

# Desenvolvimento Web JavaScript

*Prof. Leonardo Cruz.*

leonardocruz@id.uff.br

Departamento de Computação, UFF

# Funções

- Funções são blocos de código reutilizáveis.
- Elas não são executadas até que sejam **chamadas**
- Podem ter parâmetros de entrada e de saída
- Podemos ter vários parâmetros de entrada separados por vírgulas
- Podemos retornar um valor através da instrução **return**

# Funções

## ■ Sintaxe:

```
function nomeDaFuncao() {  
    //códigos referentes à função.  
    ...  
}  
function nomeDaFuncao(p1, p2, p3, ...) {  
    //códigos referentes à função.  
    ...  
}  
function nomeDaFuncao(p1, p2, p3, ...) {  
    ...  
    return p1+p2-p3;  
    ...  
}
```

# Funções

```
...  
<a href = "#" onclick = "alo();" >Chamar a função</a>  
...
```

```
function alo() {  
    alert("Link clicado!");  
}
```

**Veja: ex5.html**

# Funções

```
<form>
  <input type = "text" name="teste"/>
  <input type = "button" value = "Chamar função"
onclick = "alo();" />
</form>
...
```

```
function alo() {
    alert("Link clicado!");
}
```

**Veja ex6.html**

# Funções – passando parâmetros

## Passagem de valor

```
...  
<form>  
  <input type = "button" value = "Chamar função"  
    onclick = "saudacao('jose');"/>  
</form>  
...
```

```
function saudacao(nome) {  
    alert("Olá, " + nome);  
}
```

Veja ex7.html

# Funções - Passando como parâmetros campos de formulário

```
...  
<form>  
  <input type="text" name="txtNome" id = "txtNome"/>  
  <input type="button" name="btn_saudacao"  
    value = "Chamar funcao"  
    onclick = "saudacao(document.getElementById('txtNome').value);"/>  
</form>  
...
```

```
function saudacao(nome) {  
    alert("Olá, " + nome);  
}
```

# Funções - Passando como parâmetros campos de formulário

```
...  
<form name = "frm" id="frm">  
    <input type="text" name="txtNome"/>  
    <input type="button" name="btn_saudacao"  
        value = "Chamar funcao"  
        onclick = "saudacao(frm.txtNome.value);"/>  
</form>  
...
```



# Eventos

- São reações a ações do usuário ou da própria página  
**ou**
- São ocorrências ou acontecimentos dentro de uma página.
- Ex:
  - ☐ Carregar uma página;
  - ☐ Clicar em um botão;
  - ☐ Modificar o texto de uma caixa de texto;
  - ☐ Sair de um campo texto;
  - ☐ etc;

# Eventos

- **onclick**: ocorre quando o usuário clica sobre algum elemento da página

```
...  
<a href = "#" onclick = "alo();" >Chamar a função</a>  
...
```

# Eventos

- **onload** e **onunload**: ocorrem respectivamente quando o objeto que as possuem são carregados (criados) e descarregados

```
...  
<body onload = " saudacao();" >  
...
```

Veja ex10.html

# Eventos

- **onmouseover:** é acionado quando o mouse se localiza na área de um elemento
- **onmouseout:** ocorre quando o mouse sai da área de um elemento

# Eventos

```
...  
<a href = "#" onmouseover = "mouseSobre();"   
      onmouseout = "mouseFora();" >
```

Passe o mouse

```
</a>
```

```
<div id = "resultado"> </div>
```

```
...
```



# Eventos

```
function mouseSobre() {  
    var divResultado = document.getElementById("resultado");  
    divResultado.innerHTML = divResultado.innerHTML +  
        "mouse sobre.<br/>";  
}
```

```
function mouseFora() {  
    var divResultado = document.getElementById("resultado");  
    divResultado.innerHTML = divResultado.innerHTML +  
        "mouse fora.<br/>";  
}
```

A propriedade **innerHTML** retorna ou define o conteúdo de um elemento HTML.

# Eventos

- **onsubmit**: usado para chamar a validação de um formulário (ao enviar os dados)
- Para validar um formulário, chamamos uma função por nós definida:
  - Ao chamar a função, usamos a palavra reservada return
- A função, por sua vez, deve retornar true ou false, representando se os dados devem ou não serem enviados.

# Eventos

```
<form name="frm"  
  action="...."  
  method="post" onsubmit = "return validaCampo()">  
  Nome: <input type="text" name="nome" id = "nome" />  
  <input type="submit" name="btnBuscar"  
  value="Buscar"/>  
</form>
```



# Eventos

```
function validaCampo() {  
    var valor = document.getElementById("nome").value;  
  
    if ((valor == null) || (valor == "")) {  
        alert("Preencha o campo de busca");  
        return false;  
    }  
    return true;  
}
```

**Veja ex12.html**

# Outros Eventos

- **onfocus**: ocorre quando um controle recebe o foco através do mouse ou do teclado.
- **onblur**: ocorre quando um controle perde o foco

# Eventos

- **onkeydown** e **onkeypress**: são semelhantes e ocorrem quando uma tecla é pressionada pelo usuário em seu teclado.
- **onkeyup**: é executado quando a tecla é liberada, ou seja, ela foi pressionada e em seguida liberada.

# Validações de formulários

# Validações de formulários

## ■ Algumas dicas:

- Ao se validar um campo, procure sempre obtê-los pelo atributo `id`
- Quase todos os elementos do formulário possuem sempre um atributo **value**, que pode ser acessado como uma `String`
- Para verificar um caractere em especial dentro de um valor, use `[ ]`, pois as `Strings` são arrays de caracteres
- As `Strings` também possuem um atributo **length** que assim como os arrays, representam o tamanho

# Validações de formulários

- Alguns exemplos de validação:
  - ☐ Campos de texto não preenchidos
  - ☐ Campo de texto com tamanho mínimo e máximo
  - ☐ Validação de campo de e-mail
  - ☐ Campos com apenas números em seu conteúdo
  - ☐ Seleção obrigatória de radio buttons, checkboxes e caixas de seleção

# Validações de formulários

- Validação de campo de texto com preenchimento obrigatório:
  - Deve-se validar se:
    - O valor é nulo
    - O valor é uma String vazia
    - O valor é formado apenas por espaço
  - A validação feita para um campo do tipo **text** serve também para um **textarea** e para um **password**
  - Validar espaços pode ser feito usando expressões regulares

# Validações de formulários

- Validação de campo de texto com preenchimento obrigatório:

```
function validaCampoTexto(id) {  
    var valor = document.getElementById(id).value;  
    //testa se o valor é nulo, vazio ou formado por apenas  
    //espaços em branco  
    if ( (valor == null) || (valor == "") || (/^\s+$/).test(valor)) {  
        return false;  
    }  
    return true;  
}
```

Veja ex13.html



# Validações de formulários

- Validação de tamanho em campos de texto:
  - É importante validar primeiramente se o campo tem algo preenchido (validação anterior)
  - Pode-se limitar o campo a um tamanho mínimo ou máximo
  - Usa-se o atributo **length** para se checar o tamanho do campo valor do componente do formulário

# Validações de formulários

- Validação de tamanho em campos de texto:

```
function validaCampoTextoTamanho(id, minimo, maximo) {  
    var valor = document.getElementById(id).value;  
    if (!validaCampoTexto(id)) {  
        return false;  
    }  
  
    if ( (valor.length < minimo) || (valor.length > maximo)) {  
        return false;  
    }  
    return true;  
}
```

Veja ex14.html

# Validações de formulários

- Validar para que um campo tenha apenas números:

```
function validaCampoNumerico(id) {  
    var valor = parseInt(document.getElementById(id).value);  
  
    if (isNaN(valor) ) {  
        return false;  
    }  
    return true;  
}
```

Veja ex15.html

# Validações de formulários

- Validar se uma caixa de checagem (checkbox) está marcada:
  - Deve-se consultar o atributo **checked** do componente

```
function validaCampoCheckbox() {  
    var check = document.getElementsByName("sexo");  
    for (var i=0; i<check.length; i++){  
        if (check[i].checked == true){  
            alert(check[i].value);  
            return true;  
        }  
    }  
    alert("Faca a escolha");  
    return false;  
}
```

**Veja ex16.html**

# Validações de formulários

- Validar se pelo menos um botão de radio de um conjunto foi selecionado:

```
function validaCamposRadio(nome) {
    var opcoes = document.getElementsByName(nome);
    var selecionado = false;
    for(var i = 0; i < opcoes.length; i++) {
        if(opcoes[i].checked) {
            selecionado = true;
            break;
        }
    }
    if(!selecionado) {
        return false;
    }
    return true;
}
```

Veja ex17.html

# Validações de formulários

- Exemplo completo

**Veja `ex18.html`**