



Estrutura e Banco de Dados

Conteúdo:

- JavaScript

- *Formas de utilização*
- *Conceitos básicos*
- *Operadores*
- *Caixas de diálogo (entrada e saída)*
- *Comando de impressão*
- *Funções*
- *Eventos*
- *Conversão de tipos*

- Claudiney Sanches Júnior

JavaScript

- Linguagem integrada ao navegador.
- Usa o chamado “modelo de execução controlado por eventos”, ou seja, o código JavaScript só é executado quando o evento ao qual está associado é acionado, mas em alguns casos, podemos inserir o código sem a associação a eventos e o mesmo é executado conforme o navegador interpreta a página.
- Processamento do lado cliente, no navegador.
- É uma linguagem interpretada, diferente de outras linguagens que fazem o processo de compilação antes de executar o programa, como por exemplo a linguagem Java.

Java x JavaScript

- JavaScript

- Interpretada
- Variáveis não declaradas
- Não é orientada a objetos, embora possamos “simular” algo neste sentido
- Utilizada em páginas Web
- Grava apenas arquivos de cookies e consegue ler arquivos no formato XML.

- Java

- Compilada
- Variáveis declaradas
- 100% orientada a objetos
- Pode ser utilizada em:
 - Aplicativos desktop
 - Aplicativos distribuídos (que envolve conexões em rede)
 - Aplicativos mobile
 - Aplicativos web
 - Aplicativos para TV
 - Entre outros...
- Criação e/ou manipulação:
 - Arquivos
 - Banco de dados

O que podemos fazer com JavaScript?

- Podemos adicionar interatividade ao HTML
 - Fazer algumas animações
 - Validar formulários antes do envio
 - Manipular arquivos XML
 - Identificar o navegador e com isso executar determinados scripts
 - Criação de menus com submenus
- Essas são algumas das várias opções que tornam as páginas desenvolvidas em HTML mais interativas, dinâmicas.

Como inserir um código JavaScript na página?

- Podemos inserir nossos códigos JavaScript das seguintes formas:
 - Dentro do corpo da página `<body>....</body>`
 - Dentro do cabeçalho `<head>....</head>`
 - Dentro de um marcador html
 - Em um arquivo separado do html, este arquivo deve ter a extensão .js

JavaScript dentro do body e head

- Se fizermos uma comparação com CSS, esse modo de inserir JavaScript em uma página, seria o modo incorporado.
- Dentro do body ou do head, utilizamos a tag <script>
- Ex:

```
<html>
<head>
  <title>Exemplo JavaScript</title>
  <script language="javascript">
    alert("Eu estou no cabeçalho.");
  </script>
</head>
<body>
  <script language="javascript">
    document.write("Eu estou no corpo do documento.");
  </script>
</body>
</html>
```

JavaScript em uma tag

- Quando inserimos um código JavaScript em uma tag, este é associado a um evento, os eventos serão vistos mais adiante, por enquanto, faremos um exemplo simples com o evento onclick que é acionado quando o usuário clica no elemento.
- Ex:

```
<html>
<head>
<title>Exemplo JavaScript</title>
</head>
<body>
<input type="button" value="Clique aqui"
  onclick="window.status='Teste de javascript...'">
</body>
</html>
```

Arquivo JavaScript

- Assim como em CSS, também podemos criar um arquivo separado do html com nossos códigos em JavaScript, esse arquivo deve ser salvo com a extensão .js e é chamado no cabeçalho da página com a tag <script>

- Ex:

```
<html>
```

```
<head>
```

```
<title>Exemplo JavaScript</title>
```

```
<script language="javascript" src="ex1.js"></script>
```

```
</head>
```

```
<body>
```

Conteúdo da página. Também podemos fazer aqui chamadas para blocos de códigos do arquivo .js.

```
</body>
```

```
</html>
```

```
ex1.js
```

```
alert("Estou em um arquivo .js");
```


Conceitos básicos

- Variáveis

- Toda variável deve começar com uma letra ou um underscore("_");
- Caracteres subsequentes devem ser letras ou números;
- Não deve conter espaço em branco ou caracteres especiais;
- Não deve ser uma palavra reservada.

- JavaScript é Case-Sensitive, logo as variáveis abaixo são todas diferentes:

- quantidade
- QUANTIDADE
- Quantidade
- QUantidade

Tipos de variáveis

- JavaScript é uma linguagem de tipos de variáveis flexíveis, ao contrário de outras linguagens que exigem a declaração de uma variável com um tipo de dado definido.
- Logo, não precisamos definir o tipo de uma variável, com isso podemos mudar seu valor de texto para número, por exemplo, durante a execução do script (essa prática deve ser evitada).
- Ex:
 - **var nome="Leonardo";**
 - ou
 - **nome="Leonardo";**
 - ou
 - **var nome;**

Nos dois primeiros exemplos, declaramos a variável nome já com a informação, no último exemplo, declaramos a variável nome vazia.

Comentários

- É uma boa prática inserirmos comentários em nossos códigos, em JavaScript podemos fazer isso de duas formas:

- Para Comentários de várias linhas:

```
/* Inserimos aqui  
Nossos comentários  
Em várias linhas  
*/
```

- Para Comentários de uma única linha:

```
//Aqui inserimos somente uma linha de comentário
```

Operadores Aritméticos

- Para você verificar a tabela abaixo, declaramos uma variável x igual a 10
- var x=10;

Operador		Exemplo	Resultado
+	Adição	$z = x + 10;$	$z = 20$
-	Subtração	$z = x - 10;$	$z = 0$
*	Multiplicação	$z = x * 10;$	$z = 100$
/	Divisão	$z = x / 10;$	$z = 1$
%	Módulo (resto de uma operação de divisão)	$z = x \% 3;$	$z = 1$
++	Incremento antes	$z = ++x;$	$z = 11$
	Incremento depois	$z = x++;$	$z = 10$
--	Decremento antes	$z = --x;$	$z = 9$
	Decremento depois	$z = x--;$	$z = 10$

Operadores de atribuição

- Para você verificar a tabela abaixo, declaramos duas variáveis:
 - var x=10;
 - var y=2;

Operador	Exemplo	Equivalência	Resultado
=	x = y	-	x = 2
+=	x += y	x = x + y	x = 12
-=	x -= y	x = x - y	x = 8
*=	x *= y	x = x * y	x = 20
/=	x /= y	x = x / y	x = 5
%=	x %= y	x = x % y	x = 0

Concatenação de Strings

- Também usamos o operador “+” para concatenar Strings (textos) em JavaScript
 - Ex:
- var x=“Bom”;**
var y=“dia”;
var z = x + y;
// após a execução o resultado será: z=Bom dia
- Cuidado, quando usamos o operador “+” com números e textos, o resultado é sempre texto;

var x=4;
var y=“a”;
var z=x+y; // o resultado será z=“4a”

var x=“4”;
var y=“4”;
var z=x+y; //o resultado será z=“44”

Operadores de comparação

- Para você verificar a tabela abaixo, declaramos uma variável
- var x=10;

Operador	Descrição	Exemplo
==	Igual a	x == 20 (falso)
===	Exatamente igual a	x === 10 (verdadeiro) x === "10" (falso)
!=	Diferente de	x != 20 (verdadeiro)
<	Menor que	x < 20 (verdadeiro)
<=	Menor ou igual a	x <= 10 (verdadeiro)
>	Maior que	x > 20 (falso)
>=	Maior ou igual a	x >= 10 (verdadeiro)

Operadores lógicos

- Para você verificar a tabela abaixo, declaramos duas variáveis:

- var x=10;
- var y=2;

Operador	Descrição	Exemplo
&&	e	(x > y && y < 5) (verdadeiro)
	ou	(x > y y > 5) (verdadeiro)
!	Negação	! (x != y) (falso)

- Tabela verdade

Expressão 1	Expressão 2	&&	
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

Expressão	!
true	false
false	true

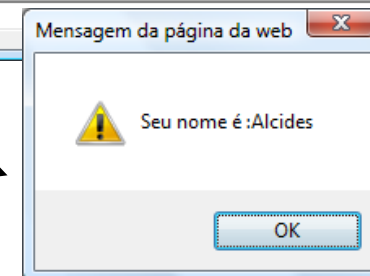
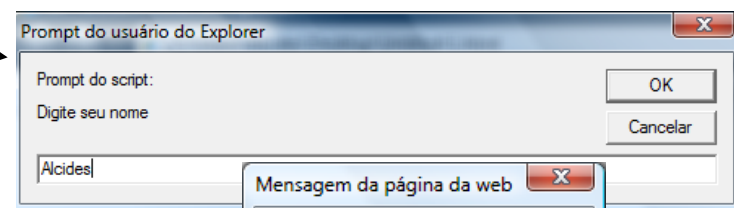
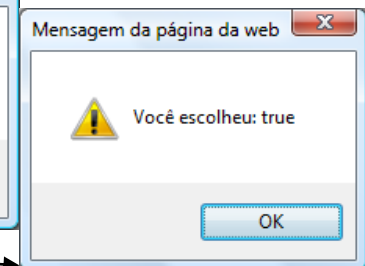
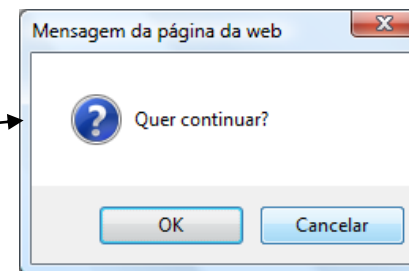
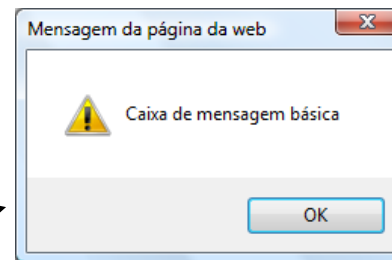
true = verdadeiro
false = falso

Caixa de diálogo

- São meios básicos para entrada e saída de informação
- **Caixa de alerta**
Caixa de diálogo para exibir mensagens, possui somente um botão OK.
Sintaxe:
`alert("Mensagem");`
- **Caixa de confirmação**
Caixa de diálogo básica para tomada de decisão, possui dois botões, o botão OK (retorna true) e o botão CANCELAR (retorna false), os valores retornados devem ser armazenados em uma variável.
Sintaxe:
`variável = confirm("Mensagem");`
- **Caixa de entrada de dados**
Caixa de diálogo para entrada de dados básica, o valor deve ser armazenado em uma variável, o valor digitado sempre será texto, logo, para operações matemáticas precisamos converter os valores digitados para números.
Sintaxe:
`variável = prompt("Mensagem","valor inicial opcional");`

Caixa de diálogo (cont.)

```
<html>
<head>
<title>JS</title>
<script language="javascript">
//Caixa de mensagem
alert("Caixa de mensagem básica");
//Caixa de confirmação
var x;
x = confirm("Quer continuar?");
//Caixa de mensagem
alert("Você escolheu: " + x);
//Caixa de entrada
var nome;
nome = prompt("Digite seu nome");
//Caixa de mensagem
alert("Seu nome é :" + nome);
</script>
</head>
<body>
</body>
</html>
```



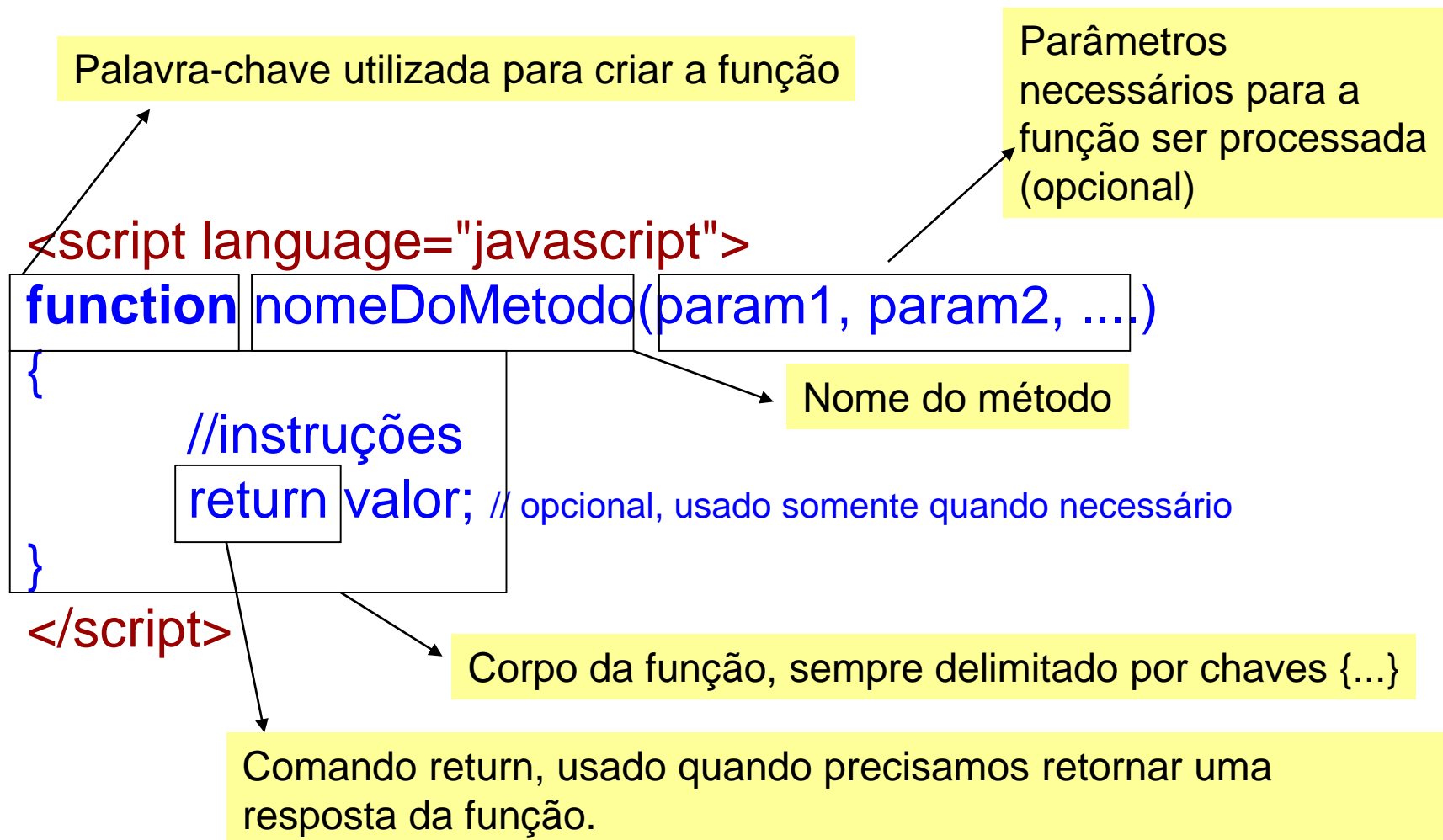
Comando de impressão

- Além da caixa de diálogo “alert”, temos disponível também em JavaScript outro comando de impressão, neste caso, as informações são impressas na página e nos permite imprimir tags html.
- Sintaxe:
`document.write(“mensagem”);`
- Exemplo:
`var nome=“Lukas”;`
`document.write(“Nome:” + nome + “”);`
- OBS: as tags devem sempre estar dentro das aspas.

Funções em JavaScript

- Funções são trechos de códigos que podem ser acionados através de uma chamada ou através de um evento.
- Podemos chamar uma função em qualquer ponto da página e se a mesma estiver em um arquivo .js, podemos chamá-la em qualquer página que faz referência ao arquivo JavaScript.
- As funções geralmente são criadas em um arquivo .js ou no cabeçalho (<head>) do documento html.
- Uma função pode ou não retornar uma resposta para o ponto em que foi chamada, caso necessite de um retorno (resposta), utilizamos o comando **return**.

Funções em JavaScript (cont.)



Exemplos básicos de funções

```
<script language="javascript">
```

```
function soma(a, b)
{
    document.write((a+b)+"<br />");
}
```

Função soma, recebe dois parâmetros (a, b), efetua a soma entre eles e imprime na página, pulando uma linha com o BR.

```
function lerNome()
{
    var nome;
    nome = prompt("Digite seu nome","");
    return nome;
}
```

Função lerNome, não recebe parâmetros, lê um nome através da janela de prompt e depois retorna esse nome (return) para o ponto em que foi chamada.

```
//chama a primeira função
```

```
soma(2,2);
soma(3,4);
```

Chama a função soma duas vezes com valores diferentes para a e b.

```
//chama a segunda função
```

```
var resp;
resp = lerNome();
alert(resp);
</script>
```

Chama a função lerNome, como a função retorna um valor, devemos atribuir este retorno para uma variável, para que posteriormente possamos manipular essa informação.

Eventos

- Os eventos são ações que o usuário faz em uma página e que são associados geralmente as funções do JavaScript ou as funções que nós criamos em nossos códigos.
- Com a utilização dos eventos, podemos criar páginas mais dinâmicas.

Principais eventos

Evento	Quando é acionado
onload	Sempre que a página é carregada ou recarregada.
onunload	Sempre que a página é trocada ou fechada.
onfocus	Quando um elemento receber o foco.
onblur	Sempre que um campo de formulário perder o foco.
onchange	Sempre que um campo tiver seu conteúdo alterado.
onclick	Sempre que um elemento receber um click.
onmouseover	Quando o mouse fica sobre o elemento.
onmouseout	Quando o mouse for retirado de cima do elemento.
onselect	Sempre que um conteúdo de um campo é selecionado.
onsubmit	Quando enviamos (botão submit) um formulário.

Existem outros eventos para mouse e alguns para tratarmos o teclado, neste material, citamos somente os principais, porém você poderá encontrar uma referência completa em livros ou em alguns site da web.

Eventos (exemplo)

```
<html>
<head>
<script language="javascript">
function lerNome()
{
    var nome;
    nome = prompt("Digite seu nome", "");
    alert("Seu nome é: " + nome);
}
</script>
</head>
<body>
<input type="button" value="Ler o nome" onclick="lerNome()" />
</body>
</html>
```

Conversão de tipos

- Como comentado anteriormente, quando utilizamos a caixa de prompt para entrada de dados do usuário, ou até mesmo através de campos de texto de formulários, os valores lidos são sempre textos, caso seja necessário realizar alguma operação matemática com os valores lidos, precisamos inicialmente converte-los em números, para isso temos duas funções em JavaScript que convertem para inteiro ou para float.


Conversão de tipos (cont.)

- Para converter um texto em número inteiro utilizado a função `parseInt`


- Sintaxe:

`variável = parseInt(valor [, base]);`

Valor a ser convertido.



Base da conversão (opcional).




- Para converter um texto em número real (float), utilizamos a função `parseFloat`.

- Sintaxe:

`variável = parseFloat(valor);`

Valor a ser convertido.



Conversão de tipos (exemplo)

```
<script language="javascript">
function soma(a , b)
{
    alert(a + " + " + b + " = " + (a+b));
}
//modo 1
var a;
var aux = prompt("Digite o valor de a","");
a = parseInt(aux);
//modo 2
var b = parseInt(prompt("Digite o valor de b",""));
//chama a função
soma(a,b);
</script>
```

Exercícios

1. Crie um arquivo HTML com o conteúdo abaixo. O nome da pessoa deve ser informado pelo prompt do javascript e inserido na página.

Boa noite, **<nome_da_pessoa>** !

Veja nossas ofertas!



Guarda-sol
R\$ 70,00



Sacola de praia
R\$ 30,00



Paleta de tintas
R\$ 45,00

Exercícios

2. Crie um arquivo HTML que utilizando javascript leia a base e a altura de um retângulo, calcule sua área e perímetro e apresente os resultados no corpo do HTML.
3. Crie um arquivo HTML que utilizando javascript leia hora e minuto, converta tudo para minutos e apresente os resultados no corpo do HTML.
4. Crie um arquivo HTML que utilizando javascript leia dois números. Dentro deste arquivo crie uma função para somar, uma para subtrair, uma para multiplicar, uma para dividir e uma para pegar o resto da divisão. Apresente os resultados do uso de todas as funções no corpo do HTML, passe os dois números lidos como parâmetros para as funções.

Exercícios

5. Monte um HTML conforme exemplo abaixo:



- Crie 3 funções:
 - Uma que seja chamada no evento onload do body. Ela deve solicitar dois números float via prompt;
 - Uma que seja chamada no evento onclick da imagem do sinal de +. Ela deve utilizar os números informados para a fazer soma e exibí-los via Alert;
 - E uma que seja chamada no evento onmouseover da imagem da mão. Ela deve exibir a mensagem "Passei por aqui!" via Alert.