

**JS**



# JavaScript Parte 1

TREINAMENTO PROF. WELLINGTON TELLES CUNHA



# JavaScript

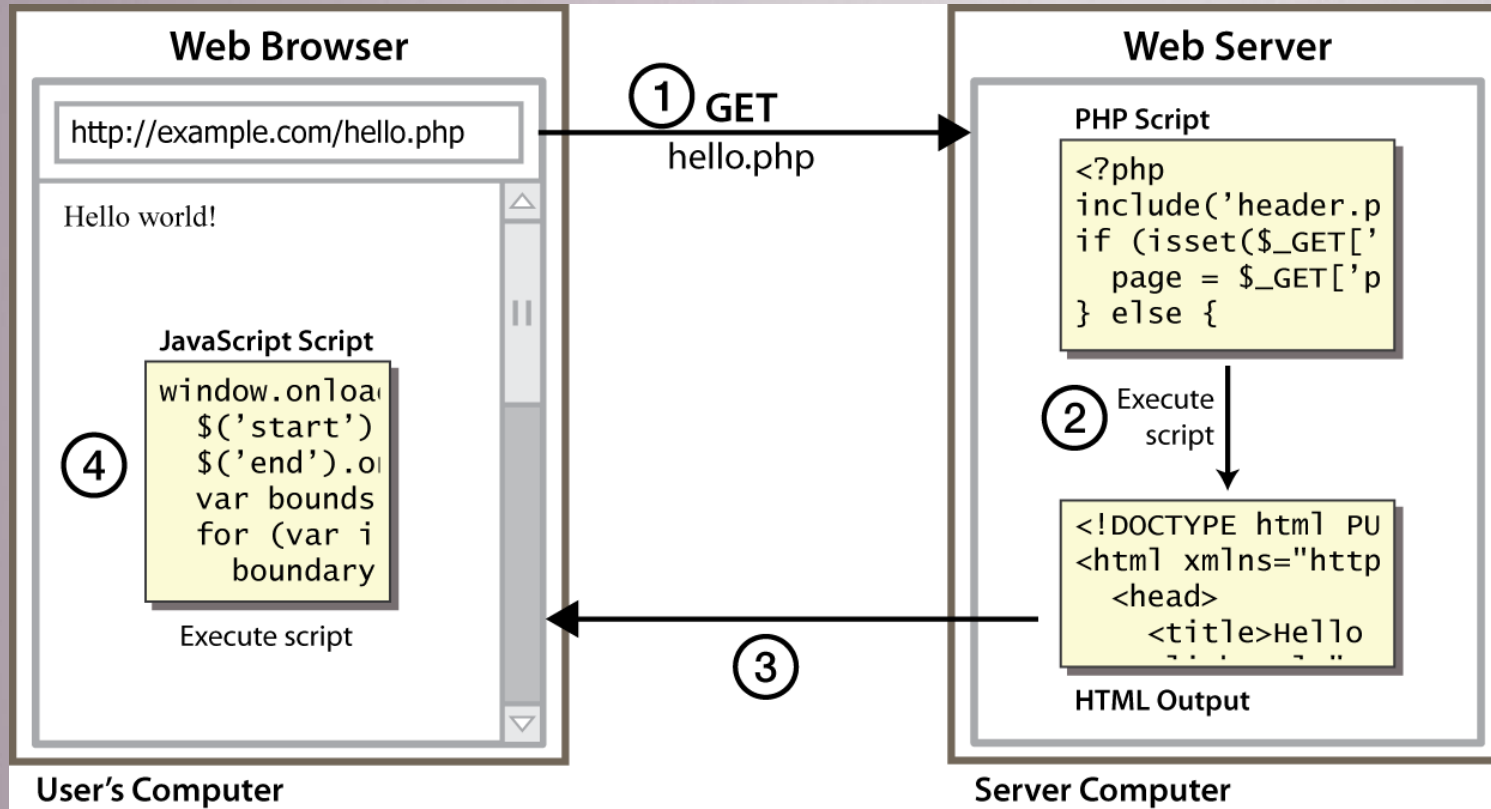
## Brendan Eich

Originalmente desenvolvido por **Brendan Eich** da Netscape sob o nome de Mocha, posteriormente teve seu nome mudado para **LiveScript** e por fim **JavaScript**.

**LiveScript** foi o nome oficial da linguagem quando foi lançada pela primeira vez na versão beta do navegador **Netscape 2.0** em setembro de 1995, mas teve seu nome mudado em um anúncio conjunto com a Sun Microsystems em dezembro de 1995 quando foi implementado no navegador Netscape versão 2.0B3.

Em novembro de 1996 a Netscape anunciou que tinha submetido JavaScript para Ecma internacional como candidato a padrão industrial e o trabalho subsequente resultou na versão padronizada chamada **ECMAScript**.

# Scripts do lado do cliente



# Por que usar a programação do lado do cliente?



Se o PHP já nos permite criar páginas web dinâmicas.

Por que usar também scripts do lado do cliente?

- ▶ Benefícios do Script do Lado do Cliente (**JavaScript**):
  - ▶ **usabilidade**: pode modificar uma página sem precisar **postar** de volta no servidor (interface do usuário mais rápida)
  - ▶ **eficiência**: pode fazer alterações pequenas e rápidas na página sem esperar pelo servidor
  - ▶ **orientada à eventos**: pode responder a ações do usuário, como cliques e pressionamentos de tecla



# Por que usar a programação do lado do servidor?

5

## ▶ Benefícios de programação do lado do servidor (PHP):

- ▶ **Segurança:** tem acesso aos dados privados do servidor; o cliente não pode ver o código-fonte
- ▶ **Compatibilidade:** não está sujeito a problemas de compatibilidade do navegador
- ▶ **Poder:** pode gravar arquivos, abrir conexões com servidores, conectar-se a bancos de dados, ...

BENEFITS





# O que é JavaScript?

- ▶ Uma linguagem de programação leve ("**linguagem de script**")
  - ▶ Usado para tornar as páginas da Web interativas
  - ▶ Insira texto dinâmico em HTML (ex.: nome de usuário)
  - ▶ **Reagir à eventos** (ex.: clique no usuário do carregamento da página)
  - ▶ Obter informações sobre o computador de um usuário (ex.: tipo de navegador)
  - ▶ Realizar cálculos no computador do usuário (ex.: validação de formulário)



# O que é JavaScript?

- ▶ Um padrão da web (mas **não** suportado de forma **idêntica** por todos os navegadores)
- ▶ **NÃO** relacionado a Java além do nome e algumas semelhanças sintáticas

# JavaScript

8



Linguagem integrada no navegador



Linguagem interpretada



Modelo de execução controlado por eventos



Baseada em Objetos (Windows, Date(), Array(), Math())



Tipagem dinâmica: tipos são associados com valores e não com variáveis



# O que podemos fazer?

9



ANIMAÇÕES



MANIPULAR  
ELEMENTOS DO  
DOCUMENTO  
HTML



ACESSAR SEUS  
VALORES



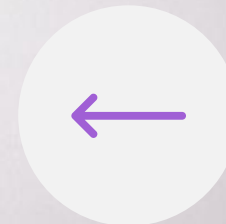
ALTERAR  
ESTILOS



ALTERAR  
VALORES



MANIPULAR  
ARQUIVOS XML  
E JSON



SERVER-SIDE:  
NODE.JS

# JavaScript vs. PHP



## ▶ Semelhanças:

- ▶ Ambos são interpretados, não compilados
- ▶ Ambos estão relaxados sobre sintaxe, regras e tipos
- ▶ Ambos são sensíveis a maiúsculas e minúsculas
- ▶ Ambos têm expressões regulares incorporadas para processamento poderoso de texto

# JavaScript vs. PHP



- ▶ Diferenças:
  - ▶ JS é mais orientado à objetos: noun.verb (), menos procedural: verbo (substantivo)
  - ▶ O JS se concentra nas interfaces do usuário e interage com um documento; O PHP é voltado para a saída HTML e processamento de arquivos / formulários
  - ▶ O código JS é executado no navegador do cliente; Código PHP é executado no servidor da web

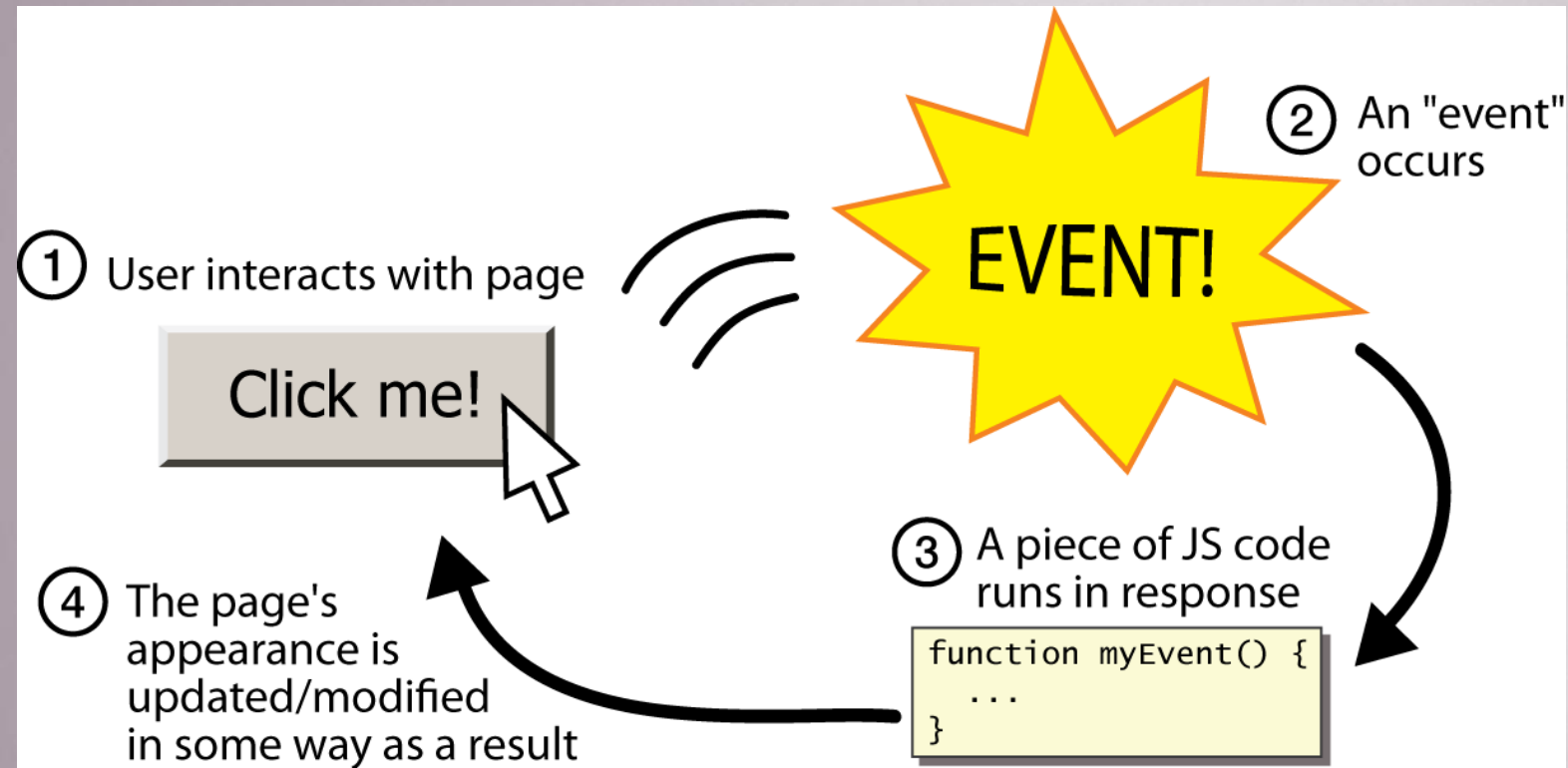
# Javascript vs Java

- ▶ interpretado, não compilado
- ▶ sintaxe e regras mais relaxadas
  - ▶ tipos de dados menos e mais "soltos"
  - ▶ variáveis não precisam ser declaradas
  - ▶ erros muitas vezes silenciosos (poucas exceções)
- ▶ chave construct é a função e não a classe
  - ▶ funções de "primeira classe" são usadas em muitas situações
- ▶ contido em uma página da Web e se integra ao conteúdo HTML / CSS





Javascript  
vs Java




## Programação Orientada à Eventos



# Programação Orientada a Eventos

15

Você está acostumado a programas começam com um método **main** (ou principal implícito como no PHP), isso não é necessário



Programas JavaScript, esperam por **ações** do usuário chamando eventos para responder à eles



Programação orientada à eventos: programas de escrita dirigidos por **eventos** do usuário

# Aonde ocorrem?

- ▶ Uma tag **<script/>** pode ser definida numa seção head, numa seção body e também pode ser definida externamente:
  - ▶ Na **seção head**, os scripts são executados quando são chamados ou quando algum evento ocorre;
  - ▶ Na **seção body**, os scripts são executados na carga da página web
  - ▶ Para **definição externa**, um arquivo “.js” precisa ser fornecido com as funções necessárias
  - ▶ Dentro de uma **tag html**



# Sintaxe

- ▶ Possui construções existentes na maioria das linguagens de programação (sintaxe similar a C):
  - ▶ Declaração de variáveis (**var x;**)
  - ▶ Comandos condicionais, repetições, definição de funções de usuário
  - ▶ Operadores de atribuição, comparação, ...

« . . . »

“ ”

. . .

‘ ’

. . .



Eventos são ações que podem ser detectadas por um script



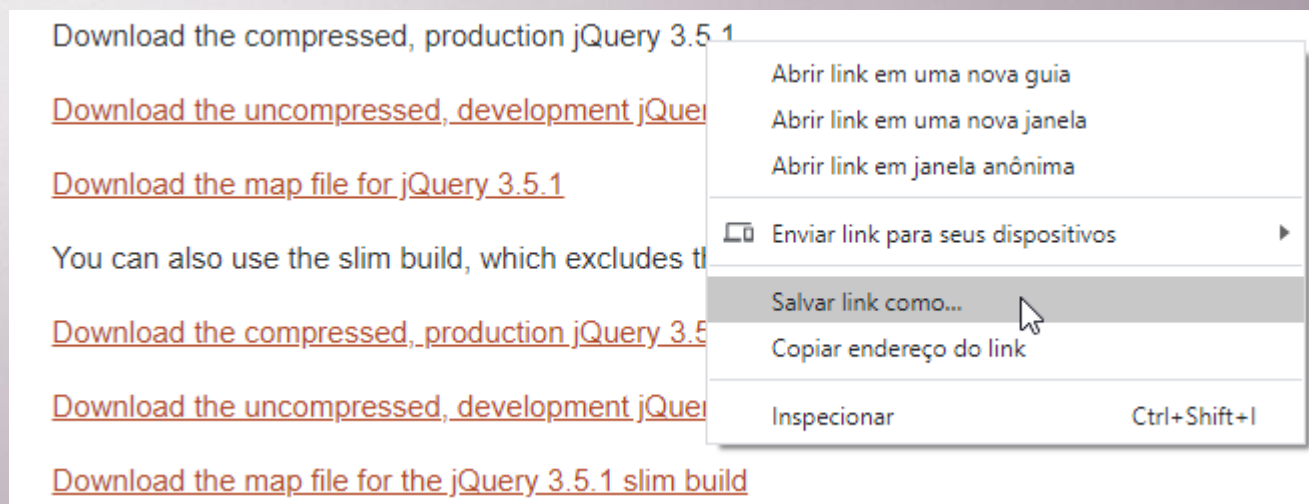
Exemplos de eventos:

Clique do mouse, abertura de uma página web ou imagem, envio de um formulário html, uma tecla pressionada, etc.

O tratamento destes eventos pode ser a chamada de funções do script



- ▶ Biblioteca do JavaScript
  - ▶ Crie primeiro uma pasta **js** dentro da pasta **www**
  - ▶ Acesse o site:
    - ▶ <https://jquery.com/download/>
    - ▶ Encontre o **COMPRESSED**
    - ▶ Salve o arquivo na pasta **js**



# Primeiro Programa

(primeiro\_js.html)

localhost diz

Hello World!

OK

20

```
<!DOCTYPE html>
<html>
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
    <title>Hello World!</title>
    <script type="text/javascript" src="../../js/jquery-3.5.1.min.js"></script>
    <script type="text/javascript" >
      $(document).ready(function() {
        alert ("Hello World!");
      });
    </script>
  </head>
  <body>
  </body>
</html>
```



# Alert

(alerta\_js.html)

localhost diz

11

OK

21

```
<!DOCTYPE html>
<html>
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
    <title>Untitled 1</title>
  </head>
  <body>
    <script>
      window.alert(5 + 6);
    </script>
  </body>
</html>
```

# Comentários

(o mesmo que Java)

- ▶ idêntica à sintaxe de comentários do Java
- ▶ recall: 4 sintaxes de comentários
  - ▶ HTML: `<!-- comentário -->`
  - ▶ CSS / JS / PHP: `/ * comentário *`
  - ▶ Java / JS / PHP: `// comentário`
  - ▶ PHP: `# comentário`

## Funções



O conteúdo pode vinculado à página HTML

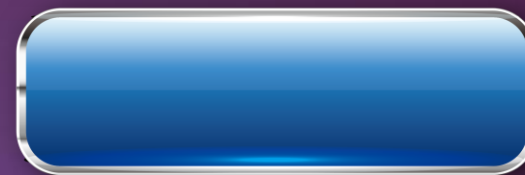


instruções inseridas em funções podem ser avaliadas em resposta à eventos do usuário

# Funções

```
function nome() {  
    declaração ;  
    ...  
    declaração ;  
}  
  
function minhaFuncao() {  
    alert("Alo!");  
    alert("Aonde você está?");  
}
```

# Botões

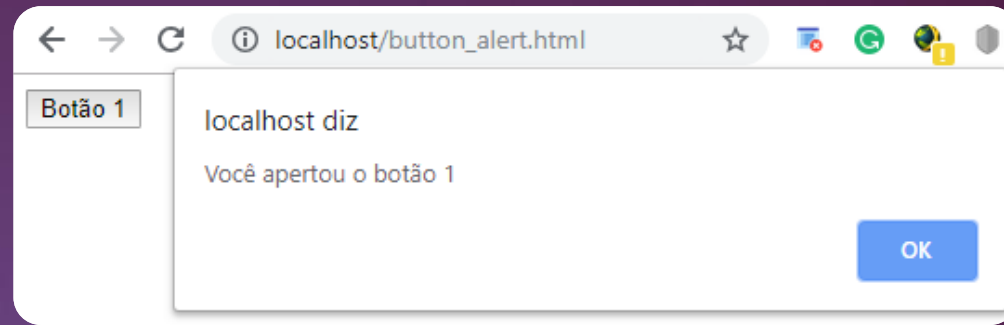


- ▶ O texto do botão aparece dentro da tag; (também pode conter imagens)
- ▶ Para fazer um botão responsivo ou outro controle de interface do usuário:
  - ▶ Escolha o controle (por exemplo, botão) e o evento (por exemplo, mouse: clique) de interesse
  - ▶ Escrever uma função JavaScript para ser executada quando o evento ocorrer
  - ▶ Anexar a função ao evento no controle

# Botão

(botao\_js.html)

26



```
<!DOCTYPE html>
<html>
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
    <title>Untitled 1</title>
    <script type="text/javascript" src="../../js/jquery-3.5.1.min.js"></script>
    <script>
      function botao1()
      {
        alert("Você apertou o botão 1");
      }
    </script>
  </head>
  <body>
    <input type="button" onclick="botao1()" value="Botão 1"/>
  </body>
</html>
```



# Manipuladores de eventos

- ▶ **Funções** JavaScript podem ser definidas como **manipuladores** de eventos
  - ▶ quando você interage com o elemento, a função será executada
  - ▶ **onclick** é apenas um dos muitos atributos HTML de eventos que usaremos
  - ▶ mas aparecer uma **janela de alerta** é perturbador e irritante
  - ▶ Uma melhor experiência do usuário seria fazer com que a mensagem **aparecesse na página** ...



# Manipuladores de eventos

28



▶ `<element attributes onclick="function();">`

▶ `<button onclick="myFunction();">Clique aqui!</button>`

# Pula linha

(pula\_js.html)

localhost diz

Você apertou  
o botão 1

OK

29

```
<!DOCTYPE html>
<html>
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
    <title>Untitled 1</title>
    <script type="text/javascript" src="../../js/jquery-3.5.1.min.js"></script>
    <script>
      function botao1()
      {
        alert("Você apertou \no botão 1");
      }
    </script>
  </head>
  <body>
    <input type="button" onclick="botao1()" value="Botão 1"/>
  </body>
</html>
```

# If

## (if\_js.html)

30

localhost/if\_alert.html

Clique no botão para exibir a caixa de confirmação.

Clique aqui

```
<!DOCTYPE html>
<html>
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
    <title>Untitled 1</title>
    <script type="text/javascript" src="../js/jquery-3.5.1.min.js"></script>
    <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>
  </head>
  <body>
    <p>Clique no botão para exibir a caixa de confirmação.</p>
    <button onclick="funcao1()">Clique aqui</button>
    <p id="demo"></p>
  </body>
</html>
```

localhost diz

Escolha um valor!

OK

Cancelar

# Prompt

(prompt\_js.html)

31

← → ↻ ⓘ localhost/prompt\_alert.html

Clique para exibir.

Clique aqui

```
<!DOCTYPE html>
<html>
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
    <title>Untitled 1</title>
    <script type="text/javascript" src="../../js/jquery-3.5.1.min.js"></script>
    <script>
      function myFunction() {
        var x;
        var idade=prompt("Digite sua idade:");
        if (idade!=null) {
          x="Idade: " + idade + " anos.";
          document.getElementById("demo").innerHTML=x;
        }
      }
    </script>
  </head>
  <body>
    <p>Clique para exibir.</p>
    <button onclick="myFunction()">Clique aqui</button>
    <p id="demo"></p>
  </body>
</html>
```

localhost diz

Digite sua idade:

OK

Cancelar

# JavaScript

## Janelas Popup

32

Funções para  
criação de janelas  
popup:

### Alerta

- `alert("Texto a ser exibido");`

### Confirmação

- `confirm("Texto a ser exibido em janela OK/Cancel");`

### Entrada de dado

- `prompt("Label do campo de entrada", "valor padrão");`

Janelas que possuem o botão Cancel, quando acionado retorna valor null



# JavaScript Janelas Popup

(popup\_js.html)

```
<!DOCTYPE html>
<html>
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
    <title>Untitled 1</title>
  </head>
  <body>
    <script type="text/javascript">
      var nome = prompt("Seu nome");
      if (nome != null && nome != "")
        document.write("Oi " + nome);
      else
        document.write("Oi anônimo!");
    </script>
  </body>
</html>
```

# Write - escrevendo na página

(write\_js.html)

Vamos usar o Write

11

34

```
<!DOCTYPE html>
<html>
<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type">
<title>Write</title>
</head>
<body>
  <p>Vamos usar o Write</p>
  <script>
    document.write(5 + 6);
  </script>
</body>
</html>
```

# Write - escrevendo na página

(write2\_js.html)

Vamos usar o Write

Soma

35

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta content="text/html; charset=utf-8" http-equiv="Content-Type">
```

```
<title>Write</title>
```

```
</head>
```

```
<body>
```

```
  <p>Vamos usar o Write</p>
```

```
  <button type="button" onclick="document.write(5 + 6)">Soma</button>
```

```
</body>
```

```
</html>
```

11

# Write Detalhes



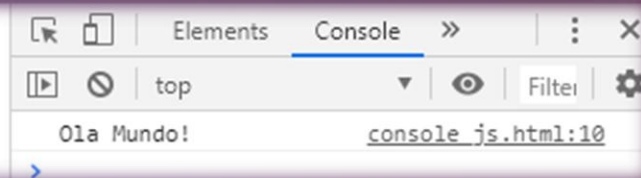
36

- ▶ Usando `document.write ()` após o carregamento de um documento HTML, **todos os HTML existentes** serão **excluídos**
- ▶ O método `document.write ()` deve ser **usado apenas para testes**.

# Console

(console\_js.html)

Aperte F12 e veja o console



37

```
<!DOCTYPE html>
<html>
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
    <title>Console Log</title>
  </head>
  <body>
    <p>Aperte F12 e veja o console</p>
    <script>
      console.log("Ola Mundo!");
    </script>
  </body>
</html>
```

# innerHTML – escrevendo dentro de um elemento

(inner\_js.html)

**Minha Página**

Você chegou aqui!

6

```
<!DOCTYPE html>
<html>
<head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
  <title>InnerHTML</title>
</head>
<body>
  <h1>Minha Página</h1>
  <p id="demo"></p>
  <p id="conta"></p>
  <script>
    document.getElementById("demo").innerHTML = "Você chegou aqui!";
    document.getElementById("conta").innerHTML = 5+1;
  </script>
</body>
</html>
```

# JavaScript – Eventos

(evento\_js.html)



39

```
<!DOCTYPE html>
<html>
<head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
  <title>Troca imagem com Mouse IN/OUT</title>
</head>
<body>
  
</body>
</html>
```

# JavaScript – Eventos em partes

(evento2\_js.html)

```
<!DOCTYPE html>
<html>
  <head>
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
    <title>Troca imagem com Mouse IN/OUT</title>
    <script src="event.js"></script>
  </head>
  <body>
    
  </body>
</html>
```



# JavaScript – Eventos em partes

(event.js)

```
/* Conteúdo do arquivo event.js */  
  
function mouseOver()  
{  
    document.imagem.src = "image/cat02.jpg";  
}  
function mouseOut()  
{  
    document.imagem.src = "image/cat01.jpg";  
}
```

# Escrevendo na página

(write3\_js.html)

```
<!DOCTYPE html>
<html>
<head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
  <title>Untitled 1</title>
</head>
<body>
  <script lang="JavaScript">
    document.write('Minha primeira página Javascript');
  </script>
</body>
</html>
```

# Funções

(funcao\_js.html)

```
<!DOCTYPE html>
<html>
<body>
  <h2>Funções</h2>
  <p>Chamando Minha Função</p>
  <p id="demo"></p>
  <script>
    function minhafuncao(a, b) {
      return a * b;
    }
    document.getElementById("demo").innerHTML = window.minhafuncao(10, 2);
  </script>
</body>
</html>
```

# Data

(data\_js.html)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Data de Hoje</title>
  <script>
    let d = new Date();
    alert("Hoje é " + d);
  </script>
</head>
<body>
</body>
</html>
```

# Data 2

(data2\_js.html)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Hoje</title>
</head>
<body>
  <script>
    let d = new Date();
    document.body.innerHTML = "<h1>Hoje é " + d + "</h1>"
  </script>
</body>
</html>
```

# Data 3 completo

(data3\_js.html)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Hoje</title>
  <link rel="stylesheet" href="data3.css" />
</head>
<body>
  <script src="data3.js"></script>
</body>
</html>
```

# Data 3 completo

(data3.js)

```
let d = new Date();  
document.body.innerHTML = "<h1>Hoje é " + d + "</h1>"
```

# Data 3 completo

(data3.css)

```
body {  
    background-color: #0080ff;  
}  
  
h1 {  
    color: #fff;  
    font-family: Arial, Helvetica, sans-serif;  
}
```



# Data 3 completo

(escorrega\_js.html)

```
<!DOCTYPE html>
<html>
<head>
  <script type="text/javascript" src="../js/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function() {
      $("button").click(function() {
        $("p").slideToggle();
      });
    });
  </script>
</head>
<body>
  <p>Um parágrafo</p>
  <button>Toggle: escorrega sobre o elemento</button>
</body>
</html>
```

# Formulários

```
<!DOCTYPE html>
<html>
<head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <title>Formulário</title>
</head>
<body>
  <form name="cliente">
    <input name="nome" /> <br/>
    <input name="telefone" /> <br/>
    <input name="email" /> <br/>
  </form>
</body>
</html>
```

Nome:

Telefone:

E-mail: