



JavaScript

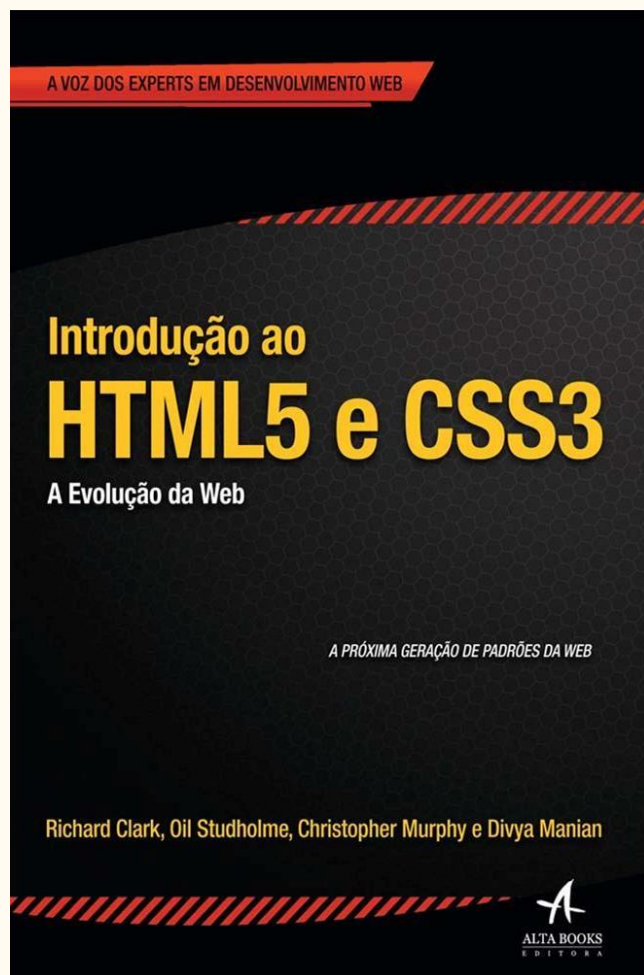
—

Prof. Victor Farias

v 1.1

Referências

w3schools.com



Introdução

JavaScript

- Linguagem de Programação padrão para páginas WEB
- JavaScript define o comportamento da página
- JavaScript tem várias utilidades:
 - Mudar conteúdo de um elemento
 - Mudar atributos HTML
 - Mudar estilos CSS
 - Esconder elementos
 - Mostrar elementos
 - Adicionar elementos

Tag <script>

- Código JavaScript deve ser colocado dentro de um elemento **script**
- Pode ser colocado no <head> ou no <body>
- É preferível colocar os scripts no final da página, pois carregamento e compilação podem atrasar a renderização da página

```
<script>
```

```
document.getElementById("demo").innerHTML = "My First JavaScript";
```

```
</script>
```

Tag <script>

- Também é possível importar um .js externo

```
<script src="myScript.js"></script>
```

- Vantagens:
 - Separa código JS e HTML
 - Facilita manutenção
 - Navegador pode deixar .js em cache

Sintaxe JavaScript



Slide Externo

HTML DOM



HTML DOM

- DOM = Document Object Model
- DOM define um padrão para acessar documentos
- HTML DOM é o modelo padrão e interface de programação para HTML
- Define:
 - Elementos HTML como objetos
 - As propriedades de elementos HTML
 - Métodos para acessar elementos HTML
 - Eventos para todos os elementos HTML

Elementos DOM

- Recuperar elemento por id
 - Método **getElementById()**
 - Se elemento não existir, myElement será nulo

```
var myElement = document.getElementById("intro");
```

- Recuperar elemento por nome da tag
 - Método `getElementsByTagName()`

```
var x = document.getElementById("main");
```

```
var y = x.getElementsByTagName("p");
```

```
// Recupera elemento com id="main" e, depois, recupera  
todos elementos <p> dentro dele
```

Elementos DOM

- Recuperar elementos por classe

- Método `getElementsByClassName()`

```
var x = document.getElementsByClassName("intro");
```

- Recuperar elemento por seletor CSS

- Método **`querySelectorAll()`**

```
var x = document.querySelectorAll("p.intro");
```

Manipulando HTML

- **document.write()**

- Sobrescreve html da página

```
document.write(Date());
```

- **Propriedade innerHTML**

- Muda conteúdo HTML de um elemento HTML

```
document.getElementById("p1").innerHTML = "New text!";
```

- **Manipulando atributos**

- `document.getElementById(id).attribute=new value`

```
document.getElementById("myImage").src =  
"landscape.jpg";
```

Manipulando CSS

- Mudando CSS

- `document.getElementById(id).style.property=new style`

```
document.getElementById("p2").style.color = "blue";
```

- Nomes das propriedades CSS compostas separados por hífen viram CamelCase no JS

- propriedade background-color ->
`object.style.backgroundColor="#00FF00"`

Manipulando Classes

- **API classList**
- **Adicionando classe**
 - `el.classList.add('classOne', 'classTwo');`
- **Removendo classe**
 - `el.classList.remove('classOne');`
- **Verificar existência de classe**
 - `if(el.classList.contains('classFour') == true){...}`
- **Alternar classe**
 - `el.classList.toggle('classThree')`

Eventos HTML

- Código JS pode ser disparado a partir de eventos
 - Quando um usuário clica o mouse
 - Quando a página é carregada
 - Quando uma imagem é carregada
 - Quando um mouse passa sobre um elemento
 - Quando um campo de entrada muda
 - Quando um formulário HTML é submetido
 - Quando o usuário aperta uma teclas

Eventos HTML

- onclick - no clique

- HTML - `<button onclick="displayDate()">Try it</button>`
- JS - `document.getElementById("myBtn").onclick = displayDate;`

- onload - no carregamento

- `<body onload="checkCookies()">`

- onchange - mudança de estado (muito usado em fomulários)

- `<input type="text" id="fname" onchange="upperCase()">`

- onmouseover - quando o mouse se desloca sobre o elemento

- onmouseout - quando o mouse se desloca para fora do elemento

- onmousedown - quando o botão do mouse é pressionado

- onmouseup - quando o botão do mouse é solto

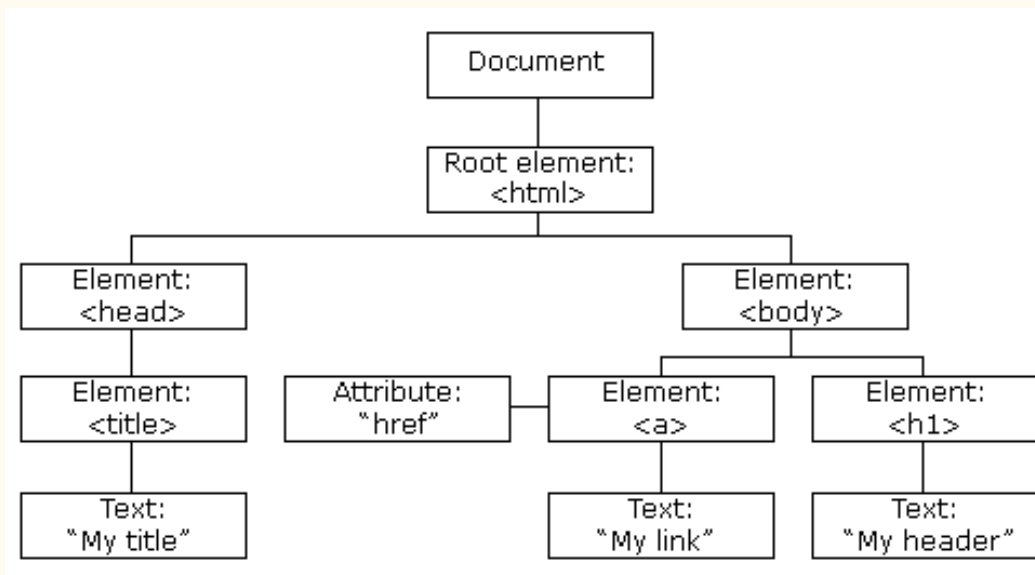
- onclick - quando a ação de clicar em um elemento é

HTML DOM EventListener

- É possível adicionar *listeners* de eventos
 - `element.addEventListener(event, useCapture);`
 - ex: `element.addEventListener("click", function(){ alert("Hello World!"); });`
 - Nota: Não usar prefixo “on”, use “click” no lugar de “onclick”
- Também é possível remover *listeners*
 - `element.removeEventListener("mousemove", myFunction);`

Navegação DOM

- De acordo com o padrão HTML DOM W3C, tudo no documento HTML é um nó
 - O documento inteiro é um nó de documento
 - Cada elemento HTML é um nó de elemento
 - O texto dentro de cada elemento HTML é um nó de texto
 - Cada atributo HTML é um nó de atributo
 - Todos os comentários são nós de comentários



Relação entre nós

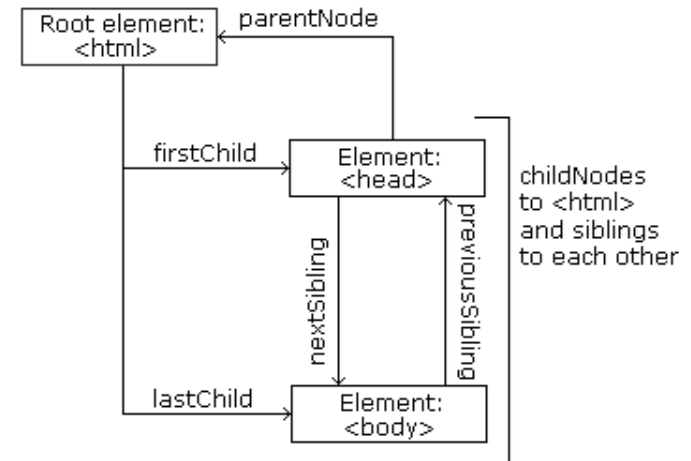
- Os nós estão dispostos em um árvore de modo hierárquico

```
<html>

  <head>
    <title>DOM Tutorial</title>
  </head>

  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>

</html>
```



Navegando entre Nós

- É possível usar as seguintes propriedades para navegar entre os nós
 - parentNode
 - childNodes[nodenumber]
 - firstChild
 - lastChild
 - nextSibling
 - previousSibling
- Nós raiz
 - document.body - Body do documento
 - document.documentElement - Documento inteiro

Nós HTML DOM

- Criar novos nós HTML.

- `document.createElement("p")` para criar nó de elemento
- `document.createTextNode("This is new.");` para criar nó de texto
- `appendChild(node);` para adicionar elemento a um elemento já existente

```
<div id="div1">  
<p id="p1">This is a paragraph.</p>  
<p id="p2">This is another paragraph.</p>  
</div>
```

```
<script>  
var para = document.createElement("p");  
var node = document.createTextNode("This is new.");  
para.appendChild(node);
```

```
var element = document.getElementById("div1");  
element.appendChild(para);  
</script>
```

This is a paragraph.

This is another paragraph.

This is new.

Lista de Nós HTML DOM

- `getElementsByTagName()` retorna um node list
 - Node list é uma estrutura que parece uma lista mas não é
 - É possível acessar elementos usando índice

Percorrendo lista:

```
var myNodelist = document.getElementsByTagName("p");

var i;

for (i = 0; i < myNodelist.length; i++) {

    myNodelist[i].style.backgroundColor = "red";

}
```

Perguntas?

Prof. Victor Farias