

SERVLETS E JSP (JEE) - Javascript

Diogo Cezar Teixeira Batista

`diogo@diogocezar.com.br`

`http://www.diogocezar.com`

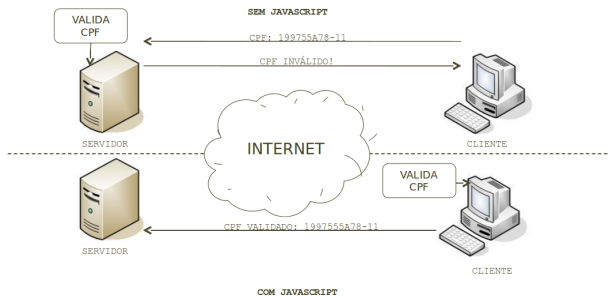
Universidade Tecnológica Federal do Paraná - UTFPR

Cornélio Procópio - 2012

Javascript - O que é? I

- 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 páginas *web*;
 - sintaticamente semelhante a C e C++;

Javascript - O que é? II



- **Java** é uma linguagem orientada a objetos 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**;

- 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;

- 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;
- 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.

Javascript - 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 páginas (.js);
- junto a tags HTML:
 - manipulação de eventos;

Código 1: Bloco <script>

```
<html>
<head>
  <title>JavaScript</title>
</head>
<body>
  <script language="JavaScript">
    document.write("Hello World!");
  </script>
</body>
</html>
```

Código 2: Arquivos Externos

```
<html>
<head>
  <title>JavaScript</title>
  <script type="text/javascript" src="adjustments.js"></script>
</head>
<body>
</body>
</html>
```

Código 3: Junto a uma Tag

```
<html>
<head>
  <title>JavaScript</title>
</head>
<body>
  <input type="button" onclick="javascript:alert('TesteJavaScript');">
</body>
</html>
```

- 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;
- ponto e vírgula opcional (recomendável utilização);
- comentários: estilo C ou C++;

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

- tipos primitivos:

- string;
- number;
- boolean;
- undefined;
- null;

- os tipos de dados em JavaScript são implícitos;
- interpretador faz tudo automaticamente;
- **declaração implícita**: atribui-se um valor a um identificador cujo nome não seja uma palavra reservada:

Código 4: Declaração Implícita

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

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

Código 5: Declaração Explícita

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

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

Código 6: Declaração Explícita

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

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

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

- 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;

- 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;

Código 7: Caixas de Diálogo

```
alert(string)
confirm(string) // retorna true ou false
prompt(string) // retorna o texto digitado
```

- 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.

Código 8: Declaração de uma função

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

Código 9: Uso de uma função

```
exibeMensagem();
```

- 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*, por exemplo;
- sua utilização se dá como atributos da linguagem XHTML, ou seja dentro das próprias Tag's XHTML;
- sua sintaxe tem a seguinte formação:

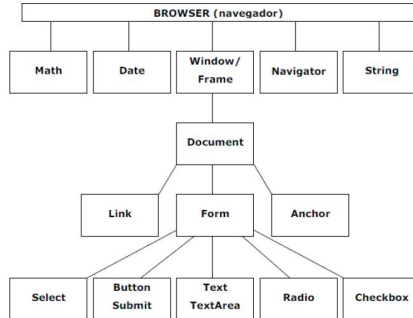
```
<tag nomeEvento=" Instruções JavaScript">
```

Evento	Manipulador	Descrição
submit	onSubmit	Quando o usuário envia um formulário
mouseDown	onMouseDown	Quando o botão do mouse é pressionado
mouseMove	onMouseMove	Quando o ponteiro do mouse se movimenta sobre o objeto
mouseOut	onMouseOut	Quando o ponteiro do mouse se afasta de um objeto (só hiperlinks)
mouseUp	onMouseUp	Quando o botão do mouse é solto
keyDown	onKeyDown	Quando uma tecla é segurada
keyPress	onKeyPress	Quando uma tecla é pressionada
keyUp	onKeyUp	Quando uma tecla é solta

Evento	Manipulador	Descrição
blur	onBlur	Quando o usuário retira o foco de um objeto de formulário
change	onChange	Quando o usuário muda o valor de de um objeto de formulário
click	onClick	Quando o usuário clica sobre o objeto
focus	onFocus	Quando o usuário focaliza o objeto
load	onLoad	Quando o usuário carrega a página
unload	onUnload	Quando o usuário abandona a página
mouseover	onMouseOver	Quando o ponteiro do mouse passa sobre um link ou âncora (só hiperlinks)
select	onSelect	Quando o usuário seleciona um elemento de um formulário

- **window**: o objeto 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.

Javascript - Hierarquia de Objetos II



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

- 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.

```
<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>
```

- 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";
```

```
<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>
```

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

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

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

Código 10: 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>
</head>
<body>
    <p>Clica no link</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>
</body>
</html>
```

Código 11: 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>
```

Código 12: Voltando

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

- 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.