



WEB

*JavaScript*

Profa. Joyce Miranda

# DHTML

---

- ▶ DHTML é uma combinação de tecnologias que permite adicionar às páginas web dinamismo e interatividade.
- ▶ DHTML de Cliente
  - ▶ Acessam recursos do navegador.
    - ▶ Ex: **JavaScript**, Flash
- ▶ DHTML de Servidor
  - ▶ Acessam recursos do servidor.
    - ▶ Ex: **PHP**, ASP, JAVA

# JAVASCRIPT

---

- ▶ Foi criada em 1995 pela empresa Netscape, inicialmente com o nome de **LiveScript**
- ▶ **Objetivo inicial**
  - ▶ Minimizar a comunicação entre o cliente e o servidor.
    - ▶ **Motivação:** Lentidão da internet (28.8 kbps)

# JAVASCRIPT

---

**HTML**

**Marcação do  
Conteúdo**

**CSS**

**Formatação e  
Apresentação**

**JavaScript**

**Comportamento  
e Inteligência**

# JAVASCRIPT

---

- ▶ Características
  - ▶ Linguagem interpretada
  - ▶ Independente de plataforma
  - ▶ Tipagem dinâmica
  - ▶ Linguagem baseada em objetos
  - ▶ Programação dirigida por eventos

# JAVASCRIPT

---

- ▶ Criação de Código
  - ▶ Programas Necessários
    - ▶ Editor de texto
    - ▶ Navegador
  - ▶ Os códigos JavaScript podem ser incluídos na página:
    - ▶ Bloco de código: **<SCRIPT> </SCRIPT>**
    - ▶ Arquivo externo (.js)

# JAVASCRIPT

---

- ▶ DESENVOLVENDO SCRIPTS COM O TAG <SCRIPT>

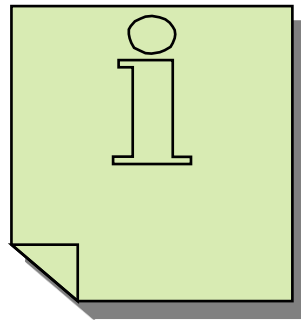
```
<html>
<head>
</head>
<script>
    //imprimindo mensagem na tela
    alert("Seja bem-vindo!!");
</script>
<body>
<h1>Minha Página</h1>
</body>
</html>
```

# JAVASCRIPT

---

- ▶ DESENVOLVENDO SCRIPTS ATRAVÉS DE UM ARQUIVO EXTERNO
  - ▶ Extensão do arquivo *.js*
  - ▶ Insira o seguinte parâmetro na página HTML:

*funcao.js*



```
<script src="funcao.js"></script>
```

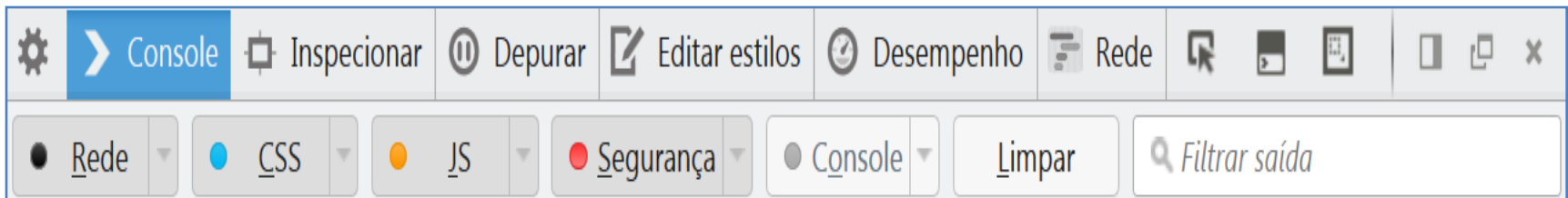


# JAVASCRIPT

---

## ► Console do Navegador

- Alguns navegadores dão suporte à entrada de comandos pelo console.
- **Chrome: Control + Shift + C**
- **Firefox: Control + Shift + K**



```
» alert('Interagindo com o navegador!!!')
```

# JAVASCRIPT

---

## ► Sintaxe Básica

### ► Operações Matemáticas

```
> 12 + 13
```

```
25
```

```
> 14 * 3
```

```
42
```

```
> 10 - 4
```

```
6
```

```
> 25 / 5
```

```
5
```

```
> 23 % 2
```

```
1
```

# JAVASCRIPT

---

## ▶ Sintaxe Básica

### ▶ Variáveis

```
var nomeVariavel;
```

### ▶ Suporte

- ▶ String
- ▶ Number
- ▶ Boolean

```
var texto = "Uma String deve ser envolvida em aspas simples ou duplas.";  
var numero = 2012;  
var verdade = true;
```

# JAVASCRIPT

---

## ► Sintaxe Básica

### ► Funções

```
function nomeFuncao{  
    /* codigo */  
}
```

```
function nomeFuncao(arg1, arg2) {  
    /* codigo */  
    return valor;  
}
```

# JAVASCRIPT

---

## ► Sintaxe Básica

### ► Funções

<code>parseInt()</code>
<code>parseFloat()</code>
<code>String()</code>
<code>Number()</code>
<code>Boolean()</code>

# JAVASCRIPT

---

## ▶ Funções

### ▶ `isNaN()`

- ▶ Retorna um valor booleano que indica que o valor não é um número (NaN).

```
// Returns false.  
isNaN(100);
```

```
// Returns false.  
isNaN("100");
```

```
// Returns true.  
isNaN("ABC");
```

```
// Returns true.  
isNaN("10C");
```

```
// Returns true.  
isNaN("abc123");
```

# JAVASCRIPT

---

## ▶ Objetos Internos

- ▶ Date
- ▶ Math
- ▶ String
- ▶ Array

# JAVASCRIPT

---

## ▶ Objeto Date

- ▶ Através deste objeto o usuário poderá manipular datas e horas.
- ▶ Sintaxe
  - ▶ `var data = new Date()`



# JAVASCRIPT

---

## ► Objeto Date

Método	Descrição
<b>getDate/setDate</b>	Dia do mês.
<b>getDay/setDay</b>	Dia da semana (0=Domingo – 6 =Sábado)
<b>getHours/setHours</b>	Horas (0 a 23).
<b>getMinutes/setMinutes</b>	Minutos (0 a 59).
<b>getMonth/setMonth</b>	Mês do ano (0=janeiro – 11=dezembro).
<b>getSeconds/setSeconds</b>	Segundos (0 a 59).
<b>getFullYear/setFullYear</b>	Ano contendo quatro dígitos.
<b>toLocaleString</b>	Converte a data do objeto <i>Date</i> para uma string nas configurações locais.

# JAVASCRIPT

---

## ► Objeto Date

### ► Exemplos

```
data = new Date();  
alert(data.toLocaleString());  
  
data = new Date(1986, 04, 14);  
alert(data.getDay());
```

## ► Objeto Math

Método	Descrição
<b>PI</b>	Valor de PI (3.14159)
<b>max/min</b>	Retorna o maior/menor de dois números
<b>pow</b>	Retorna a base elevada ao expoente. Calcula o valor de um número elevado a outro.
<b>sqrt</b>	Retorna a raiz quadrada de um número.
<b>random</b>	Retorna um número aleatório entre 0 e 1
<b>round</b>	Retorna um número arredondado para o inteiro mais próximo.

# JAVASCRIPT

---

## ► Objeto Math

### ► Exemplos

```
num = Math.random() * 100;  
alert(Math.round(num));
```

# JAVASCRIPT

## ► Objeto String

Método	Descrição	Exemplo
<b>length</b>	Verifica a quantidade de caracteres presentes na String.	var nome = "Joyce" <u><b>nome.length</b></u> <b>Resultado = 5</b>
<b>charAt</b>	Busca o caracter em determinada posição da String.	var nome = "Joyce" <u><b>nome.charAt(2)</b></u> <b>Resultado = "y"</b>
<b>concat</b>	Concatena strings, retornando uma string apenas.	var s1= "Bom dia, " var s2= "Joyce" <u><b>s1.concat(s2)</b></u> <b>Resultado = "Bom dia, Joyce"</b>
<b>indexOf</b>	Indica a posição de um caracter na string.	var nome = "Joyce" <u><b>nome.indexOf("c")</b></u> <b>Resultado = 3</b> <b>Caso não encontre : -1</b>

# JAVASCRIPT

## ► Objeto String

Método	Descrição	Exemplo
<b>replace</b>	Substitui uma sequência de caracteres por outra.	var nome = "Joyce" <u><b>nome.replace("e", "inha")</b></u> <b>Resultado = "Joycinha"</b>
<b>substring</b>	Retorna um conjunto de caracteres entre a posição inicial e a posição final.	var nome = "Joyce" <u><b>nome.substring(1,3)</b></u> <b>Resultado = "oy"</b>
<b>toLowerCase</b>	Muda as letras do texto para minúsculas.	var nome = "AAAAAA" <u><b>nome.toLowerCase()</b></u> <b>Resultado = "aaaaaa"</b>
<b>toUpperCase</b>	Muda as letras do texto para maiúsculas.	var nome = "aaaaaa" <u><b>nome.toUpperCase()</b></u> <b>Resultado = "AAAAAA"</b>

# JAVASCRIPT

---

## ► Objeto String

### ► Exemplos

```
frase = "Eu, ?, prometo que vou estudar para a prova.";
alert(frase.replace("?", "Joyce"));
```

# JAVASCRIPT

---

## ► Objeto Array

```
var pessoas = ["João", "José", "Maria", "Sebastião", "Antônio"];
```

```
    alert(pessoas[0]);  
    alert(pessoas[1]);  
    alert(pessoas[4]);
```

```
for (var i = 0; i < pessoas.length; i++) {  
    alert(pessoas[i]);  
}
```



# JAVASCRIPT

## ► Objeto Array

Método	Descrição	Exemplo
<b>split</b>	Divide uma string em diversas partes.	<pre>var s = "asp, php, java" <u>linguagem = s.split(",")</u> <b>Resultado</b> linguagem[0] = "asp" linguagem[1] = "php" linguagem[2] = "html"</pre>
<b>join</b>	Contrário do split. Junta em uma string os dados presentes em uma matriz.	<pre>local[0] = "RJ" local[1] = "SP" local[2] = "AM" <u>s = local.join(",")</u> <b>Resultado: "AM, RJ, SP"</b></pre>
<b>sort</b>	Retorna uma versão array ordenada.	<pre><u>so = local.sort()</u> <b>Resultado</b> so[0] = "AM" so[1] = "RJ" so[2] = "SP"</pre>

# JAVASCRIPT

---

## ► Objeto Array

### ► Exemplos

```
relacao = "Ana, Sara, Paulo, Felipe"  
aluno = relacao.split(",");  
for(i = 0; i < aluno.length; i++) {  
    document.writeln(aluno[i]);  
}
```

# JAVASCRIPT

---

## ► Objeto Array

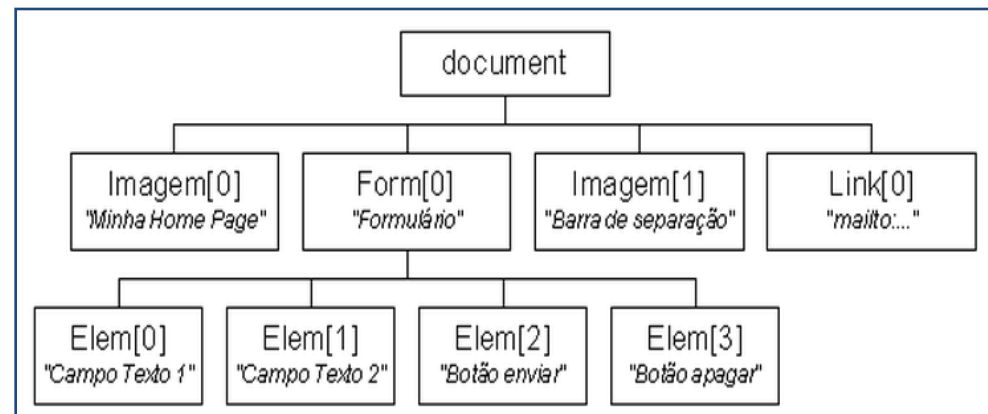
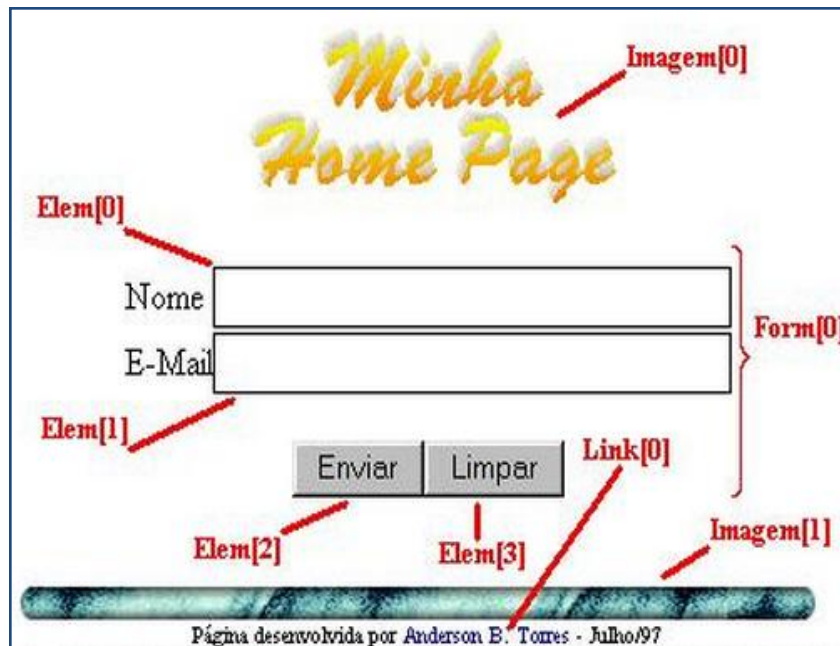
```
var s = ["a", "b", "c"]
```

Método	Descrição	Exemplo
<b>reverse</b>	Inverte a ordem dos elementos.	<u>var x= s.reverse();</u>
<b>slice</b>	Cria uma cópia de uma porção de um array	<u>var x= s.slice(0,2);</u>
<b>push</b>	Adiciona um elemento no final.	<u>s.push("d");</u>
<b>delete</b>	Remove um elemento.	<u>delete s[0];</u>
<b>pop</b>	Remove e retorna o último elemento.	<u>var x= s.pop();</u>
<b>shift</b>	Remove e retorna o primeiro elemento.	<u>var x= s.shift();</u>
<b>unshift</b>	Adiciona elemento na primeira posição	<u>var x= s.unshift();</u>

# JAVASCRIPT

## ► DOM – Document Object Model

- JavaScript organiza todos os elementos de uma página dentro de uma hierarquia.
- Cada elemento é visto como um objeto.



# JAVASCRIPT

---

## ► DOM – Document Object Model

- Os objetos podem ter propriedades, métodos e responder a certos eventos.

Objeto	Descrição
<b>document</b>	Contém as propriedades do documento contido na janela, tais como o seu conteúdo, título, cores.
<b>window</b>	Contém propriedades que se aplicam a toda a janela.
<b>location</b>	Contém as propriedades da URL atual.
<b>history</b>	Contém as propriedades das URLs visitadas anteriormente.

# JAVASCRIPT

## Alguns Métodos de Objetos

<b>alert</b>	Mostra uma caixa de alerta, seguido de um sinal sonoro e o botão de OK. <b>Ex: <code>window.alert('Esta é uma janela de alerta!')</code></b>
<b>confirm</b>	Mostra uma caixa de diálogo, seguida de um sinal sonoro e os botões de OK e Cancel. Retorna um valor verdadeiro se o usuário escolher OK. <b>retorno=<code>window.confirm('Deseja prosseguir?')</code></b>
<b>open</b>	Abre uma nova sessão do browser <b>Ex: <code>window.open("URL")</code></b>
<b>back</b>	Retorna à página anterior. Equivale a clicar no botão <i>back</i> do browser. <b>Ex: <code>history.back()</code></b>
<b>forward</b>	Passa para a próxima página. Equivale a clicar o botão <i>forward</i> do browser. <b>Ex: <code>history.forward()</code></b>
<b>go</b>	Avança ou retorna páginas de acordo com parâmetro. <b>Ex: <code>history.go(2); history.go(-2)</code></b>

# JAVASCRIPT

## ▶ DOM – Document Object Model

### ▶ Recuperando Elementos

#### ▶ Hierarquia do documento

```
<html>
<head>
  <meta charset="utf-8">
  <title>Minha página</title>
</head>
<body>
  <form name="formulario">
    <input type="text" name="campo" >
    <input type="button" name="botao">
  </form>
</body>
</html>
```

```
document.title
document.formulario
document.formulario.campo
document.formulario.campo.value
```

# JAVASCRIPT

---

## ▶ DOM – Document Object Model

### ▶ Recuperando Elementos

#### ▶ **getElementById**

- ❑ Retorna um elemento específico

```
var elemento = document.getElementById("conteudo");
```

#### ▶ **getElementsByName**

- ❑ Recupera um array com os elementos recuperados

```
var array = document.getElementsByName("categoria");
```

#### ▶ **getElementsByTagName**

- ❑ Recupera um array com os elementos recuperados

```
var array = document.getElementsByTagName("p");
```

#### ▶ **getElementsByClassName**

- ❑ Recupera um array com os elementos recuperados

```
var array = document.getElementsByClassName("confirmado");
```



# JAVASCRIPT

---

## ▶ DOM – Document Object Model

### ▶ Recuperando Elementos

#### ▶ Seletores CSS

##### ❑ **querySelector**

- ❑ Retorna o primeiro elemento compatível com o seletor

##### ❑ **querySelectorAll**

- ❑ Retorna um array de elementos compatíveis com o seletor

```
var elemento = document.querySelector("div.aprovado");  
var array = document.querySelectorAll("div.aprovado");
```

# JAVASCRIPT

---

## ▶ DOM – Document Object Model

### ▶ Recuperando Atributos

#### ▶ **getAttribute**

- Retorna o atributo de um elemento específico

```
var valor = elemento.getAttribute("type");
```

```
<figure>  
    
</figure>
```

```
elemento = document.getElementById("logo");  
valorAlt = elemento.getAttribute("alt");  
alert(valorAlt);
```

# JAVASCRIPT

---

## ▶ DOM – Document Object Model

### ▶ Recuperando Conteúdo

#### ▶ **innerHTML**

- Retorna o conteúdo presente entre as tags de abertura e de encerramento de um elemento HTML.

```
var conteudo = elemento.innerHTML;
```

```
<h1 class="principal">Meu Título</h1>
```

```
elemento = document.querySelector("h1");  
alert(elemento.innerHTML);
```

# JAVASCRIPT

---

## ▶ DOM – Document Object Model

### ▶ Recuperando Conteúdo

#### ▶ **value**

- Retorna o valor de um elemento HTML.

\*Nome:

```
<form action="processa.php" method="get" >
  <label for="text_id" > *Nome: </label> </td>
  <input type="text" id="txtNome" name="txtNome" size="58" required >
  <input id="botao_id" type="submit" value="Enviar">
</form>
```

```
alert(document.getElementById("txtNome").value);
```

# JAVASCRIPT

---

## ▶ DOM – Document Object Model

### ▶ Alterando Elementos

#### ▶ **setAttribute**

- Modifica atributos de um elemento HTML

```
elemento.setAttribute("type", "text");
```

#### ▶ **style.propriedade**

- Modifica atributos de um elemento HTML

```
elemento.style.color = "red";
```

#### ▶ **innerHTML**

- Modifica o conteúdo de um elemento HTML

```
> document.querySelector("h1").innerHTML = "Título Modificado"
```

# JAVASCRIPT

---

## ▶ DOM – Document Object Model

### ▶ Removendo Elementos

#### ▶ **remove()**

```
> document.querySelector("h1").remove()
```

### ▶ Criando e Adicionando Elementos

#### ▶ **createElement()**

#### ▶ **appendChild()**

```
var subTitulo = document.createElement("h2");  
subTitulo.style.color = "pink";  
subTitulo.innerHTML = "Meu subTítulo";  
document.body.appendChild(subTitulo);
```

# JAVASCRIPT

---

- ▶ DOM – Document Object Model
  - ▶ Criando e Adicionando Elementos
    - ▶ **createElement()**
    - ▶ **insertBefore()**

```
var subTitulo = document.createElement("h2");  
subTitulo.style.color = "pink";  
subTitulo.innerHTML = "Meu subTítulo";  
var formulario = document.querySelector("form");  
document.body.insertBefore(subTitulo, formulario);
```

# JAVASCRIPT

---

## ▶ Eventos

- ▶ Quaisquer ações iniciadas por parte do usuário.
- ▶ Eventos reconhecidos pelos navegadores
  - ▶ <http://www.w3.org/TR/DOM-Level-3-Events>
  - ▶ [http://en.wikipedia.org/wiki/DOM\\_events](http://en.wikipedia.org/wiki/DOM_events)



# JAVASCRIPT

---

## ▶ Eventos

- ▶ As rotinas de tratamento de eventos são colocadas dentro de TAGs HTML.
- ▶ Sintaxe básica:

`<tag atributos rotina="código javascript">`

```
<body onload="alert('Seja Bem-Vindo ao nosso site!!')">
```

```
<h1 class="principal" onclick="alert(this.innerHTML)">Meu Título</h1>
```

ROTINA	EXEMPLO DE APLICAÇÃO	DESCRIÇÃO
<b><i>onload</i></b>	<b>body, img</b>	É disparado ao término do carregamento de um recurso.
<b><i>onbeforeunload</i></b>	<b>body</b>	É disparado antes de fechar uma página.
<b><i>onresize</i></b>	<b>body, img, form, textarea</b>	É disparado quando ocorre uma alteração no tamanho de determinados elementos.
<b><i>onscroll</i></b>	<b>body, section, div, textarea</b>	É disparado quando a barra de rolagem de um determinado elemento HTML é movida.
<b><i>onfocus</i></b>	<b>input, select e textarea</b>	É disparado quando um determinado elemento ganha o foco.
<b><i>onchange</i></b>	<b>input, select e textarea</b>	É disparado quando um determinado elemento HTML tem o seu conteúdo alterado.
<b><i>onblur</i></b>	<b>input, select e textarea</b>	É disparado quando um determinado elemento perde o foco.
<b><i>onselect</i></b>	<b>input, textarea</b>	É disparado quando o texto contido em um <b>input</b> ou <b>textarea</b> é selecionado.

# JAVASCRIPT

---

ROTINA	EXEMPLO DE APLICAÇÃO	DESCRIÇÃO
<b><i>onsubmit</i></b>	<b>body</b> <b>input type submit</b>	É disparado imediatamente antes do envio de um formulário.
<b><i>onclick</i></b>	<b>p, div, table, img,</b> <b>input type button</b>	É disparado quando um determinado elemento recebe um clique.
<b><i>ondblclick</i></b>	<b>p, div, table, img,</b> <b>input type button</b>	É disparado quando um determinado elemento recebe um duplo clique.
<b><i>onmouseover</i></b>	<b>p, div, table, img</b>	É disparado quando o ponteiro do mouse passa a estar sobre um determinado elemento .
<b><i>onkeydown</i></b>	<b>input e textarea</b>	É disparado no momento em que a tecla é pressionada.

# JAVASCRIPT

---

## ▶ Exemplos

### ▶ onsubmit

Nome:	<input type="text"/>
Login:	<input type="text"/>
Senha:	<input type="password"/>
<input type="submit" value="Cadastrar"/>	

```
<form method="post" action="cadastra.html" novalidate onsubmit="return validar()">
<table>
<tr> <td>Nome:</td>
  <td><input type="text" id="txtNome" name="txtNome" size="50" required></td> </tr>
<tr><td>Login:</td>
  <td><input type="text" id="txtLogin" name="txtLogin" size="50" required></td> </tr>
<tr> <td>Senha:</td>
  <td><input type="password" id="txtSenha" name="txtSenha" size="50" required></td> </tr>
<tr> <td colspan="2"> <input type="submit" value="Cadastrar" > </td> </tr>
</table>
</form>
```

# JAVASCRIPT

- ▶ Exemplos
  - ▶ onsubmit

```
<script type="text/javascript">  
    function validar(){  
        var nome = document.getElementById("txtNome").value;  
        var login = document.getElementById("txtLogin").value;  
        var senha = document.getElementById("txtSenha").value;  
        if(nome == ""){  
            alert("Informe o nome.");  
            return false;  
        }else if(login == ""){  
            alert("Informe o login.");  
            return false;  
        }else if(senha == ""){  
            alert("Informe a senha.");  
            return false;  
        }else{  
            return true;  
        }  
    }  
</script>
```

# JAVASCRIPT

---

## ▶ Exemplos

### ▶ onsubmit

```
<form method="post" action="cadastra.html" onsubmit="return validarCheck()" name="form">  
  <table border=1>  
    <tr> <td>Interesses:</td>  
      <td>  
        <input type="checkbox" name="ckInteresse" value="E"> Exatas  
        <input type="checkbox" name="ckInteresse" value="H"> Humanas  
        <input type="checkbox" name="ckInteresse" value="B"> Biológicas  
      </td> </tr>  
    <tr>  
      <td colspan="2" align=center>  
        <input type="submit" name="btnCadastrar" value="Validar" >  
      </td>  
    </tr>  
  </table>  
</form>
```

Interesses:	<input type="checkbox"/> Exatas	<input type="checkbox"/> Humanas	<input type="checkbox"/> Biológicas
<input type="button" value="Validar"/>			

# JAVASCRIPT

- ▶ Exemplos
  - ▶ onsubmit

```
<script type="text/javascript">  
    function validarCheck0{  
        var checado = "";  
        var campoCheck = document.getElementsByName("ckInteresse");  
        for (i=0; i< campoCheck.length; i++) {  
            if ( campoCheck[i].checked) {  
                checado = true ;  
                break;  
            }  
        }  
        if (checado != true) {  
            alert("Escolha uma opção!!");  
            return false;  
        }  
    }  
</script>
```

# JAVASCRIPT

## ▶ Exemplos

### ▶ onclick

Escolha sua cor de preferência:

```
79 <form method="post" action="cadastra.html" name="form">
80 <table border=1>
81   <tr>
82     <td>Cor:</td>
83     <td>
84       <select name="cbCor" onchange="mudaCor(this)">
85         <option value="white">Padrão</option>
86         <option value="blue">Azul</option>
87         <option value="red">Vermelho</option>
88         <option value="pink">Rosa</option>
89       </select>
90     </td>
91   </tr>
92 </table>
93 </form>
```

```
function mudaCor(element){
    document.body.style.backgroundColor=element.value;
}
```



# JAVASCRIPT

---

Somando Valores	
Valor 1	<input type="text"/>
Valor 2	<input type="text"/>
Resultado	<input type="text"/>
<input type="button" value="Somar"/>	

```
function somar() {  
  
    var valor1 = parseFloat(document.getElementById('valor1').value);  
    var valor2 = parseFloat(document.getElementById('valor2').value);  
    var soma = valor1 + valor2;  
    document.getElementById('resultado').value = soma;  
  
}
```

# JAVASCRIPT

---

## ▶ Associando Funções a Eventos

### ▶ Outras formas

#### ▶ **addEventListener**(*evento*, *funcao*)

- É possível associar várias funções para tratar um evento

```
function click() {  
    console.log("click");  
}  
  
var elemento = document.getElementById("conteudo");  
elemento.addEventListener("click", click);
```

# JAVASCRIPT

---

## ▶ Associando Funções a Eventos

### ▶ Outras formas

#### ▶ Funções anônimas

```
var elemento = document.getElementById("conteudo");  
elemento.addEventListener(  
    "click",  
    function() {  
        console.log("click")  
    }  
);
```

# JAVASCRIPT

---

## ▶ Associando Funções a Eventos

### ▶ Outras formas

#### ▶ Propriedades dos objetos DOM

```
var elemento = document.getElementById("conteudo");  
elemento.onclick = function() {  
    console.log("click")  
};
```

### ▶ Removendo

#### ▶ Não pode ser utilizado por funções anônimas

```
elemento.removeEventListener("click", click);
```