

FÁBRICA DE DESENVOLVIMENTO

FÁBRICA DE DESENVOLVIMENTO - 2018

Prof. Thiago T. I. Yamamoto

#14 - JAVASCRIPT



thiagoyama



thiagoyama@gmail.com

#14 - JAVASCRIPT

- Sintaxe
- Utilização
- Funções e Eventos
- DOM
- RegEx
- E etc.

- Javascript é uma linguagem de scripts muito popular na internet e funciona na maioria dos browsers;
- O principal objetivo do javascript é adicionar interatividade nas páginas;
- Não é necessário compilador para executar os scripts;
- Javascript NÃO tem nada a ver com Java, exceto algumas similaridades de sintaxe;
- Não é necessário “tipar” as variáveis;
- Javascript é Case Sensitive;

- Eventos
 - São utilizados quando algo acontece na pagina:
 - » Clique do mouse
 - » Carregamento da pagina
 - » Envio de informações
- Ler e Escrever
 - É possível escrever em conteúdo de elementos HTML e da mesma forma ler os seus valores.
- Informações do visitante
 - É possível detectar informações do visitante como versão de browser

- Cookies
 - É possível gravar e recuperar informações de cookies no computador do visitante.
- Validar dados
 - É possível validar informações antes mesmo delas serem enviadas ao servidor.

- A sintaxe do JS é definida pelas TAGs HTML

<script> ... </script>

- 3 formas de utilização do JS:
 - Em um evento da tag HTML (inline)
 - No cabeçalho (<head> e/ou <body>) de uma página HTML
 - Em um arquivo JS externo

UTILIZAÇÃO DO JAVASCRIPT

- Em um evento da tag HTML (inline)

```
<tag evento="javascript:/* script */;"></tag>
```

- Exemplo

```
<button onclick="javascript:alert("oi");" />
```

UTILIZAÇÃO DO JAVASCRIPT

- No cabeçalho <head>

```
<html>  
<head>  
  <script type="text/javascript">  
    alert("oi!")  
  </script>  
</head>  
...  
</html>
```

- No corpo <body>

```
<html>
<head>
  <title>JS</title>
</head>
<body>
  <script type="text/javascript">
    alert("oi!")
  </script>
</body>
</html>
```

- No arquivo externo JS:

```
<html>  
<head>  
  <script type="text/javascript" src="arquivo.js"> </script>  
</head>  
...  
</html>
```

- Em JS os comentários podem ser adicionados no código de duas maneiras:

- Linha única, representada por duas barras seguida do texto

// um exemplo de comentário numa linha única

- Bloco, que é representado por uma barra e um asterisco

/*

**este é um comentário em JS
em bloco**

***/**

- Em JS as variáveis não são tipadas, logo basta declará-las com “var”

```
var minhaVariavel = “Valor qualquer”;
```

```
var idade = 30;
```

- Em JS os operadores
 - Comparação
 - » **==** é igual a
 - » **!=** não é igual a
 - » **>** maior que
 - » **<** menor que
 - » **>=** maior ou igual a
 - » **<=** menor ou igual a

- Em JS os operadores

- Lógico

- » **&&** e

- » **||** ou

- » **!** não

- Em JS as condicionais são tratadas pelo bloco de código if em diferentes formatos

- Exemplo 1

```
if (condição) { ação; }
```

- Exemplo 2

```
if (condição) { ação; } else { ação; }
```

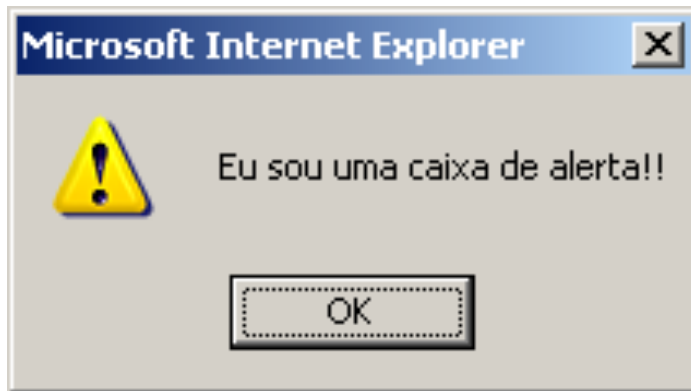
- Exemplo 3

```
if (condição) { ação; } else if { ação; } else { ação; }
```

CAIXAS DE MENSAGENS

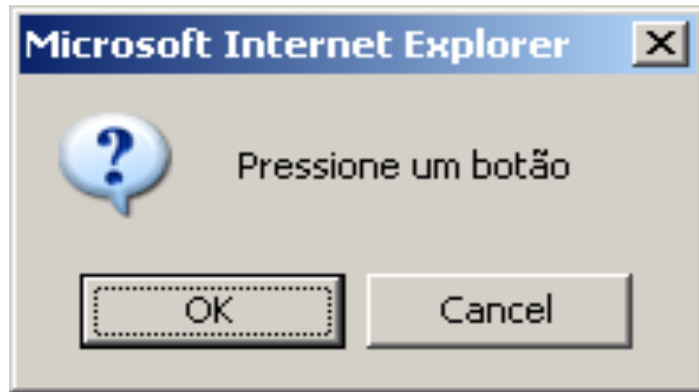
CAIXA DE MENSAGEM

- Em JS as caixas de mensagens são usadas para apresentar informações para o usuário. São elas:
 - Exemplo 1: Caixa de alerta



alert("Eu sou uma caixa de alerta!!");

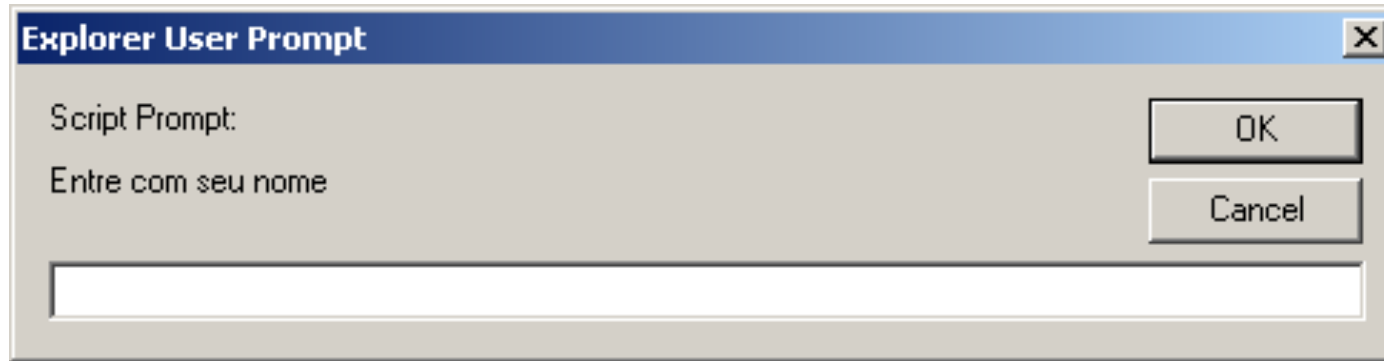
- Exemplo 2: Caixa de confirmação



confirm("texto qualquer");

- A caixa de confirmação é uma função que retorna **true** para o botão OK e **false** para Cancel.

- Exemplo 2: Caixa de texto



prompt("texto qualquer");

- A caixa de texto é uma função que retorna o **texto digitado** para o botão OK e **null** para Cancel.

FUNÇÕES

- Em JS as funções (functions) são úteis para que um código JS seja executado somente no momento que for requisitado.
- Uma função, contém códigos JS que serão executados por um evento HTML ou pela própria chamada da função.
- As funções podem ser definidas em qualquer posição da página, mas recomenda-se que fique no escopo da TAG <head>
- As funções também são executadas quando utiliza-se um arquivo .js
- Uma função é declarada da seguinte forma:

function nomeDaFuncao() { código }

function nomeDaFuncao(p1, p2, pN) { código }

- Em JS as funções podem ou não retornar valores, para isto, devemos utilizar a declaração **return** logo após a execução do código.

```
function nomeDaFuncao() {  
  
    var idade = 30;  
  
    return idade;  
  
}
```


EVENTOS

- Os eventos do HTML são usados para disparar ações nas páginas deixando-as mais interativas e dinâmicas.
 - Estes eventos podem ser: foco, alteração, envio, carregamento, descarregamento, clique de mouse e etc.
 - Cada evento acima citado, poder ter associado uma função de JS para que seja executada uma ação.
 - Os eventos são categorizados em
 - » Eventos do Windows, Elementos de formulários, Teclado e Mouse.

- Eventos do Windows

Atributo	Valor	Descrição
onload	<i>script</i>	Exec o script no load do documento
onunload	<i>script</i>	Exec o script no unload do documento

- Eventos de elementos do formulário

Atributo	Valor	Descrição
onchange	<i>script</i>	Exec o script quando um elemento é alterado
onsubmit	<i>script</i>	Exec o script quando o formulario é submetido
onreset	<i>script</i>	Exec o script quando o formulario tem seus valores reinicializado
onselect	<i>script</i>	Exec o script quando o elemento é selecionado
onblur	<i>script</i>	Exec o script quando o elemento perde foco
onfocus	<i>script</i>	Exec o script quando o elemento recebe o foco

- Eventos do teclado

Atributo	Valor	Descrição
onkeydown	<i>script</i>	Exec o script quando uma tecla é pressionada
onkeypress	<i>script</i>	Exec o script quando uma tecla é pressionada e liberada
onkeyup	<i>script</i>	Exec o script quando uma tecla é liberada

- Eventos do mouse

Atributo	Valor	Descrição
onclick	<i>script</i>	Exec o script no click do mouse
ondblclick	<i>script</i>	Exec o script no duplo click do mouse
onmousedown	<i>script</i>	Exec o script quando o botao do mouse é pressionado
onmousemove	<i>script</i>	Exec o script quando o ponteiro do mouse se move
onmouseout	<i>script</i>	Exec o script quando o ponteiro do mouse se move fora de um elemento
onmouseover	<i>script</i>	Exec o script quando o ponteiro do mouse se move sobre um elemento
onmouseup	<i>script</i>	Exec o script quando o botao do mouse é liberado

- Função JS

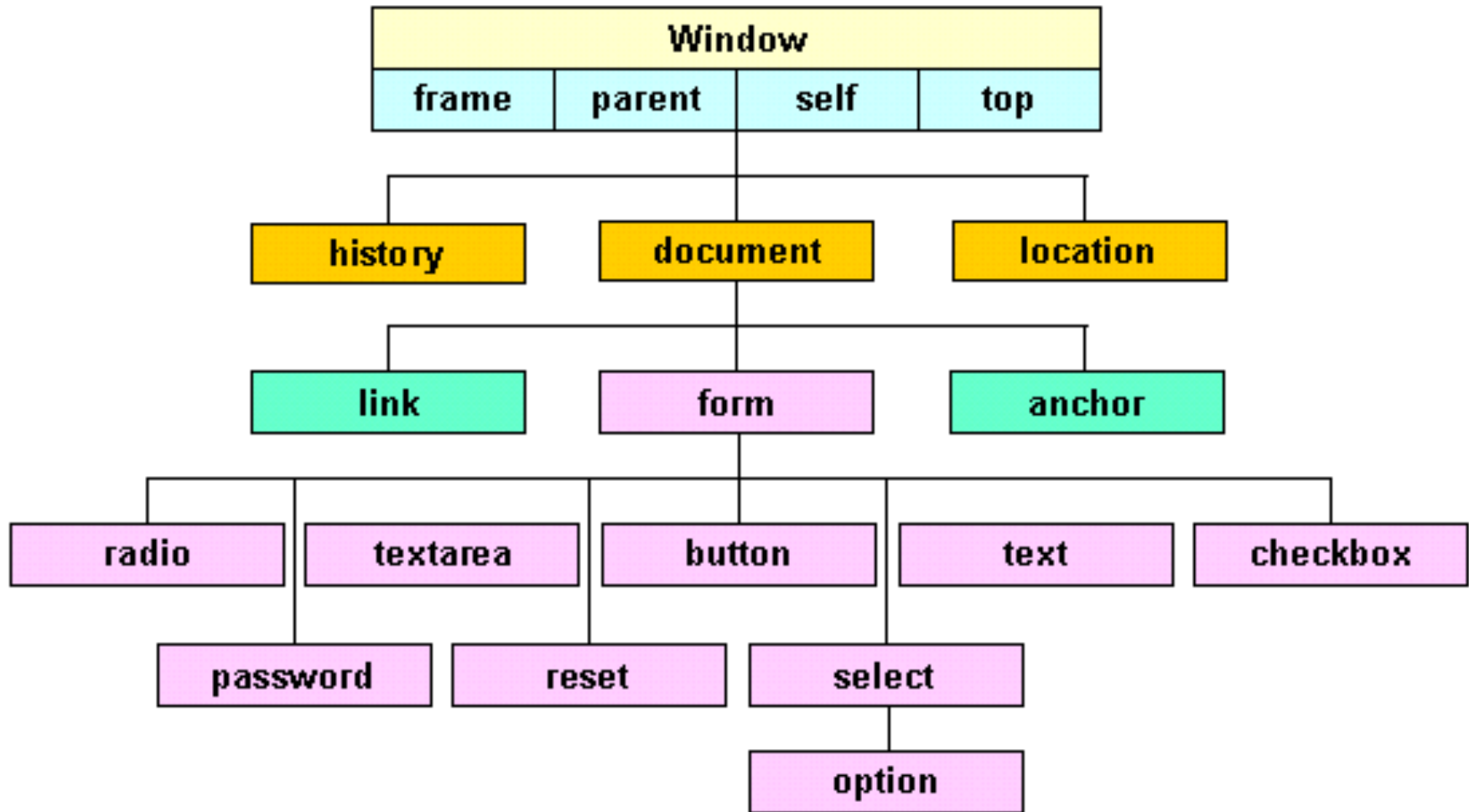
```
function trocaCorFundo(pNome)
{
    pNome.style.backgroundColor="#00FF00";
}
```

- Evento XHTML

```
<form action="#" method="post" name="form1">
<input type="text" name="nome" id="nome" size="30"
    onfocus="javascript:trocaCorFundo(this)" />
</form>
```

DOM

- Para interagir com HTML utilizando o JS precisamos primeiro conhecer o DOM.
 - Document Object Model – DOM é uma coleção de objetos para se fazer manipulação de documentos.
 - O DOM é uma recomendação do W3C e está dividida em 3 categorias
 - » Core DOM – Modelo padrão para qualquer estrutura de documento
 - » XML DOM – Modelo padrão para documentos XML
 - » HTML DOM – Modelo padrão para documentos HTML



- HTML DOM
 - Com o HTML DOM podemos, recuperar, alterar, adicionar e deletar elementos HTML
 - Segundo a especificação DOM todo documento HTML é considerado um node
 - » O documento como um todo é um node
 - » Um elemento HTML é um node
 - » Um texto do elemento HTML é um node

- Exemplo de HTML DOM

```
<html>  
  <head>  
    <title>DOM Tutorial</title>  
  </head>  
  <body>  
    <h1>DOM Lesson one</h1>  
    <p>Hello world!</p>  
  </body>  
</html>
```

- A tag **<html>** é o node root
- HTML node tem dois filhos node **<head>** e **<body>**
- **<body>** node possui dois nodes **<h1>** e **<p>**

- Texto do HTML
 - Os textos do HTML sempre serão armazenados no text node.
 - Neste exemplo `<h1>DOM Lesson one</h1>` o elemento `<h1>` possui o texto “DOM Lesson one”. Isto não quer dizer que este texto é um valor do elemento `<h1>`.
 - Em HTML DOM um texto de elemento pode ser acessado pela propriedade **innerHTML**

- innerHTML

- A forma mais rápida e fácil de recuperar ou alterar um conteúdo de um elemento HTML é utilizar a propriedade **innerHTML**

- Exemplo

```
function insereTexto(pTexto){  
    var texto = document.getElementById("divTexto");  
    texto.innerHTML = pTexto;  
}
```

```
<button onclick="insereTexto('FIAP')" >Clique aqui</button>  
<div id="divTexto"></div>
```

- Document DOM
 - Métodos
 - » **getElementById(id)** – Recupera o elemento pelo id especificado
 - » **getElementsByName(name)** – Recupera todos os elementos pelo nome especificado
 - » **getElementsByTagName(tagname)** – Recupera todos os elementos pelo nome da tag especificado
 - » **write() ou writeln()** – Escreve no documento HTML

- Propriedades

- » **innerHTML** - Recupera um texto de um elemento
- » **title** - retorna o título do documento
- » **URL** - retorna a URL do documento

- Coleções

- » **forms[]** - retorna coleção (array) com todos os forms do documento
- » **images[]** - retorna coleção com todos as imagens do documento
- » **links[]** - retorna coleção de todos os href do documento
- » **anchors[]** - retorna coleção de todos os links <a> do documento

- Exemplo de Document DOM

```
<script type="text/javascript">  
    document.write(document.title);  
    document.write(document.URL);  
</script>
```

- Form DOM

- Métodos

- » **submit()** – submete um formulário

- » **reset()** – Limpa os valores dos elementos do formulário

- Propriedades

- » **action** – Especifica ou recupera o destino do formulário

- » **method** – Especifica ou recupera a forma de envio das informações

- » **name** – Respecifica ou retorna o nome do formulário

- » **length** – Retorna o número de elementos do formulário

- Exemplo de Form DOM

```
<script type="text/javascript">
```

```
function formSubmit()  
{  
    document.getElementById("myForm").submit()  
}
```

```
</script>
```

```
<form id="myForm" action="form_action.asp" method="get">
```

```
...
```

```
</form>
```

- Propriedade **value**

Email:

OK

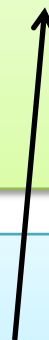
Inputs de texto: Retorna o valor inserido pelo usuário.

```
<script type="text/javascript">
function mostrarResultado(){
  var x=document.getElementById("email").value;
  alert(x);
}
</script>
```

javascript

```
<form>
  Email: <input type="text" id="email" />
  <button type="button" onclick="mostrarResultado()">OK</button>
</form>
```

HTML



- Propriedade **checked** (checkbox e radio)

```
<script type="text/javascript">
function check(){
  document.getElementById("check1").checked=true;
}
function uncheck(){
  document.getElementById("check1").checked=false;
}
function mostrarResultado(){
  if (document.getElementById("check1").checked == true){
    alert("Marcado!");
  }else{
    alert("Desmarcado!");
  }
}
</script>
```

javascript**Resultado:**☐ Gosta da aula de DIG?

Marcar

Desmarcar

Resultado

```
<form><input type="checkbox" id="check1" />Gosta da aula de DIG?</form>
<button onclick="check()">Marcar</button>
<button onclick="uncheck()">Desmarcar</button>
<button onclick="mostrarResultado()">Resultado</button>
```

HTML

Copyright © 2018 - Prof. Me. Thiago T. I. Yamamoto

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

“Se você traçar metas absurdamente altas e falhar, seu fracasso será muito melhor que o sucesso de todos” – James Cameron