Desenvolvimento de Aplicações WEB

JavaScript

Profa. Joyce Miranda

JAVASCRIPT

- **Objetivo inicial**
  - Minimizar a comunicação entre o cliente e o servidor.
    - **Motivação**: Lentidão da internet (28.8 kbps)

JAVASCRIPT

- Características
  - Linguagem interpretada
    - Scripts são embutidos nas páginas HTML e interpretados pelo navegador
  - Tipagem dinâmica
    - Tipos de variáveis não são definidos, seu conteúdo é que as define.
  - Linguagem baseada em objetos
    - Os elementos de uma página são tratados como objetos que podem ter propriedades, métodos e responder a eventos.
  - Programação dirigida por eventos
    - A página deixa de ser um documento estático e passa a interagir com as ações do usuário.

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

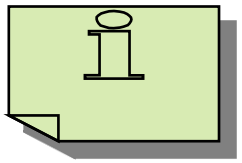
- DESENVOLVENDO SCRIPTS COM A TAG <SCRIPT>

```
1  <html>
2      <head>...</head>
3      <script type="text/javascript">
4          alert("Seja bem-vindo(a)!");
5      </script>
6      <body>
7          ...
8      </body>
9  </html>
```

JAVASCRIPT

- DESENVOLVENDO SCRIPTS ATRAVÉS DE UM ARQUIVO EXTERNO
  - Crie um arquivo com a extensão ***.js*** e inclua o código JavaScript nele.
  - Inclua a referência do arquivo na página HTML.

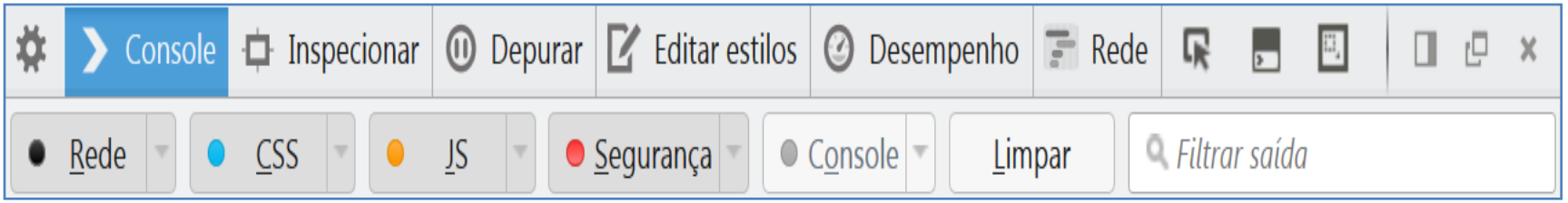
funcoes.js



```
1  <html>
2      <head>...</head>
3      <script type="text/javascript" src="funcoes.js"/>
4      <body>
5          ...
6      </body>
7  </html>
```

JAVASCRIPT

- Console do Navegador
  - Alguns navegadores dão suporte à entrada de comandos pelo console.
    - **Botão direito: Inspecionar Elemento**




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

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

- Objeto Date
  - Sintaxe

```
data = new Date();
alert(data.toLocaleString());

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

JAVASCRIPT

- Objeto Date

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

JAVASCRIPT

- Objeto String

Método	Exemplo
length	var nome = “Joyce” **nome.length** **Resultado = 5**
indexOf	var nome = “Joyce” **nome.indexOf(“c”)** **Resultado = 3** ***Caso não encontre : -1***
substring	var nome = “Joyce” **nome. substring(1,3)** **Resultado = “oy”**
replace	var nome = “Joyce” **nome.replace(“e”, “inha”)** **Resultado = “Joycinha”**

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
split	Divide uma string em diversas partes.	var s = “asp, php, java” **linguagem = s.split(“,“)** **Resultado** **linguagem[0] = “asp”** **linguagem[1] = “php”** **linguagem[2] = “html”**
join	Contrário do split. Junta em uma string os dados presentes em uma matriz.	**local[0] = “RJ”** **local[1] = “SP”** **local[2] = “AM”** **s = local.join(“,“)** **Resultado: “AM, RJ, SP”**
sort	Retorna uma versão array ordenada.	**so = local.sort()** **Resultado** **so[0] = “AM”** **so[1] = “RJ”** **so[2] = “SP”**

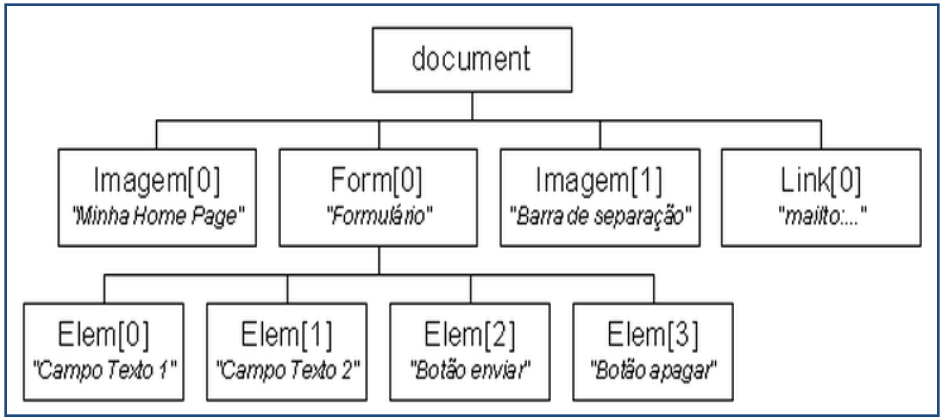
JAVASCRIPT

- Objeto Array
  - Exemplos

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

JAVASCRIPT

- DOM – Document Object Model
  - JavaScript organiza todos os elementos de uma página dentro de uma hierarquia.



JAVASCRIPT

- DOM – Document Object Model
  - Os objetos podem ter propriedades, métodos e responder a certos eventos.

Objeto	Descrição
window	Contém propriedades que se aplicam a toda a janela.
document	Contém as propriedades do documento contido na janela.
location	Contém as propriedades da URL atual.
history	Contém as propriedades das URLs visitadas anteriormente.

JAVASCRIPT

Alguns Métodos de Objetos	
alert	Ex: window.alert('Esta é uma janela de alerta!‘)
confirm	Ex: retorno=window.confirm('Deseja prosseguir?')
open	Ex: window.open("URL")
back	Ex: history.back()
forward	Ex: history.forward()

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**

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

    - **getElementsByName**

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

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 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.


```
<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

- Eventos
  - Quaisquer ações iniciadas por parte do usuário.
  - Eventos recnhecidos pelos navegadores
    - http://www.w3.org/TR/DOM-Level-3-Events
    - http://en.wikipedia.org/wiki/DOM_events

JAVASCRIPT

- Tratamento de Eventos
  - Sintaxe básica:

<tag atributos rotina="código javascript">

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

JAVASCRIPT

ROTINAS
onsubmit
onclick
ondblclick
onmouseover
onkeydown
onChange

JAVASCRIPT

- Exemplos
  - onsubmit


```
<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>
```

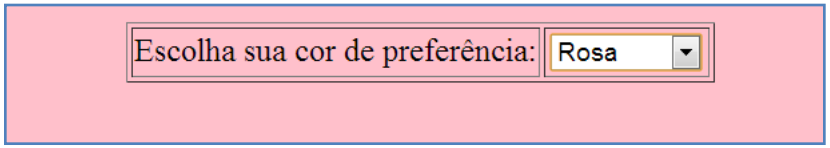
JAVASCRIPT

- 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

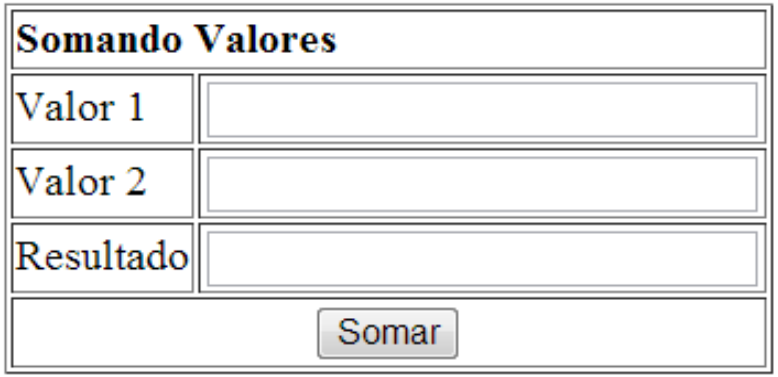
- Exemplos
  - onclick




```
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




```
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

- Web Storage
  - É possível armazenar dados localmente no navegador do usuário.
  - Objetos para armazenar dados no cliente
    - *window.localStorage*
      - Armazena dados sem expiração de data
    - *window.sessionStorage*
      - *Armazena dados para uma sessão do usuário (os dados são perdidos quando o usuário fecha o navegador*

Fonte: http://www.w3schools.com/html/html5_webstorage.asp

JAVASCRIPT

- Web Storage
  - Verificar compatibilidade do navegador

API					
Web Storage	4.0	8.0	3.5	4.0	11.5

```
if (typeof(Storage) !== "undefined") {
    // Code for localStorage/sessionStorage.
} else {
    // Sorry! No Web Storage support..
}
```

JAVASCRIPT

- *window.localStorage*
  - Criando/Recuperando

```
// Store
localStorage.lastname = "Smith";
// Retrieve
document.getElementById("result").innerHTML =
localStorage.lastname;
```

  - Removendo

```
localStorage.removeItem("lastname");
```

```
function clickCounter() {
    if(typeof(Storage) !== "undefined") {
        if (localStorage.clickcount) {
            localStorage.clickcount = Number(localStorage.clickcount)+1;
        } else {
            localStorage.clickcount = 1;
        }
        document.getElementById("result").innerHTML = "You have clicked the
button " + localStorage.clickcount + " time(s).";
    } else {
        document.getElementById("result").innerHTML = "Sorry, your browser
does not support web storage...";
    }
}
```

```
<p><button onclick="clickCounter()" type="button">Click me!</button></p>
<div id="result"></div>
```

JAVASCRIPT

- *window.sessionStorage*

```
if (sessionStorage.clickcount) {
    sessionStorage.clickcount =
Number(sessionStorage.clickcount) + 1;
} else {
    sessionStorage.clickcount = 1;
}
document.getElementById("result").innerHTML = "You have clicked
the button " +
sessionStorage.clickcount + " time(s) in this session.";
```

Simulando Carrinho de Compras

```
do{
    produto = "";
    if(!sessionStorage.carrinho){
        //criando carrinho (array) de produtos
        carrinho = [];
        //incluindo produto no carrinho (array)
        produto = prompt("Informe o produto. Digite Fim para encerrar.");
        if(produto == "Fim") break;
        carrinho[0] = produto;
        //incluindo carrinho (array) na sessão
        sessionStorage.carrinho =  carrinho;
    }else{
        //-- atualizando carrinho  --//
        //recuperando carrinho sessão
        carrinho = (sessionStorage.carrinho).split(",");
        //recuperando quantidade produtos no carrinho
        qtde = carrinho.length;
        //adicionando novo produto ao carrinho
        produto = prompt("Informe o produto. Digite Fim para encerrar.");
        if(produto == "Fim") break;
        carrinho[qtde] = produto;
        //atualizando carrinho da sessão
        sessionStorage.carrinho =  carrinho;
    }
}while(produto != "Fim");
```

JAVASCRIPT

- *Simulando Carrinho de Compras*

```
//exibindo informações do carrinho
if(sessionStorage.carrinho){
    carrinho = (sessionStorage.carrinho).split(",");
    for(i=0; i < carrinho.length; i++){
        document.write(carrinho[i] + "<br>");
    }
}
```

JAVASCRIPT

- jQuery
  - Biblioteca JavaScript
  - Vantagem
    - Maior compatibilidade de um mesmo código com diversos navegadores
    - Biblioteca padrão na programação *front-end*
    - Sintaxe mais fluida

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

JAVASCRIPT

- jQuery
  - **Sintaxe Básica**
    - **A Função $**

$(*selector*).*action*()

```
$('#form').css('background', 'black');

$('.headline').hide();

$('p').text('alô :D');

$("#txtNome").val()
```

JAVASCRIPT

- jQuery
  - **Document Ready Event**
    - Previne que um código seja executado antes da finalização do carregamento do documento no navegador.

```
$(document).ready(function(){

    // jQuery methods go here...

});
```

JAVASCRIPT

- jQuery
  - **Seletores**

```
$('div > p:first'); // o primeiro elemento <p> imediatamente filho de um <div>

$('input:hidden'); // todos os inputs invisíveis

$('input:selected'); // todas as checkboxes selecionadas

$('input[type=button]'); // todos os inputs com type="button"

$('td, th'); // todas as tds e ths
```

JAVASCRIPT

- jQuery
  - **Eventos**

```
$("p").click();
```

```
$("p").click(function(){
    $(this).hide();
});
```

```
$("p").on("click", function(){
    alert("The paragraph was clicked.");
});
```

JAVASCRIPT

- jQuery
  - **Utilitários**
    - **isNumeric**

```
$.isNumeric($("#txtNome").val());
true
```

JAVASCRIPT

- jQuery
  - **Utilitários**
    - **Iteração**

```
$("input").each(
      function(index, item){
            alert($(item).val());
       });
```

JAVASCRIPT

- jQuery
  - **Utilitário de Iteração**

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

$.each(pessoas, function(index, item) {
  alert(item);
})
```