

Introdução à Programação Web



prof. Wesley Freitas Souza

JavaScript

- JavaScript é a linguagem de programação do HTML;
 - Pode modificar o conteúdo do HTML;
 - Pode modificar atributos do HTML;
 - Pode modificar estilos CSS;
 - Pode esconder ou exibir elementos HTML;

JavaScript

- Modificando o conteúdo de elementos HTML:
 - Um dos métodos do JavaScript é o **getElementById()**
 - Encontra um elemento HTML pelo seu **id**
 - **innerHTML** (HTML interno)

```
document.getElementById("demo").innerHTML = "Hello JavaScript";
```

http://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_inner_html

JavaScript

- Modificando conteúdo dos elementos HTML:

```
<!DOCTYPE html>
<html>
<body>

<h1>What Can JavaScript Do?</h1>
|
<p id="demo">JavaScript can change HTML content.</p>

<button type="button" onclick="document.getElementById('demo').innerHTML = 'Hello JavaScript!'">Click Me!</button>

</body>
</html>
```

JavaScript

- Modificando atributos de elementos HTML:
 - O exemplo abaixo procura pelo elemento com **id myImage** e altera o valor de seu atributo **src**;

```
document.getElementById('myImage').src='pic_bulbon.gif'
```

http://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_lightbulb

JavaScript

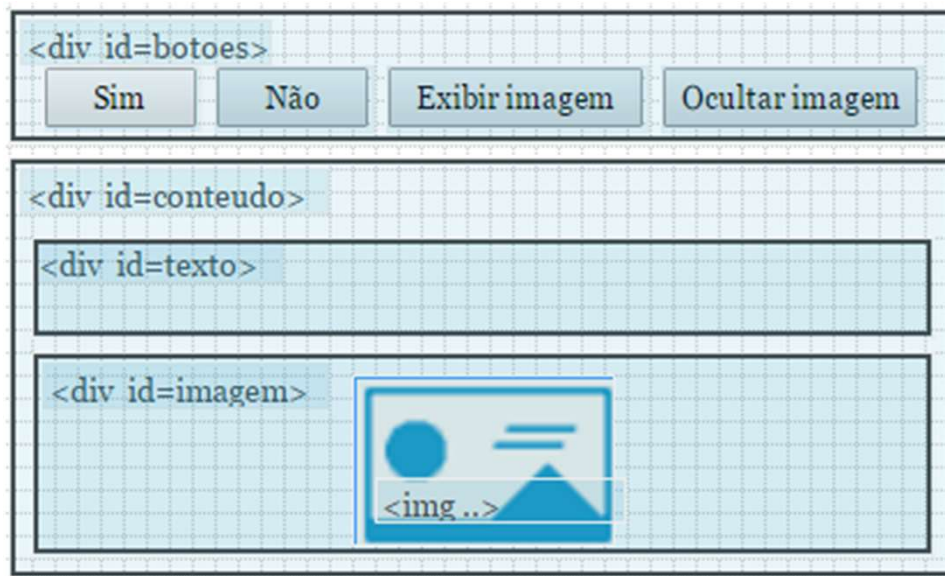
- Modificando o estilo de elementos HTML:
 - O exemplo abaixo procura pelo elemento com **id demo** e altera o valor do estilo de tamanho da fonte;

`document.getElementById('demo').style.fontSize="25px";`

http://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_style

JavaScript - Exercício 1

wesleyifcamboriu@googlegroups.com



1. Criar um documento HTML com a estrutura apresentada:
2. O botão "Sim" deve alterar o conteúdo da **div texto** para "**SIM**";
3. O botão "Não" deve alterar o conteúdo da **div texto** para "**NÃO**";
4. O botão "Exibir imagem" deve alterar o **display** da **div imagem** para "**block**";
5. O botão "Ocultar imagem" deve alterar o **display** da **div imagem** para "**none**";

Javascript - Funções e Eventos

Com o uso de Funções é possível modularizar os programas;

Portanto, as funções permitem melhor organizar o código, e a reutilização de trechos de programa, facilitando a sua leitura e manutenção;

As funções são também chamadas de métodos, procedures ou módulos, dependendo da linguagem e do paradigma de programação.

Javascript

Funções com Passagem de Parâmetros

Já utilizamos funções com passagem de parâmetros:

alert() - Coloca-se entre os parênteses o valor a ser exibido na mensagem - É UM VALOR PASSADO POR PARÂMETRO

Parâmetros ampliam as possibilidades de uso da função, tornando-a mais genérica, funcionando para mais casos;

Com uma mesma função, podemos exibir diferentes mensagens;

Javascript

Funções com Passagem de Parâmetros

```
<script>
function exibirSituacao(nota, media){
    if (nota >= media){ //compara o valor das variaveis
        document.getElementById('resposta').innerHTML = "Aprovado";
    }else{
        alert("Reprovado");
    }
}
var prova1 = Number(prompt("Informe a nota: ")); //lê uma nota
exibirSituacao(prova1, 7); //chama a função, passando 2 parâmetros
</script>
```

Javascript

Funções com Passagem de Parâmetros

A vantagem do uso de funções é que, agora, podemos associar a chamada da função a um EVENTO:

- Clicar em um botão ou texto;
- Digitar uma tecla;
- Passar o mouse sobre um elemento qualquer;
- etc...

```
<button onclick="exibirSituacao(5, 7)">Verificar</button>
```

Javascript

Funções com Valor de Retorno

Até agora, vimos funções que, após resolver o problema, **exibem um resultado**.

Mas e se quisermos fazer uma função com possibilidade de uso mais ampliado, mais genérico;

Será que sempre que eu calcular a situação do aluno, quero exibir o resultado em um alert??

E se em vez de exibir, a função **retornar** o valor para quem a chamou???

Javascript

Funções com Valor de Retorno

Para **retornar um valor** da execução da função, utilizaremos a palavra **return**;

```
<script>
function exibirSituacao(nota, media){
    if (nota >= media){ //compara o valor das variaveis
        resultado = "Aprovado";
    }else{
        resultado = "Reprovado";
    }
    return resultado; //o valor será retornado para quem chamar a função
}
</srint>
```

Javascript

Funções com Valor de Retorno

Portanto, quem chama a função decide o que fazer com o resultado:

```
<script>
function exibirSituacao(nota, media){
    ... //código da função
}
var situacao = exibirSituacao(5, 7); //situacao guarda o resultado da execução

alert(situacao) //exibe, mas se quisesse fazer outra coisa, já tem em variável
</script>
```

JavaScript - A tag <script>

- O JavaScript pode ser colocado no <body> e no <head> do HTML;
 - O código deve ser inserido entre as tags <script> e </script>;
 - **Atenção!** Se o elemento ainda não tiver sido criado, não será encontrado (ver a posição da tag <script>).

```
<script>
```

```
document.getElementById("demo").innerHTML = "My First JavaScript";
```

```
</script>
```

JavaScript - Funções e eventos

- Uma **função** JavaScript é um bloco de código que pode ser executado **quando chamado**;

```
<script>
```

```
document.getElementById("demo").innerHTML = "My First JavaScript";
```

```
</script>
```


JavaScript - A tag <script>

- O JavaScript pode ser colocado no <body> e no <head> do HTML;
 - O código deve ser inserido entre as tags <script> e </script>;

```
<script>
```

```
document.getElementById("demo").innerHTML = "My First JavaScript";
```

```
</script>
```

http://www.w3schools.com/js/js_where.asp

JavaScript - A tag <script>

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph
changed.";
}
</script>
</head>
<body>
<h1>My Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>
</body>
```

<script> dentro do <head>

definição da função JavaScript

chamada da função

JavaScript - A tag <script>

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>My Web Page</h1>
```

```
<p id="demo">A Paragraph</p>
```

```
<button type="button" onclick="myFunction()">Try it</button>
```

<script> dentro do <head>

```
<script>
```

```
function myFunction() {  
    document.getElementById("demo").innerHTML = "Paragraph changed.";
```

definição da função JavaScript

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

chamada da função

JavaScript - A tag <script>

- Boas práticas

- É recomendável manter os códigos em um único local;
- Dependendo do tamanho dos códigos, é uma boa ideia posicioná-los ao final do código HTML, ainda dentro do <body>;
 - Isso pode melhorar o tempo de carregamento da página

JavaScript - Exercício 2

wesleyifcamboriu@googlegroups.com



Com o documento HTML criado anteriormente:

1. Javascript em arquivo separado
2. Crie 4 funções para fazer as mesmas ações dos botões;
3. Cada botão deverá, agora, fazer a chamada à função correspondente;

JavaScript - Funções com parâmetros

- Uma **função** JavaScript pode receber parâmetros como entrada para sua execução:

```
<script>
function mudaValorDemo(texto) {
    document.getElementById("demo").innerHTML = texto;
}
</script>

<button onclick="mudaValorDemo('Sim')">Muda</button>
```

JavaScript - Exercício 3

wesleyifcamboriu@googlegroups.com



Com o documento HTML criado anteriormente:

1. Crie 2 funções para fazer as ações dos botões, recebendo parâmetros;
 - Mudar texto da DIV;
 - Exibir ou Ocultar a div da imagem;
2. Cada botão deverá, agora, fazer a chamada à função correspondente;

JavaScript - Exercício 4

wesleyifcamboriu@googlegroups.com



Com o documento HTML criado anteriormente:

1. Altere as funções para que seja possível:
 - 1 botão mudar texto da DIV para Sim/Não;
 - 1 botão Exibir/Ocultar a div da imagem;