

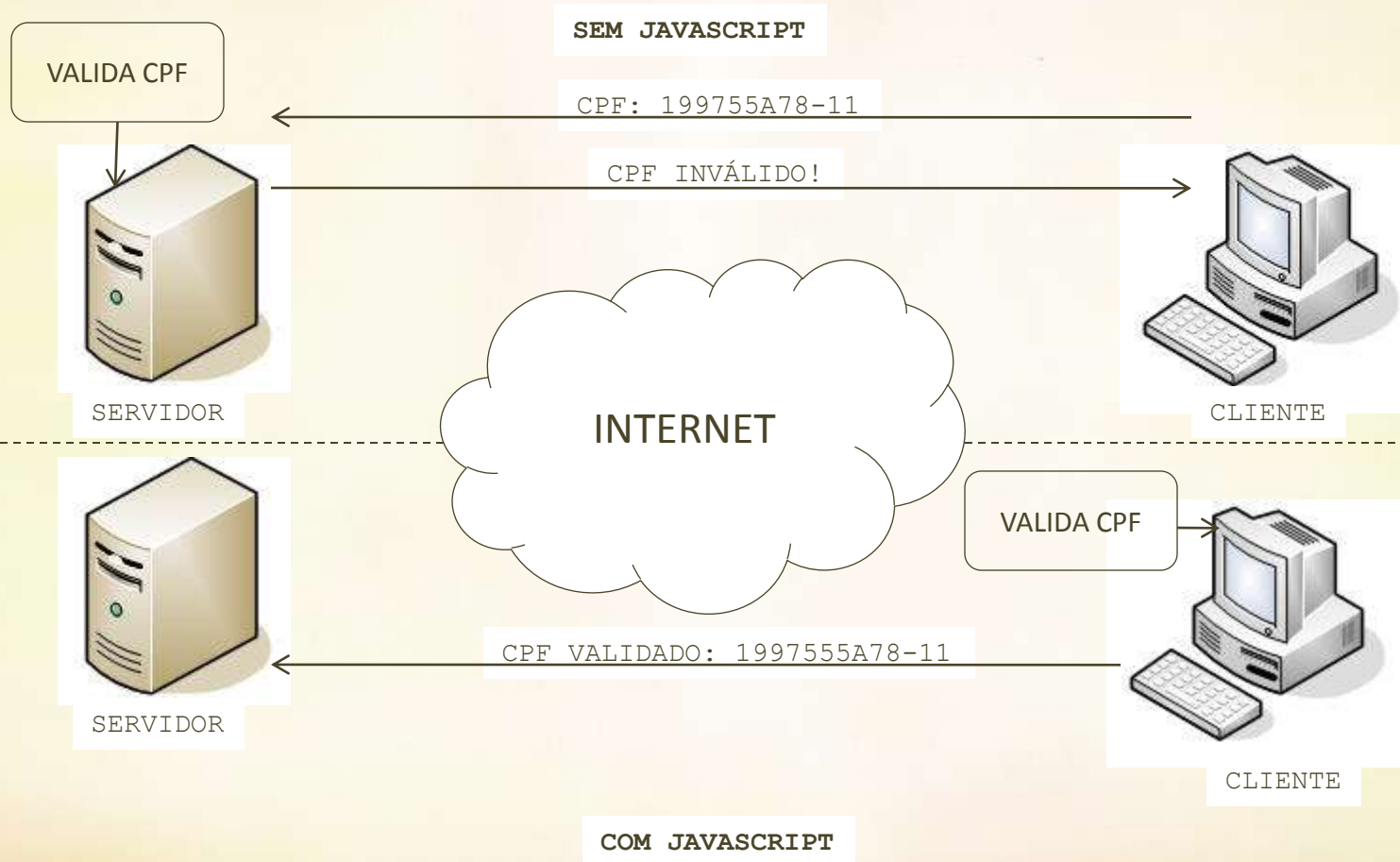
# JAVASCRIPT

DIOGO CEZAR TEIXEIRA BATISTA  
<http://inf.cp.utfpr.edu.br/diogo>  
[diogo@diogocezar.com](mailto:diogo@diogocezar.com)

## JAVASCRIPT

- é uma linguagem de programação interpretada com características de orientação a objetos;
  - desenvolvida pela Netscape a fim de estender as capacidades de seu browser;
  - permite que um conteúdo executável seja incluído em paginas web;
  - sintaticamente semelhante a C e C++;

# JAVASCRIPT



## JAVASCRIPT != JAVA

- Java é uma linguagem orientada a objetos desenvolvida pela *Sun Microsystems* com um escopo bastante amplo (não necessariamente ligado a web);
- o nome *JavaScript* foi escolhido por mera questão de marketing;
- seu nome ia ser *LiveScript*;

## DESTAQUES DA LINGUAGEM

- controlar a aparência e conteúdo do documento:
  - escrever código em um documento enquanto ele esta sendo interpretado pelo browser;
  - alterar cor de fundo da página em tempo de execução;
- controlar o browser:
  - abrir novas janelas, exibir mensagens na barra de status, etc;
- manipular formulários HTML.



## CÓDIGO JAVASCRIPT

- podemos dividir o código JavaScript em três partes:
  - código comum:
    - realiza qualquer tarefa.
  - manipulação de eventos:
    - funções devem ser chamadas para manipular os eventos (onClick, onLoad...).
  - manipulação do *browser*:
    - o browser é tratado como uma hierarquia de objetos.

## ONDE SÃO USADOS?

- blocos `<script>`:
  - usado para definir funções que serão usadas dentro da mesma pagina;
- arquivos externos:
  - definir funções que serão usadas por varias paginas (.js)
- junto a tags HTML:
  - manipulação de eventos.

## BLOCO <SCRIPT>

```
<html>
<head>
<title>JavaScript</title>
</head>
<body>

<script language="JavaScript">
document.write("Hello World!");
</script>

</body>
</html>
```



## ARQUIVOS EXTERNOS

```
<html>
<head>
<title>JavaScript</title>
</head>
<body>

<script language="JavaScript"
src="../js/jquery.js"></script>

</body>
</html>
```

## JUNTO A TAGS HTML

```
<html>
<head>
<title>JavaScript</title>
</head>
<body>

<input type="button"
onclick="javascript:alert('Teste
JavaScript');">

</body>
</html>
```

## SINTAXE E ESTRUTURA

- *case sensitive*:
  - Valor, VALOR e VaLOR são variáveis diferentes;
- na maior parte dos casos, JavaScript ignora espaços em branco;
  - inclusive tabulações e quebras de linha.

## SINTAXE E ESTRUTURA

- ponto e virgula opcional:
  - inserido automaticamente.

```
a = 3;  
b = 4;  
c = 5  
d = 6  
a = 3; b = 4;
```

## SINTAXE E ESTRUTURA

- Comentários:
  - Suporta estilos de C e C++:

```
/* texto entre duas marcas */  
// texto ate o fim da linha
```



## SINTAXE E ESTRUTURA

- declaração de variáveis:
  - não é necessário inicializar a variável na declaração;
  - nesse caso o valor é indefinido (*undefined*).

## TIPOS DE DADOS E LITERAIS

- tipos primitivos
  - string;
  - number;
  - boolean;
  - undefined;
  - null.

## TIPOS DE DADOS E LITERAIS

- os tipos de dados em JavaScript são implícitos:
  - interpretador faz tudo automaticamente;
  - strings → **"Texto de uma string"**;
  - números → 15, 15.6, 18.23 e 10;
  - arrays e tipos complexos.

## VARIÁVEIS

## • declaração implícita:

- atribui-se um valor a um identificador cujo nome não seja uma palavra reservada:

```
salario = 1000;  
nome = "Joao";
```

variáveis globais

## • declaração explícita:

- usa-se a palavra reservada var para definir que o identificador a seguir é uma variável:

```
var salario;  
var nome;
```

variáveis locais

## CONVERSÃO DE TIPOS

- em JavaScript, os tipos das variáveis são implícitos;
  - o tipo da variável depende do valor assumido naquele instante:

```
y = 10  
y = "nome"
```

- a conversão de tipos é automática e transparente ao usuário;
- não existe *casting*;



## NULL E UNDEFINED

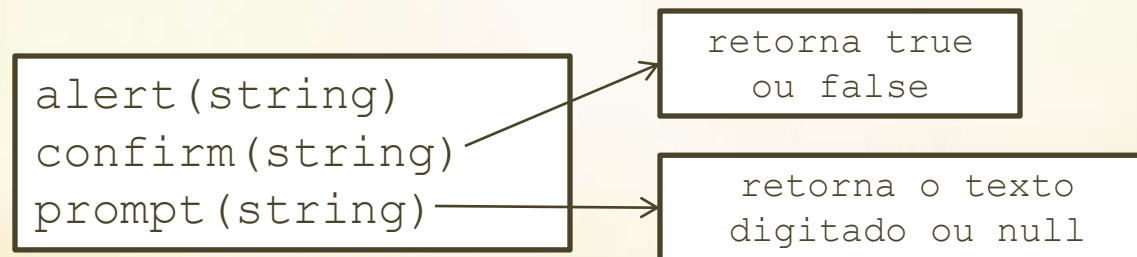
- JavaScript define dois valores especiais relacionados:
  - null → Nenhum valor;
  - undefined → Valor não definido (padrão na declaração).

## CAIXAS DE MENSAGENS

- JavaScript permite a comunicação com o usuário através de caixas de mensagem abertas pelo navegador;
- caixas de mensagem:
  - simples de implementar;
  - poucas opções de personalização;
  - interrupção do usuário.

## CAIXAS DE MENSAGENS

- a função *alert* exibe uma caixa de mensagem simples na tela, com uma frase e um botão OK;
- a função *confirm* exibe uma caixa de mensagens com dois botões: OK e Cancelar;
- a função *prompt* exibe uma caixa de mensagem que permite ao usuário digitar um texto.



## FUNÇÕES

- uma função (ou subrotina, procedimento...) é um pedaço de código reutilizável;
- a função é definida apenas uma vez, e pode ser utilizada um número indefinido de vezes;
- em JavaScript, não é necessário fazer declaração de funções;
- uma função é definida com o uso da palavra *function*.

## FUNÇÕES

- Definição:

```
function exibeMensagem() {  
    document.write("Olá!");  
}
```

- Uso:

```
exibeMensagem();
```



## EVENTOS

- a utilização de JavaScript quase sempre vem atrelada a um eventos de um de nossos componentes no XHTML;
- são ações iniciadas por parte dos usuários;
- podemos chamar determinada função ao clicar em um botão, ou ao entrar em determinado input text.

## EVENTOS

- sua utilização se dá como atributos da linguagem XHTML, ou seja dentro das próprias **Tag's** XHTML;
- sua sintaxe tem a seguinte formação:
  - `<TAG nomeEvento="Instruções JavaScript">`

## EVENTOS

EVENTO	MANIPULADOR	DESCRIÇÃO
blur	onBlur	Ocorre quando o usuário retira o foco de um objeto de formulário.
change	onChange	Ocorre quando o usuário muda o valor de um objeto de formulário.
click	onClick	Ocorre quando o usuário clica sobre o objeto.
focus	onFocus	Ocorre quando o usuário focaliza o objeto.
load	onLoad	Ocorre quando o usuário carrega a página.
unload	onUnload	Ocorre quando o usuário abandona a página.
mouseover	onMouseOver	Ocorre quando o ponteiro do mouse passa sobre um link ou âncora. <b>Válidos apenas para hiperlinks.</b>
select	onSelect	Ocorre quando o usuário seleciona um elemento de um formulário.

## EVENTOS

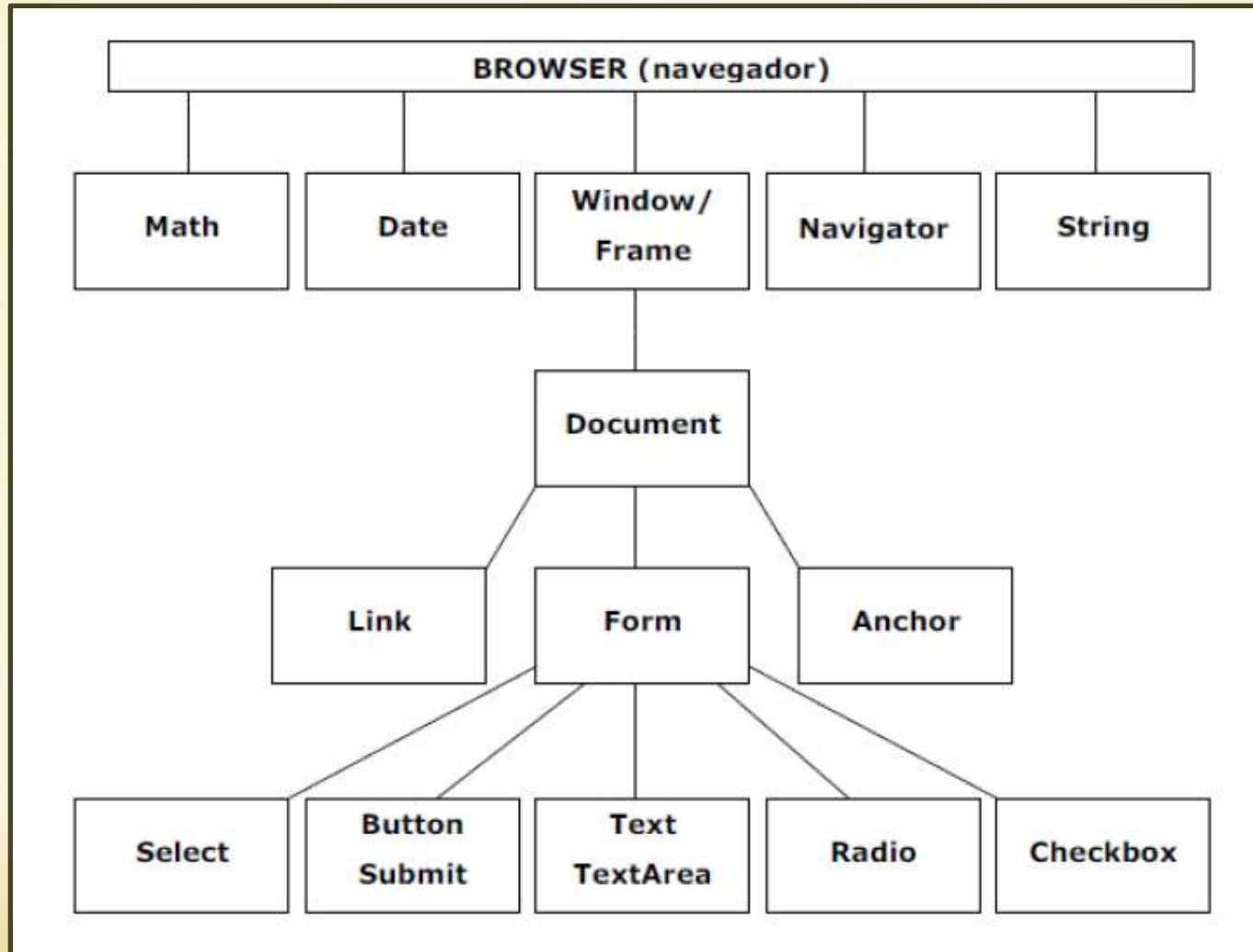
EVENTO	MANIPULADOR	DESCRIÇÃO
submit	onSubmit	Ocorre quando o usuário envia um formulário.
mouseDown	onMouseDown	Ocorre quando o botão do mouse é pressionado.
mouseMove	onMouseMove	Ocorre quando o ponteiro do mouse se movimenta sobre o objeto.
mouseOut	onMouseOut	Ocorre quando o ponteiro do mouse afasta de um objeto. <b>Válidos apenas para hiperlinks.</b>
mouseUp	onMouseUp	Ocorre quando o botão do mouse é solto.
keyDown	onKeyDown	Ocorre quando uma tecla é segurada.
keyPress	onKeyPress	Ocorre quando uma tecla é pressionada.
keyUp	onKeyUp	Ocorre quando uma tecla é solta.

## HIERARQUIA DE OBJETOS

- window: O objeto mais acima na hierarquia, contém propriedades que se aplicam a toda a janela. Há também um objeto desta classe para todas as "sub-janelas" de um documento com frames;
  - location: Contém as propriedades da URL atual;
  - history: Contém as propriedades das URLs visitadas anteriormente
  - document: Contém as propriedades do documento contido na janela, tais como o seu conteúdo, título, cores, etc.



## HIERARQUIA DE OBJETOS



## PROPRIEDADES E MÉTODOS

- sintaxe de utilização de propriedades:
  - nomeObjeto.propriedade;
- sintaxe de utilização dos métodos:
  - nomeObjeto.método(argumento).

## COMANDOS INTERESSANTES

- capturando um elemento por seu id ou nome:

```
<script>  
var nome = document.getElementById("nome");  
var nome =  
document.getElementsByName("nome");  
</script>
```

- a partir desse momento a variável nome assume as propriedades do elemento com id nome;
- se for um input, terá o atributo value;
- se for um div terá o atributo style.

## CAPTURANDO UM TEXTO DE UM INPUT

```
<html>
<head>
<title>Get Text</title>
<script>
function mostra_texto() {
    var texto = document.getElementById('texto');
    alert('Você digitou: ' + texto.value);
}
</script>
</head>
<body>
Digite seu nome:
<input id="texto" name="texto" type="text" maxlength="50"
/>
<input name="botao" type="button" id="botao"
onclick="mostra_texto();" value="Ok"/>
</body>
</html>
```

## ALTERANDO UM ESTILO

- caso o elemento seja selecionado é possível alterar seu estilo, para isso utilizamos o seguinte comando:

```
var texto = document.getElementById('mudaCor');  
texto.style.color = 'red';
```

## ALTERANDO A COR DE UM DIV

```
<html>
<head>
<title>Muda Cor</title>
<script>
function mudar_cor() {
    var texto =
document.getElementById('mudaCor');
    texto.style.color = 'red';
}
</script>
</head>
<body>
<div id="mudaCor">Testando um novo texto</div>
<input name="botao" type="button" id="botao"
onclick="mudar_cor();" value="Ok"/>
</body>
</html>
```



## ANEXANDO HTML

- é possível alterar o conteúdo de uma div com o seguinte comando:

```
var texto = document.getElementById('mudaDiv');  
texto.innerHTML = 'O texto foi trocado!'
```

## ALTERANDO O CONTEÚDO DE UMA DIV

```
<html>
<head>
<title>Muda Div</title>
<script>
function mudar_div() {
    var texto =
document.getElementById('mudaDiv');
    texto.innerHTML = 'O texto foi trocado!'
}
</script>
</head>
<body>
<div id="mudaDiv">Testando um novo texto</div>
<input name="botao" type="button" id="botao"
onclick="mudar_div();" value="Ok"/>
</body>
</html>
```

## DETALHES DE CONTEÚDO

```
<html>
<head>
<title>Detalhes</title>
<script>
function toggle(obj) {
    var el = document.getElementById(obj);
    if ( el.style.display != "none" ){
        el.style.display = 'none';
    }
    else{
        el.style.display = ''
    }
}
</script>
```

## DETALHES DE CONTEÚDO

```
<p>Clica no link para abrir a caixa de informação  
escondida.</p>
```

```
<p><a href="#" onclick="toggle('maisinfo');">Mais  
Informação</a></p>
```

```
<div id="maisinfo" style="display:none">  
<p> Texto que ficará invisível até clicar no link.</p>  
</div>
```

## REDIRECIONANDO

```
<html>
<head>
<title>Redirecionando</title>
<script>
function redirect() {
    window.location = 'http://www.globo.com';
}
</script>
</head>
<body>
<p><a href="#"
onclick="redirect();">Redirecionar</a></p>
</body>
</html>
```

## VOLTANDO

```
<html>
<head>
<title>Redirecionando</title>
<script>
function voltar() {
    window.history.go(-1);
}
</script>
</head>
<body>
<p><a href="#" onclick="voltar();">Voltar</a></p>
</body>
</html>
```



## ATIVIDADE

- criar um script contador:
  - input text que mostrará a contagem atual;
  - botão [+] para incrementar o valor;
  - botão [-] para decrementar o valor;
  - o valor deve ter o valor máximo de 15 e mínimo de -3;
  - exiba uma mensagem caso um valor extrapole os limites estabelecidos.

## ATIVIDADE

- página personalizada:
  - crie uma página em XHTML que tenha um conteúdo em uma única div;
  - faça os seguintes botões:
    - [Verde];
    - [Vermelho];
    - [Azul];
    - [Aumentar];
    - [Diminuir].
  - ao clicar em um dos botões de cor o texto mudará sua cor;
  - ao clicar em aumentar ou diminuir o texto mudará seu tamanho.