r==true) {
          x="você pressionou OK!";
        }
        else {
          x="Você pressionou Cancelar!";
        }
        document.getElementById("demo").innerHTML=x;
      }
    </script>
  </body>
</html>
```

Caixas de Dialogo - Confirmação

- Como ficou:

Clique no botão para exibir a caixa de confirmação.

Clique aqui



Clique no botão para exibir a caixa de confirmação.

Clique aqui

você pressionou OK!

Caixas de Dialogo - Prompt

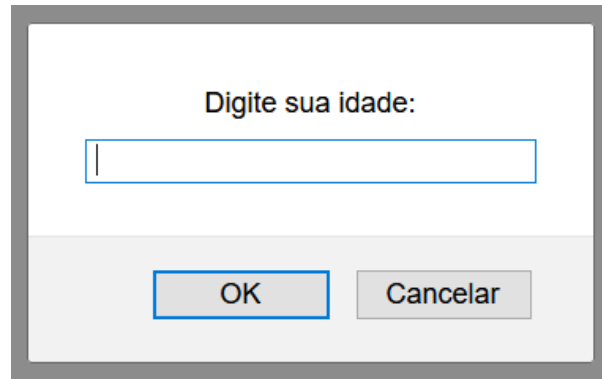
```
<!DOCTYPE html>
<html>
  <head>
    <title>Tutorial de Alert em JavaScript - Linha de Código</title>
  </head>
  <body>
    <p>Clique para exibir.</p>
    <button onclick="myFunction()">Clique aqui</button>
    <p id="demo"></p>
    <script>
      function myFunction() {
        var x;
        var idade=prompt("Digite sua idade:");
        if (idade!=null) {
          x="Idade: " + idade + " anos.";
          document.getElementById("demo").innerHTML=x;
        }
      }
    </script>
  </body>
</html>
```

Caixas de Dialogo - Prompt

- Como ficou:

Clique para exibir.

Clique aqui



Digite sua idade:

OK Cancelar

Clique para exibir.

Clique aqui

Idade: 30 anos.

Praticando

- Exercício 1: Para treinar um pouco, ajuste o exemplo 3 para que ao colocar o mouse sobre o DIV colorido o plano de fundo seja alterado para a cor dele e ao clicar sobre o o plano de fundo do body, ele deverá voltar a ser branco.

Praticando

■ Exercício 1: Resposta

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exercicio 1 Javascript</title>
    <meta charset="UTF-8">
    <meta name="viewport" , content="width=device-width, initial-scale=1.0">
    <style>
      div{
        margin: 20px;
        padding: 20px;
        border: solid 2px #000;
      }
      red{
        background-color: red;
      }
    </style>
    <script src="meujs.js"></script>
  </head>
  <body onclick="mudar('white');">
    <h1>Exercicio 1 Javascript</h1>
    <div style="background-color: red;" class="box verm" onmouseover="mudar('red');">
    </div>
    <div style="background-color: green;" class="box verd" onmouseover="mudar('green');">
    </div>
    <div style="background-color: blue;" class="box azul" onmouseover="mudar('blue');">
    </div>
  </body>
</html>
```

```
function mudar(color) {
  document.getElementsByTagName("BODY")[0].style.backgroundColor = color;
}
```

Praticando

- Exercício 2: Faça um código para ler o nome de um aluno e 3 notas. Calcule a média e verifique se o aluno é aprovado ou reprovado (média > 7). Imprima se aprovado ou reprovado.

Praticando

■ Exercício 2: Resposta

```
<!DOCTYPE html>
<head>
  <meta charset="utf-8">
  <title>Exercício</title>
</head>
<body>
  <script>
    var nome=prompt("Digite o nome do aluno:");
    var notal = parseInt(prompt("Digite a primeira nota:"));
    var nota2 = parseInt(prompt("Digite a segunda nota:"));
    var nota3 = parseInt(prompt("Digite a terceira nota:"));
    var media = (notal + nota2 + nota3)/3;
    if (media>=7) {
      document.write ("Aluno Aprovado! - Media = "+media)
    }
    else {
      document.write ("Aluno Reprovado! - Media = "+media)
    }
  </script>
</body>
</html>
```

Obrigado !!

Prof. Dr. Fernando Parra dos Anjos Lima
fernando.lima@tga.ifmt.edu.br