

08

Introdução às Tecnologias Web

JavaScript (3/3)

A yellow square with the letters 'JS' in a bold, black, sans-serif font.

Perguntas

```
for (j=0; j<5; j++) { console.log(j); }
```

```
c.marca = "BMW";
```

```
c["marca"] = "BMW"; // Iguais ?
```

```
var a = [3, 5, 7];
```

```
a.push(9);      // Conteúdo de a ?
```

Qual o significado?

```
3 function relógio (h, m, s) {  
4     this.hora = h;  
5     this.min = m;  
6     this.sec = s;  
7     this.totalMinutos = function () {  
8         return (this.hora * 60 + this.min);  
9     }  
10 }  
11  
12 function forObject () {  
13     var rel1 = new relógio(10, 45, 34);  
14     var rel2 = new relógio(5, 8, 13);  
15  
16     rel1.miliSeg=55;  
17  
18     // Relógio 1  
19     for(var prop in rel1){  
20         console.log(rel1[prop]);  
21     }  
22     alert(rel1.totalMinutos());  
23  
24     // Relógio 2  
25     for(var prop in rel2){  
26         console.log(rel2[prop]);  
27     }  
28     alert(rel2.totalMinutos());  
29 }  
30  
31 window.onload = forObject;
```

Qual o significado?

```
3 function relógio (h, m, s) {  
4     this.hora = h;  
5     this.min = m;  
6     this.sec = s;  
7     this.totalMinutos = function () {  
8         return (this.hora * 60 + this.min);  
9     }  
10 }  
11  
12 function forObject () {  
13     var rel1 = new relógio(10, 45, 34);  
14     var rel2 = new relógio(5, 8, 13);  
15  
16     rel1.miliSeg=55;  
17  
18     // Relógio 1  
19     for(var prop in rel1){  
20         console.log(rel1[prop]);  
21     }  
22     alert(rel1.totalMinutos());  
23  
24     // Relógio 2  
25     for(var prop in rel2){  
26         console.log(rel2[prop]);  
27     }  
28     alert(rel2.totalMinutos());  
29 }  
30  
31 window.onload = forObject;
```

Resumo Aula Anterior

Ciclos

while, do /while, for, for / in

Objetos

Guardam vários valores

Têm Propriedades e Métodos

São manipulados por referência

Arrays

Caso particular de objeto

Têm vários métodos para os manipular

Sumário

Objeto Window

Manipulação do DOM

Eventos

01

O OBJETO WINDOW

O Objeto Window

Principal ponto de entrada do JS

Representa a janela do browser

Tem como identificador: `window`

Objeto Window == Objeto Global

Objetos, funções e variáveis globais pertencem a Window

Não é necessário usar o identificador window

```
window.alert("Teste");  
alert("Teste");
```

Propriedades e Métodos

location – URL da página corrente

`location.href` – Devolve a URL da página

`location.assign("http://
www.w3schools.com")` – Carrega uma nova
página

screen – Informação sobre o ecrã

`screen.width` – Largura do ecrã em pixéis

`screen.height` – Altura do ecrã em pixéis

Propriedades e Métodos

open ("novaPag") – Abre uma nova janela

close () – Fecha a janela atual

alert ("mensagem") – mostra uma mensagem num
diálogo

confirm ("mensagem") – Pede confirmação

Devolve um booleano

true -> botão OK ; false -> botão Cancel

prompt ("mensagem") – Diálogo para pedir dados

OK -> Devolve dados ; Cancel -> Devolve null

Propriedades e Métodos

setInterval (func, tempo) – Executa uma função repetidamente

setTimeout (func, tempo) – Executa uma função ao fim de um intervalo de tempo

clearInterval (timer) – Cancela um setInterval

```
var ff=function(){alert ("Timer!!");}
```

```
var myTimer = setInterval(ff, 3000);  
setTimeout (ff, 8000);  
clearInterval (myTimer);
```

Propriedades e Métodos

document – Objeto que representa o conteúdo apresentado na janela (página Web)

Tem vários métodos (ver mais à frente)

Ex:

```
document.getElementById ("myID");
```

Devolve o elemento do documento cujo id é igual a "myID"

02

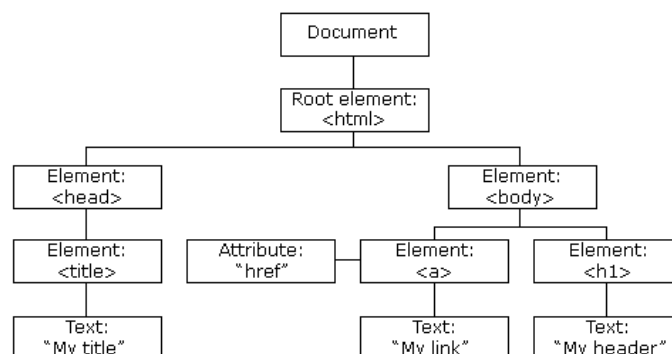
DOM

DOM

DOM - Document Object Model

Define estrutura lógica de uma página web

Criado pelo browser quando carrega a página



HTML DOM

O DOM é um modelo de objetos e uma API para HTML

Define:

- Os elementos HTML como objetos
- As propriedades dos elementos HTML
- Os métodos para aceder aos elementos HTML
- Os eventos para todos os elementos HTML

Em resumo, o DOM é um standard para:

Aceder, alterar, adicionar e apagar elementos HTML

Métodos e Propriedades do DOM

No DOM todos os elementos são objetos

Métodos – Ações que se podem fazer

Ex: Adicionar ou remover um elemento

Propriedades – Valores que podemos usar ou guardar

Ex: Alterar o conteúdo de um elemento HTML

Exemplo

```
<body>
  <h1>My First Page</h1>
  <p id="demo"></p>
  <script>
    document.getElementById("demo").
      innerHTML = "Hello World!";
  </script>
</body>
```

Método

Propriedade

Nós do DOM

Tudo numa página HTML é um Nó no DOM

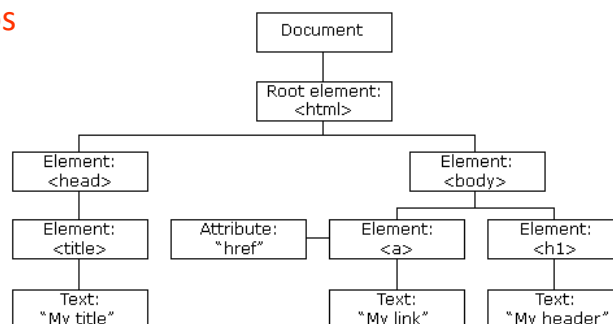
A página

Cada um dos elementos HTML

O texto dentro de um elemento HTML

Os atributos de um elemento HTML

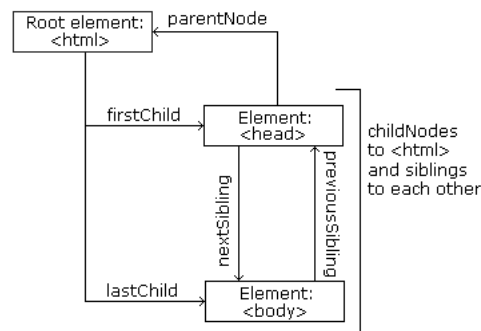
Os comentários



Relações entre Nós do DOM

Nós do DOM têm uma hierarquia entre si

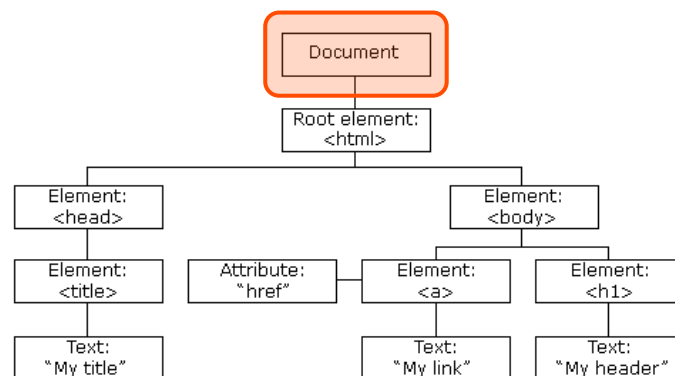
Relações: **parent, child, sibling**



O Objeto Document

Representa a página Web

É o “pai” de todos os elementos na página



Navegação entre Nós do DOM

Propriedades para navegar nos nós

parentNode – Nó pai

childNodes[nodenum] – Um dos filhos

firstChild – Primeiro filho

lastChild – Último filho

nextSibling – Irmão seguinte

previousSibling – Irmão anterior

Aceder ao conteúdo de um elemento

Conteúdo dos elementos HTML guardado em nó do DOM

Aceder ao nó do elemento HTML não chega

Necessário aceder ao nó DOM com o conteúdo

```
var elem = document.getElementById("myH1");
```

```
var text1 = elem.childNodes[0].nodeValue;
```

```
var text2 = elem.firstChild.nodeValue;
```

```
var text3 = elem.innerHTML;
```

JavaScript + DOM Permitem

Alterar

- Todos os elementos HTML
- Os atributos de todos os elementos HTML
- Todos os estilos CSS

Remover

- Elementos HTML
- Atributos de elementos HTML

Adicionar

- Novos elementos HTML
- Atributos a elementos HTML

Reagir a todos os elementos HTML (eventos)

Criar novos eventos HTML

**ENCONTRAR
ELEMENTOS HTML**

Encontrar elementos HTML

Usando **id**

Usando o **nome** da **tag**

Usando o nome da **classe**

Usando um **seletor** CSS

Usando **coleções** de objetos

Encontrar usando id

document.getElementById()

Devolve um elemento com base no seu ID

Se não existir devolve null

```
<p id="intro">Hello World!</p>
```

```
<p>Outro parágrafo!</p>
```

```
<p id="demo">Parágrafo três</p>
```

```
elem1 = document.getElementById("intro");
```

```
elem2 = document.getElementById("demo");
```

Encontrar usando tag

document.getElementsByTagName()

Devolve os elementos de uma tag (array)

Se não existir devolve null

```
<p id="intro">Hello World!</p>
<p>Outro parágrafo!</p>
<p id="demo">Parágrafo três</p>
```

```
elems = document.getElementsByTagName("p");
elem2 = elems[1];
```

Encontrar usando id + tag

```
<div id="cont">
<p id="intro">Hello World!</p>
<p>Outro parágrafo!</p>
<p id="demo">Parágrafo três</p>
</div>
```

```
elemC = document.getElementById("cont");
myPars = elemC.getElementsByTagName("p");
p1 = myPars[2];
```

Encontrar usando class

document.getElementsByClassName()

Devolve os elementos de uma classe (array)

Se não existir devolve null

```
<p class="azul">Hello World!</p>  
<p >Outro parágrafo!</p>  
<p class="azul">Parágrafo três</p>
```

```
elem1 = document.  
    getElementsByClassName("azul")[1];
```

Encontrar usando CSS

document.querySelectorAll()

Devolve os elementos que respeitam o seletor CSS (array)

Usam os seletores das regras CSS

```
<p class="azul">Hello World!</p>  
<p >Outro parágrafo!</p>  
<p class="azul">Parágrafo três</p>
```

```
elems = document.querySelectorAll("p.azul");  
elem1 = elems[0];
```

Encontrar usando Coleções de Objetos

document.images

Devolve todas as imagens da página (array)

document.links

Devolve todos os links da página (array)

document.title

Devolve o título da página

etc.

http://www.w3schools.com/js/js_htmlDOM_elements.asp

ALTERAR ELEMENTOS HTML

Alterar Conteúdo HTML

Propriedade **innerHTML**

Permite alterar o conteúdo de um elemento HTML

```
<p id="azul">Hello World!</p>
```

```
document.getElementById("azul").  
  innerHTML = "Novo Texto!";
```

ou

```
el = document.getElementById("azul");  
el.innerHTML = "Novo Texto!";
```

Alterar Atributo elemento HTML

Encontrar elemento HTML

Alterar os seus atributos

```

```

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

Alterar Estilo elemento HTML

Encontrar elemento HTML

Alterar os seus estilos

```
<p id="p1">Hello World!</p>
```

```
document.getElementById("p1").  
    style.color = "green";
```

CRIAR + REMOVER ELEMENTOS HTML

Criar elementos HTML

Criar o elemento

Criar o seu conteúdo

Adicionar o conteúdo ao elemento

Adicionar o elemento a um “pai”

Exemplo criação elemento HTML

```
<div id="div1">
  <p id="p1">Parágrafo 1</p>
  <p id="p2">Parágrafo 2</p>
</div>

var newP = document.createElement("p");
var cont = document.createTextNode("Parágrafo 3");
newP.appendChild(cont);

var elem = document.getElementById("div1");
elem.appendChild(newP);
```

Exemplo insertBefore

```
<div id="div1">
  <p id="p1">Parágrafo 1</p>
  <p id="p2">Parágrafo 2</p>
</div>

var newP = document.createElement("p");
var cont = document.createTextNode("Parágrafo 1.5");
newP.appendChild(cont);

var elem = document.getElementById("div1");
var pos = document.getElementById("p2");
elem.insertBefore(newP, pos);
```

Remover elementos HTML

Saber qual é o pai

Saber qual o elemento (filho) a remover

Remover o elemento (filho) do pai

Exemplo remoção elemento HTML

```
<div id="div1">
  <p id="p1">Parágrafo 1</p>
  <p id="p2">Parágrafo 2</p>
</div>

var pai = document.getElementById("div1");
var filho = document.getElementById("p1");
pai.removeChild(filho);
ou
var filho = document.getElementById("p1");
filho.parentNode.removeChild(filho);
```

Exemplo substituição de elemento

```
<div id="div1">
  <p id="p1">Parágrafo 1</p>
  <p id="p2">Parágrafo 2</p>
</div>

var newP = document.createElement("p");
var cont = document.createTextNode("Parágrafo 3");
newP.appendChild(cont);

var pai = document.getElementById("div1");
var oldP = document.getElementById("p1");
pai.replaceChild(newP, oldP);
```

03

EVENTOS

Eventos

O DOM permite executar código quando um evento acontece

Eventos são gerados pelo browser quando algo acontece aos elementos HTML

Um elemento é clicado

Uma página é carregada

Campos de entrada alteram-se

O rato mexe-se sobre um elemento

Uma tecla é premida

etc.

Lista de Eventos

Rato

onclick
onmousedown
onmouseup
onmouseenter
onmouseleave
onmouseover
onmousemove

Teclado

onkeydown
onkeyup
onkeypress

Forms

onchange
onfocus
oninput
etc.

http://www.w3schools.com/jsref/dom_obj_event.asp

Atribuir Eventos a elementos HTML

```
<h1 onclick="mudaTexto()">  
  Carrega-me!</h1>
```

Event Handler

ou

```
<h1 id="myH1">Carrega-me!</h1>  
  
document.getElementById("myH1").  
  onclick = mudaTexto;
```

Event Handler

Event Handler executado ao carregar no H1

Event Listeners

addEventListener()

Acrescenta um *Event Handler* a um elemento HTML

Não substitui os *Event Handlers* já associados

Pode-se acrescentar vários *Event Handlers*

Mesmo tipo ou tipos diferentes de eventos

Permite separar o código HTML do JavaScript

Melhor legibilidade do código

removeEventListener()

Remove um *Event Listener*

Adicionar Event Handlers

```
<button id="myBtn">Click me!</button>
```

Evento

```
document.getElementById("myBtn").  
  addEventListener("click",  
    function(){alert("Olá!");});
```

EventHandler executado ao carregar no botão

Event Handler

Lista de Eventos (Event Listener)

Com Event Listener não se usa o prefixo on

Rato

click
mousedown
mouseup
mouseenter
mouseleave
mouseover
mousemove

Teclado

keydown
keyup
keypress

Forms

change
focus
input
etc.

http://www.w3schools.com/jsref/dom_obj_event.asp

Adicionar Event Handlers

Event Handler : função com nome

```
element.addEventListener(  
    "click", myFunction);  
  
function myFunction () {  
    alert ("Olá!");  
}
```

Adicionar Event Handlers

Event Handler : função com parâmetros

```
var p1=6;
element.addEventListener("click",
    function(){ myFunction(p1); });

function myFunction (a) {
    alert ("Olá!" + a);
}
```

Adicionar Vários Event Handlers

Vários *Event Handlers* para o mesmo elemento

```
element.addEventListener(
    "click", myF1);

element.addEventListener(
    "mouseover", myF2);

element.addEventListener(
    "click", myF3);
```

04

RESUMINDO

Resumo

Objeto Window

Janela do browser = objeto global

Oferece vários métodos e propriedades

DOM

Estrutura lógica da página

Encontrar + Alterar + Criar + Remover elementos HTML

Eventos

Gerados quando algo acontece

Event Handler – Código executado quando acontece evento

Um elemento pode ter vários *Event Handlers*

Próxima Aula

