Os exemplos de código usados neste material
foram baseados no material da W3C

Sistemas Web I

Aula 4

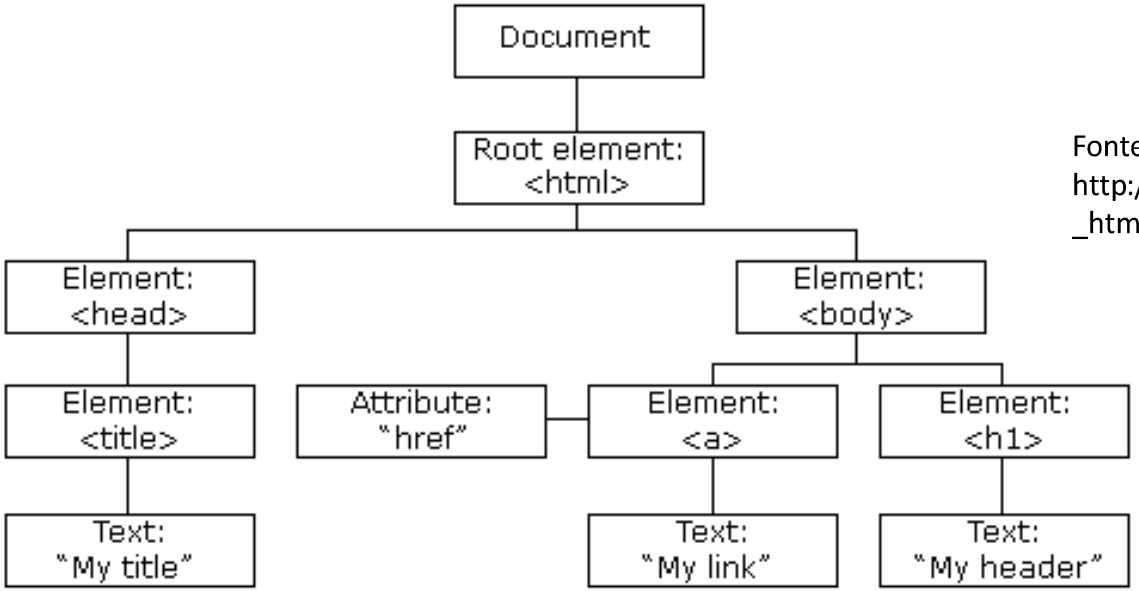
JavaScript

Tiago Cruz de França

tcruz.franca@gmail.com

DOM (*Document Object Model*)

- **Acesse todos os elementos do HTML**
  - Os navegadores criam um modelo de objeto do documento ao abrir uma página HTML
    - **Uma "árvore" de objetos com todo conteúdo da página**



Fonte da imagem:
http://www.w3schools.com/js/js_htmldom.asp

DOM e JavaScript

- **Com o modelo de objeto é possível dar dinamismo as páginas HTML com Javascript**
  - Mudar elementos da página
  - Mudar atributos HTML e estilos CSS na página
    - **Pode remover e adicionar elementos e atributos HTML**
  - Realizar tarefas após um evento ocorrer na página
  - Também pode gerar um evento

DOM

- **Padrão W3C**
  - Definição de padrão para acessar documentos
  - Dividida em 3 partes:
    - **Core DOM – modelo padrão para todos os documentos**
    - **XML DOM - modelo padrão para documentos XML**
    - **HTML DOM - modelo padrão para docs HTML**

Métodos DOM – O que você pode fazer...

- **DOM define métodos (ações) e propriedades (valores)**
- **DOM possui uma interface de programação**
  - Todos elementos HTML são vistos como objetos
  - Ações/propriedades são realizados/acessado por meio da interface

Exemplo

```
<!DOCTYPE html>
<html>
<body>

<p id=“aula"></p>

<script>
document.getElementById(“aula").innerHTML = “Web1!";
//getElementById é um método (mais comum utilizado, pega o id de um elemento)
//innerHTML é uma propriedade (maneira mais fácil de usar uma propriedade)
</script>

</body>
</html>
```

DOM

- **A página Web é o documento HTML DOM (ver imagem da árvore)**
  - É a raiz(pai) de todos os outros objetos da página HTML
  - Sem comece com "*document*" quando quiser navegar e pagar algum elemento da página

Mudando, Adicionando e Apagando Elementos

Método	Descrição
element.innerHTML=	Change the inner HTML of an element
element.attribute=	Change the attribute of an HTML element
element.setAttribute(attribute,value)	Change the attribute of an HTML element
element.style.property=	Change the style of an HTML element
document.createElement()	Create an HTML element
document.removeChild()	Remove an HTML element
document.appendChild()	Add an HTML element
document.replaceChild()	Replace an HTML element
document.write(text)	Write into the HTML output stream

Adicionando Eventos e Encontrando Objetos

Método	Descrição
document.getElementById(id).onclick=function(){code}	Adding event handler code to an onclick event

Método	Descrição
document.anchors	Returns all <a> elements that have a name attribute
document.baseURI	Returns the absolute base URI of the document
document.body	Returns the <body> element
document.cookie	Returns the document's cookie
document.documentElement	Returns the <html> element
document.forms	Returns all <form> elements

Lista Completa: http://www.w3schools.com/js/js_htmldom_document.asp

Elementos DOM

Procurando elementos HTML

- Possibilidades:
  - Encontrar elemento HTML pelo id, nome, nome da classe, seletor CSS ou coleções de objetos HTML

```
<!DOCTYPE html>
<html>
<body>
<p class=“web”> Método getElementById</p>
<p id=“exem“></p>
<script>
myElement = document.getElementById(“exem");
document.getElementById(“exem").innerHTML =  “Aula de DOM web-1" +
myElement.innerHTML;
</script>
</body>
</html>
```

Substitua o Código anterior e teste

```
//Pegando Elementos

//Por nome
var x = document.getElementsByTagName("p");

//Pelo nome da classe
var x = document.getElementsByClassName(“web");

//Por seletor CSS
var x = document.querySelectorAll("p.web");

//Iterando DOM
var x = document.forms["frm-web"];
var text = "";
var i;
for (i = 0; i < x.length; i++) {
   text += x.elements[i].value + "<br>";
}
document.getElementById(“exem").innerHTML = text;
```

Mudando Valor de Um Atributo

```
<!DOCTYPE html>
<html>
<body>

<img id="image" src="smiley.gif" width="160" height="120">

<script>
document.getElementById("image").src = "landscape.jpg";
</script>

<p>The original image was smiley.gif, but the script changed it to landscape.jpg</p>

</body>
</html>
```

Mudando o Estilo da Página

```
<!DOCTYPE html>
<html>
<body>

<p id="p1">Web</p>
<p id="p2">Estrutura</p>

<script>
document.getElementById("p2").style.color = "blue";
document.getElementById("p2").style.fontFamily = "Arial";
document.getElementById("p2").style.fontSize = "larger";
</script>

<p>The paragraph above was changed by a script.</p>

</body>
</html>
```

Usando Eventos

```
<!DOCTYPE html>
<html>
<body>

<h1 id="id1">Primeiro evento</h1>

<button type="button"
onclick="document.getElementById('id1').style.color = 'red'">
Click Me!</button>

</body>
</html>
```

Usando Eventos

```
<!DOCTYPE html>
<html>
<body>

<h1 onclick="changeText(this)">Click on this text!</h1>

<script>
function changeText(id) {
   id.innerHTML = "Ooops!";
}
</script>

</body>
</html>
```

Usando Eventos

```
<html>
<head>
</head>
<body>

<p>Observe a função na chamada do evento</p>
<button id=“btnAula">Teste e veja</button>
<p id=“exemplo"></p>

<script>
document.getElementById(“btnAula").onclick = displayDate;
function displayDate() {
   document.getElementById(“exemplo").innerHTML = Date();
}
</script>
</body>
</html>
```

Mais exemplos de Eventos

```
<script>
function checkCookies() {
    var text = "";
    if (navigator.cookieEnabled == true) {
        text = "Cookies are enabled.";
    } else {
        text = "Cookies are not enabled.";
    }
    document.getElementById("demo").innerHTML = text;
}
</script>
```

```
<script>
function mOver(obj) {
    obj.innerHTML = "Thank You"
}

function mOut(obj) {
    obj.innerHTML = "Mouse Over Me"
}
</script>
```

EventListener

O método addEventListener()

```
<script>
document.getElementById("myBtn").addEventListener("click", displayDate);

function displayDate() {
   document.getElementById("demo").innerHTML = Date();
}
</script>
```

- A sintaxe é:
  - element.addEventListener(event, function, useCapture);

Navegação DOM

- **Qualquer elemento no documento é um nó**
  - Relações hieráquicas
    - **Dentro do documento o primeiro nó é o <html>**

```
...
<h1 id="intro">My First Page</h1>

<p id="demo">Hello!</p>

<script>
var myText = document.getElementById("intro").childNodes[0].nodeValue;
document.getElementById("demo").innerHTML = myText;
</script>
...
```

Navegação DOM – Elemento Raiz

```
…
<script>
alert(document.body.innerHTML);
</script>

Ou

<script>
alert(document.documentElement.innerHTML);
</script>
…
```

Nós

Adicionando um elemento

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

<script>
//crie o elemento
var para = document.createElement("p");
var node = document.createTextNode("This is new.");
para.appendChild(node);
//depois adicione a outro elemento já existente
var element = document.getElementById("div1");
element.appendChild(para);
</script>
```

Nós

Removendo um elemento

- Você deve conhecer o nó pai do elemento que se deseja remover

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

<script>
var parent = document.getElementById("div1");
var child = document.getElementById("p1");
parent.removeChild(child);
</script>
```

Nós

Substituindo um elemento

```
<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 parent = document.getElementById("div1");
var child = document.getElementById("p1");
parent.replaceChild(para,child);
</script>
```

NodeList

- **Um nó permite que seus elementos sejam acessados por um índice**

```
<!DOCTYPE html>
<html>
<body>
<p>Hello World!</p>
<p>The DOM is very useful!</p>
<p id="demo"></p>
<script>
var myNodelist = document.getElementsByTagName("p");
document.getElementById("demo").innerHTML =
"The innerHTML of the second paragraph is: " +
myNodelist[1].innerHTML;
</script>
</body>
</html>
```

Tamanho da Lista

```
<!DOCTYPE html>
<html>
<body>

<p>Hello World!</p>
<p>How many paragraphs in this document?</p>
<p>This example demonstrates the length property of a nodelist.</p>
<p id="demo"></p>

<script>
var myNodelist = document.getElementsByTagName("p");
document.getElementById("demo").innerHTML = myNodelist.length;
</script>

</body>
</html>
```

Exercício

- **Comente este código descrevendo o que é feito em cada linha**

```
<!DOCTYPE html>
<html>
<body>
<p>This is a p element</p>
<p>This is also a p element.</p>
<p>This is also a p element - Click the button to change the background color of all p elements in this document.</p>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
   var myNodelist = document.getElementsByTagName("p");
   var i;
   for (i = 0; i < myNodelist.length; i++) {
      myNodelist[i].style.backgroundColor = "red";
   }
}
</script>
</body>
</html>
```