



# PROGRAMAÇÃO PARA INTERNET RICA

## FUNDAMENTOS DO JAVASCRIPT

Prof. Dr. Daniel Caetano

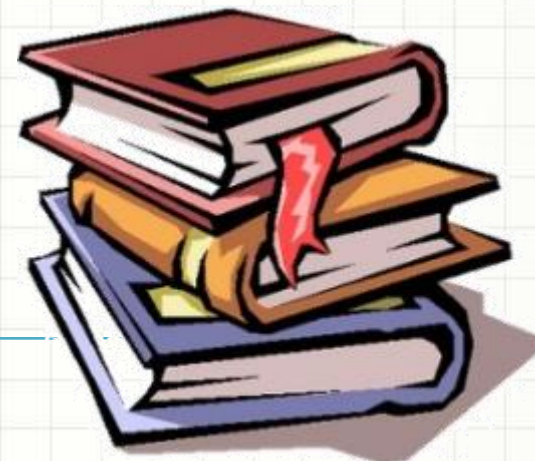
2013 - 1

# Objetivos

- Apresentar a natureza da linguagem JavaScript
- Apresentar o JavaScript com ferramenta de manipulação de elementos do HTML
- Apresentar os estilos e eventos mais comuns utilizados com o JavaScript
- **TRABALHO A ONLINE!**



# Material de Estudo



## Material

## Acesso ao Material

Notas de Aula

<http://www.caetano.eng.br/>  
(Aula 6)

Apresentação

<http://www.caetano.eng.br/>  
(Aula 6)

Material Didático

Aprenda a Criar Páginas Web c/ HTML, páginas 609 a 648

Google

+“JavaScript” +tutorial

Web Sites

<http://www.w3.org/>



# **JAVASCRIPT: UMA LINGUAGEM INTERPRETADA**

# Introdução

- Páginas Web Dinâmicas: Capazes de Mudar
- Como implementar?
- Através de pequenos programas...
  - Associados ao código da página
  - Associados a ações do usuário
- Quais linguagens permitem fazer isso?

# Introdução

- Linguagens mais usadas na Web

Nome	Empresa	Tipo	Similar	Execução
JavaScript	Mozilla	Interpretada	Java/C++	Cliente
PHP	PHP	Interpretada	C++/Java	Servidor
ASP	Microsoft	Interpretada	Visual Basic	Servidor
JSP	Oracle	“Compilada”	Java	Servidor
Servlets	Oracle	“Compilada”	Java	Servidor
ASP .Net	Microsoft	“Compilada”	Visual Basic .Net	Servidor

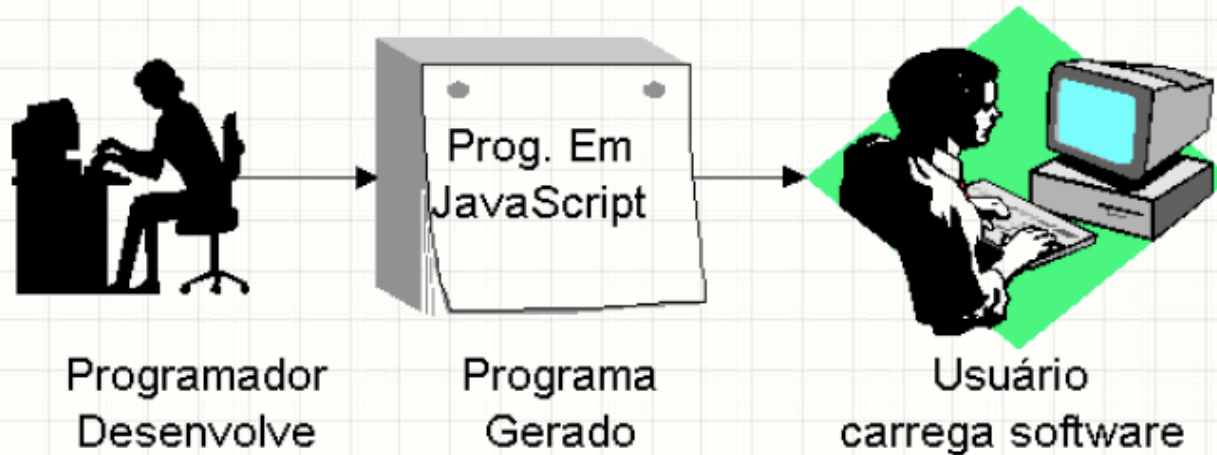
# Linguagem Interpretada

- Linguagens podem ser de dois tipos
  - Compiladas
  - Interpretadas
- Compiladas



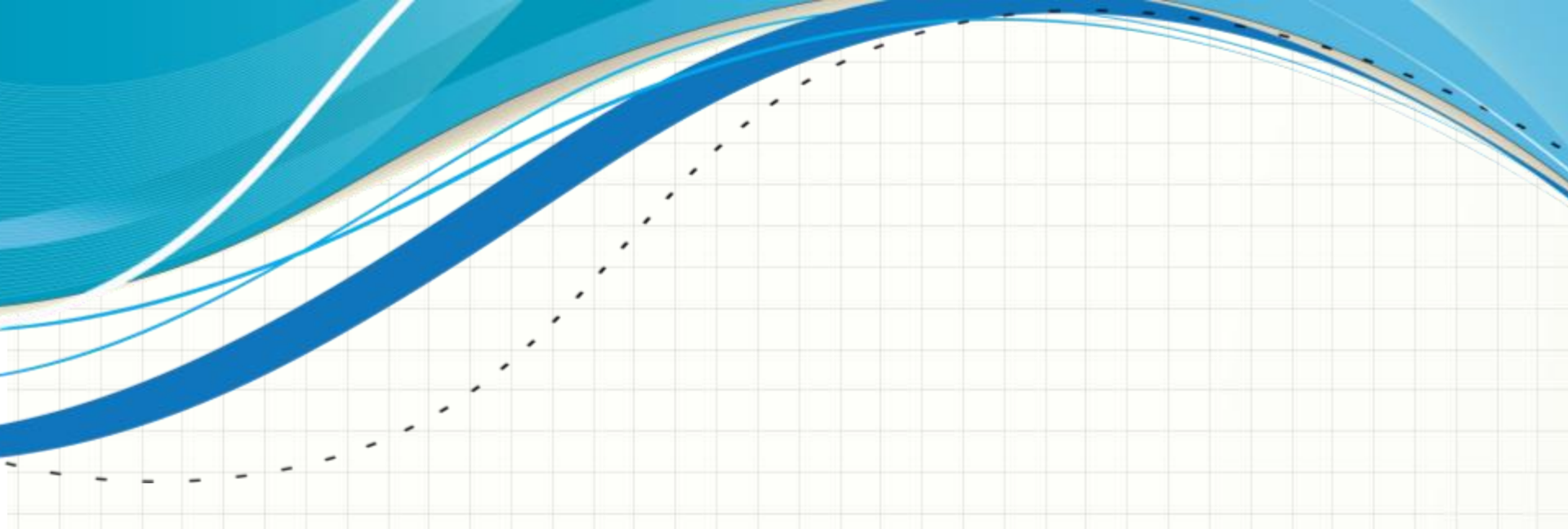
# Linguagem Interpretada

- Interpretadas



- Diferença na distribuição
  - Comprar um livro traduzido
  - Ir ao Oscar com um intérprete
- JavaScript: intérprete é o navegador!

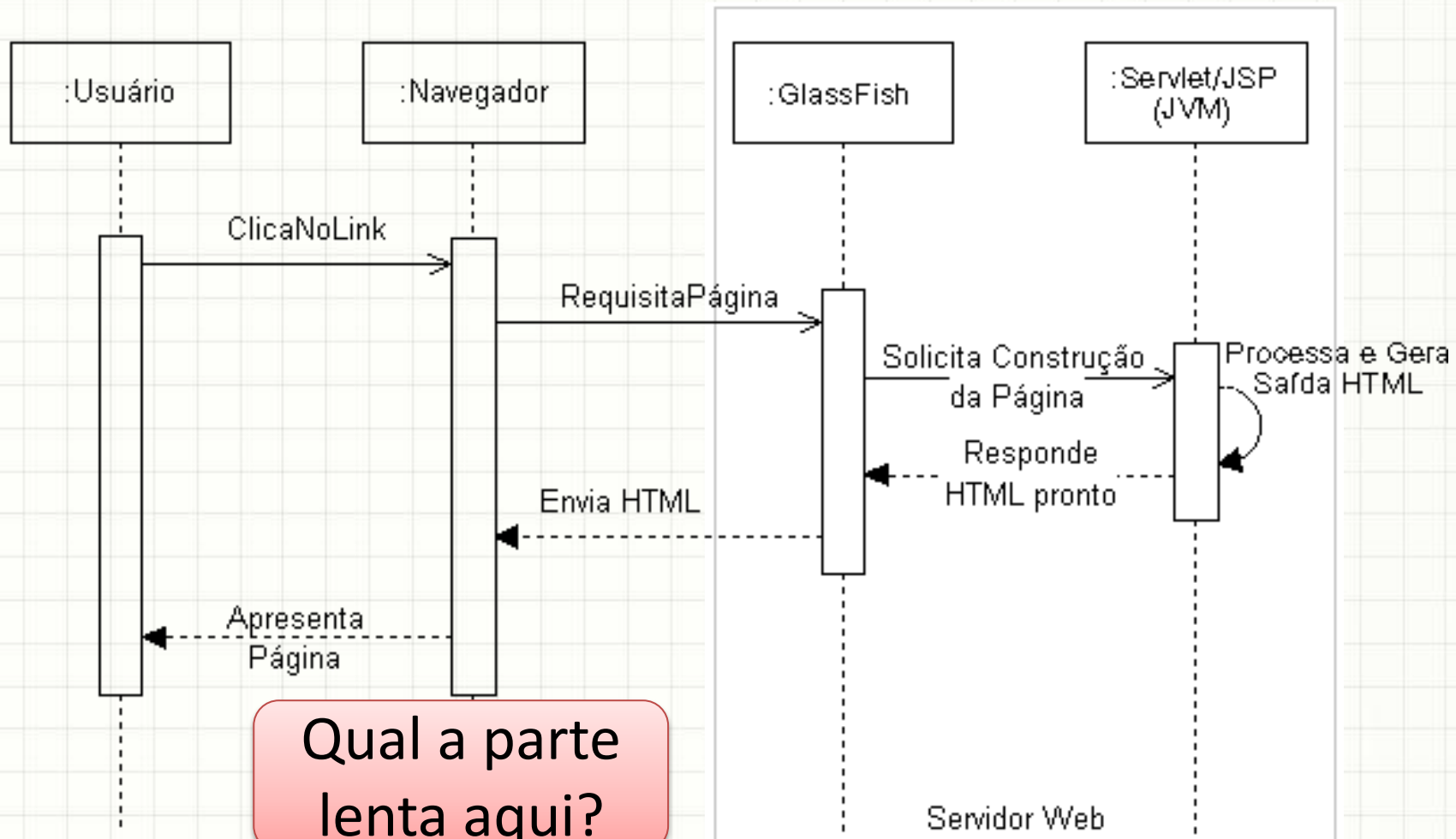




**CLIENT SIDE X  
SERVER SIDE**

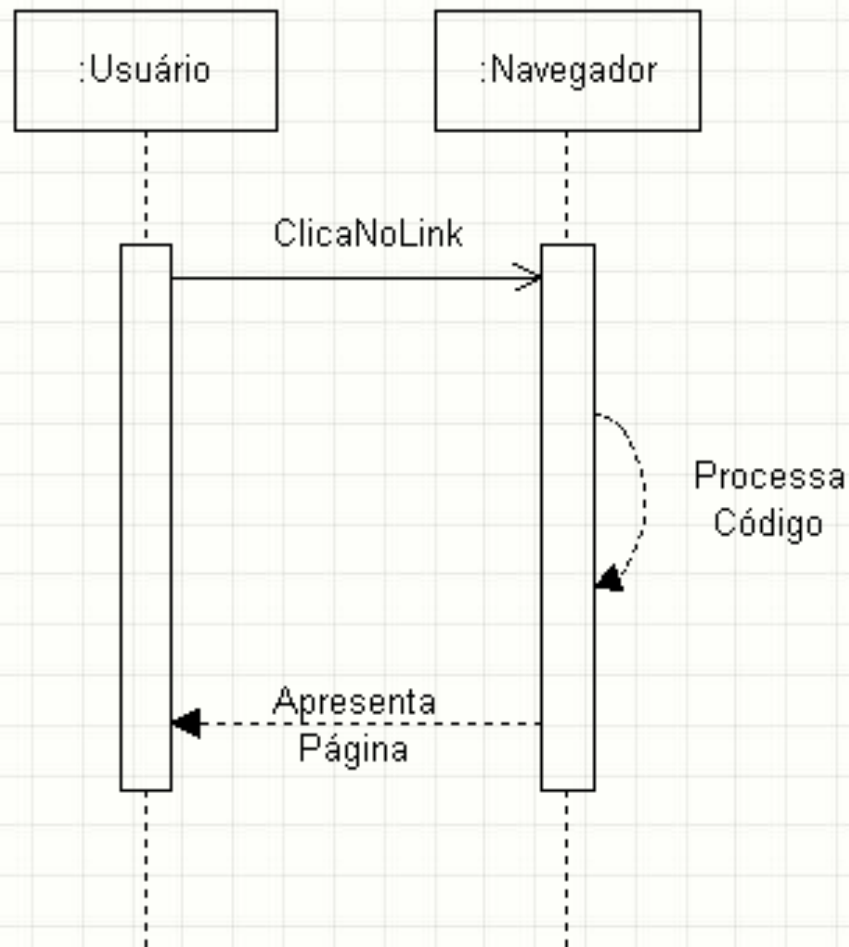
# Server Side x Cliente Side

- Exemplo de Processamento Server Side



# Server Side x Cliente Side

- Exemplo de Processamento Client Side



# Server Side x Cliente Side

- Exemplo de Processamento Client Side



# Server Side x Cliente Side

- E a segurança...
- Client Side é mais fraco!
  - Usuário tem acesso ao código
  - Usuário pode modificar o código



Manage Accounts [v] Logout Preferences Inbox File Transfer Mode [v] FAQ Tutorial

### My computer

C:\Documents and Settings\hkim\Desktop

File Name	File Size(KB)	Modified Time
z-size.txt	1	03/21/2007 05:21 PM
ys888 5.01.lnk	2	03/21/2007 00:30 AM
rsB-260107_RemoveP1....	36	01/31/2007 08:53 AM
windows_xp_keyboard_...	93	01/12/2007 09:31 AM
web2swatches.zip	2	01/31/2007 04:49 PM
UpstartAqueous02(2).zip	149	03/08/2007 09:19 AM
Untitled-2.html	1	03/16/2007 10:38 AM
Trx01_Icon_Set.rar	6854	12/29/2006 01:55 PM
Thumbs.db	925	03/21/2007 02:44 PM
theme.psd	1766	03/08/2007 02:21 PM
testcode.html	1	03/21/2007 09:44 AM
test.html	2	02/06/2007 09:44 AM
test table.html	3	01/22/2007 03:22 PM
soccer-testing01.psd	484	03/21/2007 03:13 PM
smooth slide.htm	3	12/18/2006 05:12 PM
testin.exe	9772	03/01/2007 11:10 AM

### My Gspace

gs:/

File Name	File Size(KB)
Runshot11.zip	234
New Folder 01/	0

### Transfers


Clear

File Name	Type	From	To	Progress	Status
Runshot11.zip (...)	Upload	C:\Documents ...	gs:/	<div style="width: 100%; height: 10px; background-color: green;"></div>	✓ Completed

### Status

Copy / Paste / Paste as / Paste as text

Send To operation :



Name : Runshot11  
 Type: File  
 File Size(in Kb): 234  
 Number of splits: 1  
 From: hpmouse@g  
 Remote Path : gs/

Status: **Logged** Logged in User Name: **hpmouse@gmail.com** Gmail Used Space: **358** Gb

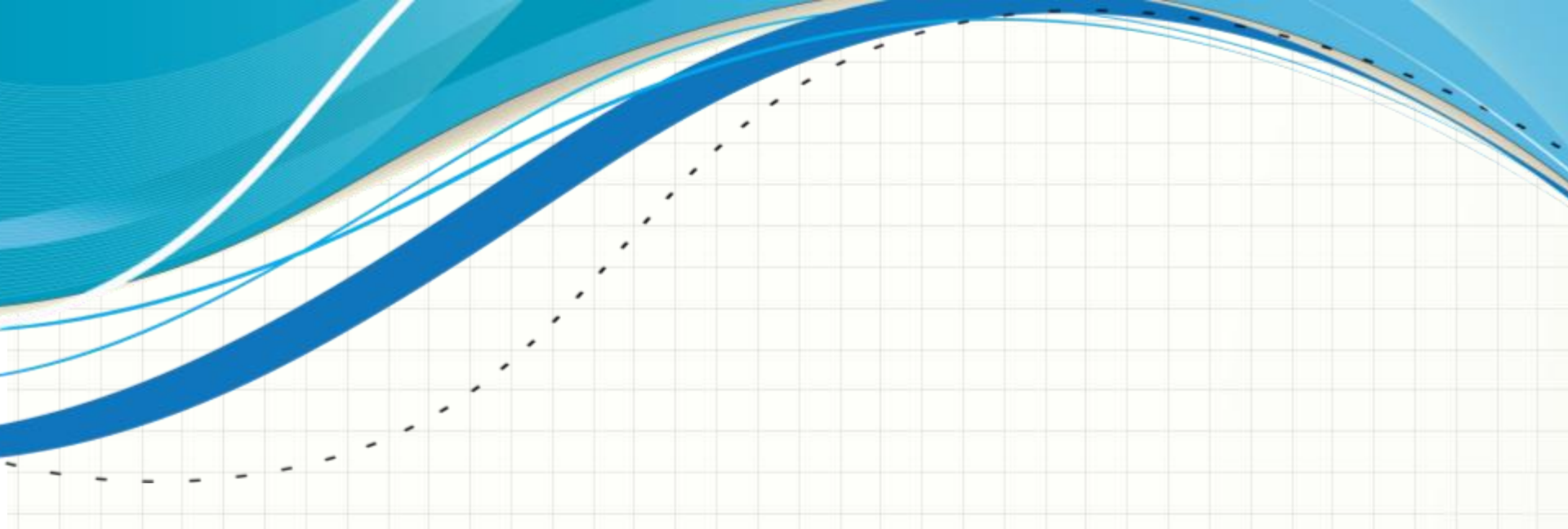
Done

Gspace  
(Firefox)

# Server Side x Cliente Side

- E a segurança...
- Client Side é mais fraco!
  - Usuário tem acesso ao código
  - Usuário pode modificar o código
- Transferir para o lado cliente é bom...
  - Mas requer cuidado!
  - Apenas se não comprometer a segurança!





# **A LINGUAGEM JAVASCRIPT**



# O JavaScript

- A linguagem JavaScript em si...
  - É muito parecida com o C/C++
  - É muito parecida com o Java
- Principal diferença:
  - Não há tipos de variáveis
- Quase tudo é objeto
  - Os elementos **fazem coisas**



# O JavaScript

- Usaremos para...
  - Mudar cores
  - Mudar textos
  - Mudar posições...
- Implementar lógica...
  - De menus
  - De configurações
- Gerar solicitações
  - AJAX



# Como Integrar JavaScript e XHTML?

- Similar a CSS
  - Não vamos inserir no XHTML

- Processo de Criação

1. Criar uma página com alguns elementos
2. Criar a função a associar a algum evento
3. Associar a função ao evento de um elemento

<script>

<!--

```
function ga(o,e){
if (document.getElementById(
a=o.id.substring(1,
p = "m.
r = "m.
g = e.target;
(a) {
g.id;
```

# Exemplo: Criando o XHTML

- Criar uma página XHTML **teste.html** contendo este corpo:

```
<body>
  <p>
    <input type="button" value="Cor" />
  </p>
  <p>Teste</p>
</body>
```

# Exemplo: Criando o JavaScript

- Criar um arquivo chamado **efeitos.js**
- Neste arquivo, vamos criar uma **função** chamada **mudaCorDeFundo()**.

```
function mudaCorDeFundo() {  
  
}
```

- Para modificar o fundo, precisamos acessar o elemento **document.body**

# Exemplo: Criando o JavaScript

- Vamos mudar a propriedade **style.backgroundColor** do corpo

```
function mudaCorDeFundo() {  
    document.body.style.backgroundColor = "black";  
}
```

- Observe que o que é **background-color** no CSS, no JavaScript virou **backgroundColor** !
- Agora, precisamos associar o **JS** ao **XHTML**

# Exemplo: Associando o JavaScript

- Voltemos ao arquivo **teste.html**
- No cabeçalho, dentro da seção `<head>`, inclua as seguintes linhas:

```
<script type="text/javascript" src="efeitos.js">  
</script>
```

- Agora carregue a página e veja o que ocorre quando você aperta o botão...

# Exemplo: Associando a Função

- **NADA?!?**
- Claro! Precisamos dizer QUANDO aquela função será executada
- Isso é feito associando-a ao evento **onclick** do botão que criamos no HTML!
- O jeito mais “simples” de fazer isso é...

```
<input type="button" value="Cor"  
      onclick="javascript:mudaCorDeFundo()" />
```



# Exemplo: Associando a Função

- Mas isso **mistura código com XHTML!**
- Tem como fazer diferente?
- Claro!
- Mas, para isso, precisamos dar um ID para o botão:

```
<input type="button" value="Cor"  
      id="bmudacor" />
```

# Exemplo: Associando a Função

- Voltemos agora ao **efeitos.js...**
  - Precisamos dizer ao navegador que a função **mudaCorDeFundo** está associada ao evento **onclick** do elemento de ID **bmudacor**.
1. Primeiro iremos criar uma referência para o elemento do botão
  2. Depois iremos associar a função ao evento do elemento

# Exemplo: Pegando o elemento

- Arquivo **efeitos.js**

```
var botao = document.getElementById("bmudacor");
```

```
function mudaCorDeFundo() {  
    document.body.style.backgroundColor = "black";  
}
```

# Exemplo: Pegando o elemento

- Arquivo **efeitos.js**

```
var botao = document.getElementById("bmudacor");
```

```
function mudaFundo() {
```

```
    document.getElementById("back");  
}
```

Pede ao documento que retorne o elemento com o ID especificado!

# Exemplo: Associando função ao Evento

- Arquivo **efeitos.js**

```
var botao = document.getElementById("bmudacor");  
botao.onclick = mudaCorDeFundo;
```

```
function mudaCorDeFundo() {  
    document.body.style.backgroundColor = "black";  
}
```

# Exemplo: Associando função ao Evento

- Arquivo **efeitos.js**

```
var botao = document.getElementById("bmudacor");
```

```
botao.onclick = mudaCorDeFundo;
```

```
function mudaCorDeFundo() {
```

Evento **onclick** do elemento armazenado na variável **botao**

```
};
```

# Exemplo: Associando função ao Evento

- Arquivo **efeitos.js**

```
var botao = document.getElementById("bmudacor");  
botao.onclick = mudaCorDeFundo;
```

```
function mudaCorDeFundo
```

Nome da função a associar,  
**SEM ()** ao final!

# Exemplo: Associando função ao Evento

- Isso funciona...
- Às vezes...
- Por quê?





# Exemplo: Associando função ao Evento

**Executa antes do botão existir  
e retorna “vazio”**

Por que!



# Como Ocorre o Processamento

```
<html>
```

```
<head>
```

```
<title>...
```

```
<script src="sc.js..."
```

```
</head>
```

```
<body>
```

```
...
```

```
<input id="b1"...
```

```
</body>
```

```
</html>
```

# Como Ocorre o Processamento

```
<html>
```

```
<head>
```

```
<title>...
```

```
<script src="sc.js..."
```

```
</head>
```

```
<body>
```

```
...
```

```
<input id="b1"...
```

```
</body>
```

```
</html>
```

# Como Ocorre o Processamento

```
<html>  
<head>  
<title>...  
<script src="sc.js...  
</head>  
<body>  
...  
<input id="b1"...  
</body>  
</html>
```

# Como Ocorre o Processamento

```
<html>
<head>
<title>...
<script src="sc.js..."
</head>
<body>
...
<input id="b1"...
</body>
</html>
```



```
var botao = get...("b1")
botao.onclick = ...
```

# Como Ocorre o Processamento

```
<html>
<head>
<title>...
<script src="sc.js...
</head>
<body>
...
<input id="b1"...
</body>
</html>
```

```
var botao = get...("b1")
botao.onclick = ...
```

# Como Ocorre o Processamento

```
<html>
<head>
<title>...
<script src="sc.js...
</head>
<body>
...
<input id="b1"...
</body>
</html>
```

```
var botao = get...("b1")
botao.onclick = ...
```

# Como Ocorre o Processamento

Qual o valor de  
botao?

```
<html>
<head>
<title>...
<script src="sc.js...
</head>
<body>
...
<input id="b1"...
</body>
</html>
```

```
var botao = get...("b1")
botao.onclick = ...
```



# Exemplo: Associando função ao Evento

- Solução: colocar em uma função...
  - ...que só seja executada depois que o XHTML carregar
- Arquivo **efeitos.js**

```
function init() {  
    var botao=document.getElementById("bmudacor");  
    botao.onclick = mudaCorDeFundo;  
}  
function mudaCorDeFundo() {  
    document.body.style.backgroundColor = "black";  
}
```

# Exemplo: Associando função ao Evento

- XHTML pronto: evento **window.onload**
- Arquivo **efeitos.js**

```
window.onload = init;
```

```
function init() {
```

```
    var botao=document.getElementById("bmudacor");
```

```
    botao.onclick = mudaCorDeFundo;
```

```
}
```

```
function mudaCorDeFundo() {
```

```
    document.body.style.backgroundColor = "black";
```

```
}
```

# Exemplo: Associando função ao Evento

- XHTML pronto: evento **window.onload**
- Arquivo **efeitos.js**

```
window.onload = init;
```

```
function init {
```

```
    var botao = document.getElementById("bmudacor");
```

```
    botao.onclick = mudaCorDeFundo;
```

Evento acionado quando o navegador termina de processar **todo o XHTML**

```
ack";
```

# Exemplo: Mudando Texto

- Agora que já vimos como associar funções a eventos, será que é possível mudar um texto?
- Arquivo **teste.html**

```
<body>
  <p>
    <input type="button" value="Cor"
      id="bmudacor" />
  </p>
  <p id="ajuda">Aqui é a ajuda!</p>
</body>
```

# Exemplo: Mudando Texto

- Para mudar o texto, alteramos a propriedade **innerHTML** do elemento

```
var texto = document.getElementById("ajuda") ;  
texto.innerHTML = "Texto da Ajuda!";
```

- Usualmente isso é associado ao evento **onmouseover**, que é quando o mouse está sobre o elemento
- **onmouseout** indica quando o mouse saiu do elemento



**ATIVIDADE**

# Atividade

1. Crie uma página com um botão que mude a cor de fundo da tela para azul, com texto amarelo
2. Acrescente um parágrafo para um texto de ajuda, que indique “Clique aqui para mudar a cor”, quando o mouse passar por cima do botão e volte o texto normal quando o mouse sair do botão



# EVENTOS COMUNES



# Eventos Comuns

- Document, Window e Body

Elementos	Evento	Quando executa
document body	onload	Quando o documento <b>inicia</b> o carregamento
document body	onunload	Quando é iniciada a <b>finalização</b> do documento (o usuário foi para outra página)
window	onload	Quando o documento foi <b>totalmente</b> carregado e processado

# Eventos Comuns

- Forms

Elemento	Evento	Quando executa
Campos de form	onchange	Quando o conteúdo do elemento foi alterado
Campos de form	onfocus	Quando o elemento ganhou foco
Combo box	onselect	Quando um item foi selecionado (combo box)
form	onsubmit	Quando formulário for enviado

# Eventos Comuns

- Teclado (quase todos os elementos)

Elemento	Evento	Quando executa
*	onkeydown	Quando uma tecla é apertada
*	onkeyup	Quando uma tecla é solta
*	onkeypress	Quando uma tecla for apertada e solta

# Eventos Comuns

- Mouse (quase todos os elementos)

El.	Evento	Quando executa
*	onclick	Quando o elemento for clicado
*	ondblclick	Quando o elemento for duplamente clicado
*	onmousemove	Quando o mouse se mover sobre o elemento
*	onmouseout	Quando o mouse sair de cima do elemento
*	onmouseover	Quando o mouse passar sobre o elemento
*	onmouseup	Quando o botão do mouse for solto sobre o elemento
*	onmousedown	Quando o botão do mouse for apertado sobre o elemento



# **PROPIEDADES COMUNS**

# Propriedades Comuns

- Plano de Fundo

Propriedade	O que altera
backgroundColor	Cor de fundo do elemento
backgroundImage	Imagem de fundo do elemento

- Textos

Propriedade	O que altera
color	Muda cor do texto
fontSize	Muda tamanho da fonte
textAlign	Muda alinhamento do texto
textDecoration	Muda a “decoração” do texto

# Propriedades Comuns

- Layout

Propriedade	O que altera
cursor	Muda o cursor apresentado
display	Muda a maneira de apresentação do elemento
overflow	Comportamento para conteúdos maiores que o elemento
visibility	Muda a visibilidade do elemento
width	Muda a largura do elemento
height	Muda a altura do elemento

# Propriedades Comuns

- Regra do Nome das Propriedades CSS

```
p: { background-color: xxx; }
```



```
p.style.backgroundColor = "xxx";
```

- Regra do Nome das Propriedades XHTML

```
<p title="xyz"></p>
```



```
p.title = "xxx";
```





# FUNÇÕES COMUNS

# Funções Comuns

- Algumas funções de campos de formulário

Função	O que altera
focus	Muda o foco para o elemento
select	Seleciona o texto do campo
add	Adiciona elemento em lista combo
remove	Remove elemento em lista combo

- Funções de formulário

Função	O que altera
reset	Limpa todos os campos
submit	Envia dados do formulário



# **ATRIBUTOS E MÉTODOS DE JANELA**

# Atributos de Janela

- Mudar janela
- Ex: desligar barra de status

```
window.statusbar = false;
```

- Muitos atributos podem ser acessados

# Atributos de Janela (window.)

- Atributos window. ...

Atributo	Se Refere A
location	Endereço da janela (URL atual)
name	Nome da janela
parent	Janela “pai”
personalbar	Barra de links pessoais
scrollbars	Barras de rolagem
status	Mensagem da barra de status
statusbar	Barra de status
toolbar	Barra de ferramentas

# Métodos de Janela (window.)

- Métodos window. ...
- `window.alert("Olá!");`

Atributo	Se Refere A
<code>alert("texto")</code>	Abre janela de mensagem
<code>r=confirm("texto")</code>	Abre uma janela "Ok/Cancel"
<code>r=prompt("texto")</code>	Abre uma janela de entrada
<code>blur()</code>	Tira o foco da janela atual
<code>close()</code>	Fecha a janela atual
<code>open("url")</code>	Abre uma nova janela do navegador
<code>print()</code>	Abre a janela de impressão
<code>createPopup("url")</code>	Abre uma janela do tipo popup

# Métodos de Janela (window.)

- Métodos window. ...
- Mover a janela

```
window.moveTo(100,100);
```

Atributo	Se Refere A
moveTo(x,y)	Move a janela de maneira absoluta
resizeTo()	Redimensiona a janela de maneira absoluta
moveBy(dx,dy)	Move a janela de maneira relativa
resizeBy()	Redimensiona a janela de maneira relativa



# **ATRIBUTOS DA TELA**



# Atributos da Tela (screen.)

- Atributos screen. ...

Atributo	Se Refere A
height	Altura total da tela
availHeight	Altura da tela (menos barra de tarefas)
width	Largura total da tela
availWidth	Largura da tela (menos barra de tarefas)



**ALTERANDO ESTILOS?**

# Nós Mudamos os Estilos?

- Podemos mudar o estilo de um elemento:

```
document.body.style.backgroundColor = "black";
```

- Isso muda de fato o documento de estilo?
- Na verdade... **NÃO!**
- Isso muda o estilo de forma "inline"!
- O mesmo efeito disso no XHTML:

```
<body style="background-color: black">
```

# Nós Mudamos os Estilos?

- Podemos mudar o estilo de um elemento:

```
document.body.style.backgroundColor = "black";
```

- Isso muda o efeito do elemento? Qual o estilo?
- Na verdade, não muda o efeito do elemento. Qual a diferença?
- Isso muda o efeito do elemento? Qual o estilo?
- O mesmo efeito é alcançado no HTML:

```
<body style="background-color: black">
```

# Nós Mudamos os Estilos?

- A diferença é que se fizermos isso:

```
var cor;
```

```
cor = document.body.style.backgroundColor;
```

```
window.alert("Cor: " + cor);
```

- Nós iremos ler o valor **inline** da cor, não o do arquivo CSS externo!
- Depois veremos como ler o valor do CSS
- **JS não altera o CSS, apenas o XHTML**



**ATIVIDADE**

# Atividade

3. Modifique o código da atividade 2 para que, ao clicar novamente no botão, o fundo volte a ser branco com texto preto
4. Modifique a função de inicialização de maneira que a janela fique com 400x300 e esteja centralizada na tela (se a configuração do navegador permitir)



# **FUNÇÕES COM PARÂMETROS**



# Parâmetros em Funções

- As funções JS pode receber parâmetros
- Basta indicar na declaração

```
function mudaCor(novaCor) {  
    document.body.style.color = novaCor;  
}
```

# Parâmetros em Funções

- As funções JS pode receber parâmetros
- Basta indicar na declaração

```
function mudaCor(novaCor) {  
    document.body.style.color = novaCor;  
}
```

- Chamando...

```
mudaCor("green");
```

- Qual a nova cor do texto?



**ATIVIDADE**

# Atividade

Vamos modificar o jogo da velha:

5. Crie uma função para mostrar a janela, chamada **mostraMensagem**, que receba 2 parâmetros: **título** e **mensagem**.

6. Quando clicarmos no botão “Iniciar Jogo!” uma janela seja mostrada indicando se o jogador escolheu X ou O

7. Crie uma função fictícia **enviarRequest** que receba um comando. O comando `start=x` ou `start=o` deve ser enviado ao iniciar o jogo.



# CONCLUSÕES

# Resumo

- JavaScript: linguagem simples
  - permite modificar elementos de uma página XHTML
- Forma “elegante” de usar JavaScript:
  - Arquivo externo
- Funções do JavaScript:
  - Devem ser associadas a eventos dos elementos HTML
- JavaScript: permite “buscar” um elemento
  - Obter referência para um elemento HTML específico
- **TAREFA**
  - Trabalho A!

# Próxima Aula



- JavaScript... é só isso?
  - Sim e não!
  - Na aula que vem veremos alguns truques que podem ser feitos com ele



**PERGUNTAS?**





**BOM DESCANSO  
A TODOS!**