

9.0 Eventos JavaScript

Quando um usuário visita uma página web e interage com ela se produzem os eventos e com

JavaScript podemos definir o que queremos que ocorra quando se produzam.

Com JavaScript podemos definir o que acontece quando se produz um evento, como por exemplo quando um usuário clica sobre um botão, edita um campo de texto ou abandona a página um evento é lançado e podemos definir que alguma função seja executada quando isso acontecer.

9.1 Relação de eventos

Abaixo uma lista de eventos suportados pelo JavaScript com suas respectivas descrições e a partir de que versão do JavaScript esse evento foi incorporado.

- **onabort**

Este evento se produz quando um usuário cancela o carregamento de uma imagem, seja porque para o carregamento da página ou porque realiza uma ação que a detém, como por exemplo, sair da página. JavaScript 1.1

- **onblur**

Desata-se um evento onblur quando um elemento perde o foco da aplicação. O foco da aplicação é o lugar onde está situado o cursor, por exemplo, pode estar situado sobre um campo de texto, uma página, um botão ou qualquer outro elemento. JavaScript 1.0

- **onchange**

Desata-se este evento quando muda o estado de um elemento de formulário, às vezes não se produz até que o usuário retire o foco da aplicação do elemento. JavaScript 1.0

- **onclick**

Produz-se quando se clica o botão do mouse sobre um elemento da página, geralmente um botão ou um link. JavaScript 1.0

- **ondragdrop**

Produz-se quando um usuário solta algo que havia arrastado sobre a página web. JavaScript 1.2

- **onerror**

Produz-se quando não se pode carregar um documento ou uma imagem e esta fica quebrada. JavaScript 1.1

- **onfocus**

O evento onfocus é o contrário de onblur. Produz-se quando um elemento da página ou a janela ganham o

foco da aplicação. JavaScript 1.0

- **onkeydown**

Este evento é produzido no instante que um usuário pressiona uma tecla, independentemente que a solte ou não. É produzido no momento do clique. JavaScript 1.2

- **onkeypress**

Ocorre um evento onkeypress quando o usuário deixa uma tecla clicada por um tempo determinado. Antes deste evento se produz um onkeydown no momento que se clica a tecla. JavaScript 1.2

- **onkeyup**

Produz-se quando o usuário deixa de apertar uma tecla. É produzido no momento que se libera a tecla. JavaScript 1.2

- **onload**

Este evento se desata quando a página, ou em JavaScript 1.1 as imagens, terminaram de se carregar. JavaScript 1.0

- **onmousedown**

Produz-se o evento onmousedown quando o usuário clica sobre um elemento da página. onmousedown se produz no momento de clicar o botão, soltando ou não. JavaScript 1.2

- **onmousemove**

Produz-se quando o mouse se move pela página. JavaScript 1.2

- **onmouseout**

Desata-se um evento onmouseout quando a seta do mouse sai da área ocupada por um elemento da página. JavaScript 1.1

- **onmouseover**

Este evento desata-se quando a seta do mouse entra na área ocupada por um elemento da página. JavaScript 1.0

- **onmouseup**

Este evento se produz no momento que o usuário solta o botão do mouse, que previamente havia clicado. JavaScript 1.2

- **onmove**

Evento que se executa quando se move a janela do navegador, ou um frame. JavaScript 1.2

- **onresize**

Evento que se produz quando se redimensiona a janela do navegador, ou o frame, no caso de que a página

os tenha. JavaScript 1.2

- **onreset**

Este evento está associado aos formulários e se desata no momento que um usuário clica no botão de reset de um formulário. JavaScript 1.1

- **onselect**

Executa-se quando um usuário realiza uma seleção de um elemento de um formulário. JavaScript 1.0

- **onsubmit**

Ocorre quando o visitante aperta sobre o botão de enviar o formulário. Executa-se antes do envio propriamente dito. JavaScript 1.0

- **onunload**

Ao abandonar uma página, seja porque se clique em um link que nos leva a outra página ou porque se fecha a janela do navegador, se executa o evento onunload.

Destes eventos iremos selecionar e estudar os principais nos próximos tópicos

9.2 Evento onclick

O evento onclick é lançado/disparado quando clicamos em um elemento da página, geralmente um botão ou um link. Esse evento é válido para os objetos ou componentes de página HTML: Button, Checkbox, Radio, Link, Reset e Submit.

Sintaxe

```
<elemento onclick="AlgumCodigoJavaScript">
```

Em HTML:

Exemplo: onclick

Quando o usuário clicar no botão “Clique aqui” o evento onclick será disparado e chamara automaticamente a função de nome “minhaFuncao” a qual irá imprimir a seguinte String em um caixa de diálogo alert: “O evento onclick foi disparado”.

```
<!DOCTYPE html>
<html Lang="pt">
  <head>
    <meta charset="utf-8">
    <title>
      Onclick
    </title>

    <script type="text/javascript">
      function minhaFuncao(numero){
        alert("Evento Onclick");
      }
    </script>
  </head>

  <body>
    <p>Clique no botão </p>
    <button onclick="minhaFuncao()">Clique Aqui</button>
  </body>
</html>
```

O mesmo exemplo poderia ser implementado dessa forma:

```
<!DOCTYPE html>
<html Lang="pt">
  <head>
    <meta charset="utf-8">
    <title>
      Onclick
    </title>
  </head>

  <body>
    <p>Clique no botão </p>
    <button onclick="alert('Evento Onclick')">Clique Aqui</button>
  </body>
</html>
```

9.3 Evento onsubmit

O evento onsubmit é suportado apenas por formulários e disparado quando o usuário clica no botão Enviar de seus formulários. É neste evento que você fará a validação do formulário, assim como enviará o usuário para uma página, informando do sucesso/falha do envio de dados, ou outra ação que você queira definir no momento.

Sintaxe em HTML:

```
<form onsubmit="AlgumCodigoJavaScript">
```

Exemplo: onsubmit

Para observarmos o evento onsubmit iremos fazer a validação de um formulário de login que irá conter os campos nome de usuário e senha que serão verificados quando o usuário clicar no botão logar. Essa ação do usuário irá gerar um evento onsubmit o qual irá chamar a função “validacao()”.

Vamos construir o nosso formulário de login com a nossa função validacao().

```

1 <!DOCTYPE html>
2 <html Lang="pt">
3   <head>
4     <meta charset="utf-8">
5     <title>
6       Pagina Principal
7     </title>
8     <script type="text/javascript">
9       function validacao(){
10         var user = "jonas";
11         var password = "qwe123";
12         if ((user == document.form1.usuario.value)
13           &&(password==document.form1.senha.value)){
14           window.alert("Seja bem-vindo "+user+"");
15           return true;
16         }else{
17           window.alert("Usuário ou senha inválido");
18           return false;
19         }
20       }
21     </script>
22   </head>
23
24   <body>
25     <form action="pagina1.html" name="form1" method="post" onsubmit="return validacao();">
26       <center>
27         <table>
28           <tr>
29             <td>Digite o nome de usuário</td>
30             <td><input type="text" name="usuario"></td>
31           </tr>
32           <tr>
33             <td>Digite a senha:</td>
34             <td><input type="password" name="senha"></td>
35           </tr>
36         </table>
37         <table>
38           <tr>
39             <td><input type="submit" value="Logar"></td>
40             <td><input type="reset" value="Limpar"></td>
41           </tr>
42         </table>
43       </center>
44     </form>
45   </body>
46 </html>

```

Vamos analisar nossa função validacao():

```

<script type="text/javascript">
  function validacao(){
    var user = "jonas";
    var password = "qwe123";
    if ((user == document.form1.usuario.value)
      &&(password==document.form1.senha.value)){
      window.alert("Seja bem-vindo "+user+"");
      return true;
    }else{
      window.alert("Usuário ou senha inválido");
      return false;
    }
  }
</script>

```

Observe que nossa função retorna **true** se o usuário e senha digitados no componente de name “usuario” e de name “senha” do formulário de name “form1”, sejam for “jonas” e “qwe123” respectivamente, caso contrário ela retorna **false**.

Esse valor retornado pela função **validacao()** irá ser salvo no evento onsubmit que caso seja **false** irá impedir que o formulário seja direcionado para a página html de nome pagina1.html , caso contrário irá permitir o redirecionamento da página.

- Atenção**

Perceba que adicionamos a palavra reservada **return** antes da chamada de nossa função, essa palavra foi utilizada com intuito de que o evento onsubmit espere e receba um retorno da função validacao().

Evento onmouseover e onmouseout

O evento onmouseover ocorre quando o ponteiro do mouse se posiciona sobre um elemento da página HTML, já o evento onmouseout acontece o oposto, o evento é disparado quando retiramos o ponteiro do mouse de sobre um determinado elemento.

Sintaxe em HTML:

```

<elemento onmouseover="AlgumCodigoJavaScript">
<elemento onmouseout="AlgumCodigoJavaScript">

```

Exemplo:

```
<!DOCTYPE html>
<html>
<head>
<title>Usando eventos no Javascript</title>

<script type="text/javascript">
function mOver(obj) { obj.innerHTML="Obrigado" }
function mOut(obj) { obj.innerHTML="Passe o mouse" }
</script>
</head>
<body>
<div onmouseover="mOver(this)" onmouseout="mOut(this)" style="background-
color:gray;width:120px;height:20px;padding:40px;">Passe o mouse
</div>
</body>
</html>
```

9.4 Eventos: onfocus e onblur

O evento onfocus ocorre quando um elemento/objeto recebe o foco. Ex: quando você clica em um campo de texto para digitar algo, aquele elemento ganhou foco e nessa ação é gerado um evento do tipo onfocus, já o onblur é o oposto o evento é gerado quando o elemento perde o foco, ou seja você tem 2 campos de textos para digitar seus dados. O primeiro você deve digitar seu nome e no outro o seu email. Após digitar o seu nome você terá que digitar o email, certo? Se você tivesse colocado um evento onBlur no campo onde digitou o nome, ao clicar no campo de e-mail um evento do tipo onBlur será gerado, pois o campo nome acaba de perder o foco na aplicação.

```
<elemento onfocus="AlgumCodigoJavaScript">
<elemento onblur="AlgumCodigoJavaScript">
```

Sintaxe em HTML:**Exemplo:**

Vamos implementar dois campos nome e e-mail, os quais inicialmente estarão preenchidos com a String “digite aqui” quando for dado foco a um dos campos o evento onfocus será gerado e o campo de texto que ganhou foco e que estava inicialmente com uma frase se encontrará vazio para ser digitado uma informação do usuário, caso o usuário não digite nenhuma informação, por esquecimento, e retire o foco do campo de texto. O evento onblur será gerado e uma frase em vermelho irá surgir dentro do campo que perdeu o foco “campo obrigatório”.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Usando eventos no Javascript</title>
6 <script type="text/javascript">
7   function alertaCampoVazio(campo){
8     if (campo.value == '') {
9       campo.value = 'campo obrigatório';
10      campo.style.color = "red";
11    } else{
12      if (campo.value == 'digite aqui' || campo.value == 'campo obrigatório') {
13        campo.value = '';
14        campo.style.color = "black";
15      };
16    }
17  }
18 </script>
19 </head>
20 <body>
21 <label>Nome:</label>
22 <input value="digite aqui" type="text"
23   onblur="alertaCampoVazio(this);"
24   onfocus="alertaCampoVazio(this);">
25 <label>E-Mail:</label>
26 <input value="digite aqui" type="text"
27   onblur="alertaCampoVazio(this);"
28   onfocus="alertaCampoVazio(this);">
29 </body>
30 </html>
31
```

9.5 Evento: onchange

O evento onchange é disparado sempre que o texto de um elemento foi alterado. Nós podemos usá-lo para fazer uma verificação em um valor digitado pelo usuário e informá-lo ou instruí-lo a digitar novamente. Este evento está disponível para os campos texto, áreas de texto e listas de seleção.

Sintaxe em HTML:

```
<elemento onchange="AlgumCodigoJavaScript">
```

Exemplo:

Como demonstração do uso desse evento, vamos validar um campo de senha. Iremos criar um campo para o usuário digitar sua senha e outro para que ele repita a senha digitada. O evento onchange será gerado quando o usuário modificar o conteúdo da caixa senha e fazer com que a mesma perca o foco. Quando o evento for gerado será verificado se o tamanho da senha digitada é maior ou igual a 6, caso seja verdade a sentença, a String “Senha válida” será exibida em uma caixa de diálogo alert, caso contrário será exibido a frase “Tamanho da senha inválido, digite uma senha de no mínimo 6 dígitos”, o campo será limpo e ganhará o foco da aplicação novamente.

Veja abaixo:


```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Usando eventos no Javascript</title>
    <script type="text/javascript">
      function valida(campo){
        if (campo.value.length >=6) {
          alert("Senha válida");
        } else{
          alert("A senha deve ter no mínimo 6 digitos");
          campo.value = '';
          campo.focus();
        }
      };
    </script>
  </head>
  <body>
    <label>Senha:</label>
    <input value="" type="password" onchange="valida(this);">
    <label>Repita a senha:</label>
    <input value="" type="password">
  </body>
</html>

```

9.6 Eventos: onload e onunload

O onload é um dos eventos mais importantes em JavaScript. Ele é disparado sempre que a página estiver carregada por completo. Todas as vezes que você escrever um script que só possa ser executado usando outros elementos da página, é sempre uma boa idéia usar este evento para dar início à execução do código. Por só ocorrer quando a página estiver carregada por completo, este evento evita que o seu código tente acessar algum elemento que não esteja baixado para a página ainda.

Já o evento onunload é disparado assim que usuário encerra a execução da página atual para exibir uma nova página. Ele pode ser usado para dar alguma informação, agradecer ao usuário a sua visita ou exibir uma nova página. Este evento também ocorre se você usar o botão Voltar do browser.

Sintaxe m HTML:

```

<body onload="AlgumCodigoJavaScript">
<body onunload="AlgumCodigoJavaScript">

```

Exemplo:

Vamos ver um exemplo da aplicação destes eventos. Queremos exibir uma mensagem de boas-vindas

quando o visitante acessar a nossa página e outra mensagem quando ele sair da nossa página.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Usando eventos no Javascript</title>
    <script type="text/javascript">
      function saudacaoChegada(){
        alert("Seja bem vindo ao nosso site!");
      }
      function saudacaoSaida(){
        alert("Volte sempre ao nosso site!");
      }
    </script>
  </head>
  <body onload="saudacaoChegada()" onunload="saudacaoSaida()">
    <h1>Teste do onload e onunload</h1>
    <p>
      "Tinha um monte de coisas que eu queria fazer...<br>
      Eu quero ser gerente, eu quero tbm ser astronauta...<br>
      Eu quero ter minha propria loja de bolos...<br>
      Eu quero ir numa loja de rosquinhas e dizer "eu quero todas!"...<br>
      Eu quero poder viver umas 5 vezes...<br>
      Entao eu renasceria em 5 cidades diferentes...<br>
      Me encheria de coisas diferentes e deliciosas 5 vezes cada...<br>
      E entao nessas 5 vezes... eu me apaixonaria pela mesma pessoa" <br>
      <i>Inoue Orihime - Bleach</i>
    </p>
  </body>
</html>

```

Atenção! O evento onunload não é suportado pelo navegador Google Chrome ou Opera, nem por algumas versões do Mozilla Firefox.

Exercícios Propostos

Criar um contador do número de dias transcorridos desde uma determinada data, conforme formulário mostrado abaixo:

Entre com a data no formato MM/DD/AAAA

Data:

Transcorridos dias.

Crie um programa que apresente cinco retângulos de 100px por 200 px de cores vermelho, verde, azul amarelo e rosa. O sistema deverá então, randomicamente, o retângulo que contém um prêmio. Quando o usuário clicar em um dos retângulos, o sistema deverá então informar se aquele é o premiado.

Caso, contrário, deverá informar que o usuário errou, informando qual era o correto. (Dica: estude o método random() contido no Objeto Math)

Crie um botão em um ficheiro html, que ao ser clicado aparece um número aleatório entre 0 e 50 e 50

Crie um botão em um ficheiro html, que ao ser clicado aparece um número aleatório entre 0

Crie uma caixa de texto em html. Crie um botão em html que chama código em JavaScript

que dependendo do valor introduzido mostra uma mensagem diferente;

- Entre 0 e 10, 10 excluído, mostra “Insuficiente”;
- Entre 10 e 14, 14 excluído, mostra “Bom”;
- Maior que 14, mostra “Muito Bom”

Crie uma caixa de texto e um botão em html; ao clicar no botão, chama uma função com um parâmetro que é o valor que está dentro da caixa de texto; A função mostra mensagens de acordo com o parâmetro da caixa de texto; por exemplo, no caso de ter introduzido três, a função seria chamado com o parâmetro 3, assim, as mensagens seriam “AIA1”, “AIA 2” e “AIA 3”

10.0 Document Object Model – DOM

Para ter um domínio completo de JavaScript, é preciso que você passe pelo estudo do DOM (Document Object Model), que é o modelo dos objetos que formam a hierarquia de toda a estrutura de uma página web. Essa estrutura começa a partir da janela do navegador e vai até o último elemento da página.

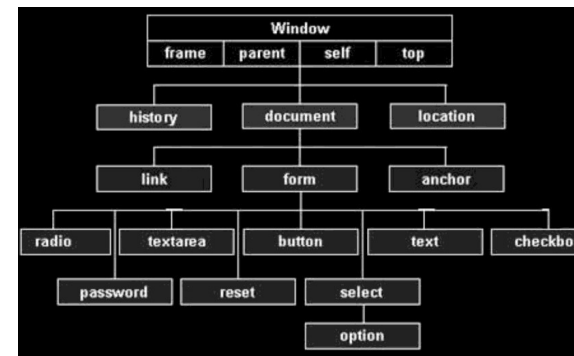
Para entendermos melhor o que é uma hierarquia de objetos, considere o seguinte exemplo.

Suponhamos que você esteja abrindo a porta de seu carro. No momento que abre a porta, se pudéssemos demonstrar em código o que você acaba de fazer, teríamos algo como o mostrado em seguida:

```
carro.porta.fechadura.aberta = true;
```

Observe que acessamos a propriedade aberta da porta, que retorna dois valores. Se estiver aberta, o valor será true e se estiver fechada, será false. Observe o ponto (.) que separa os elementos da hierarquia.

Em JavaScript as coisas não são diferentes. Observe a figura em seguida que mostra bem toda a hierarquia dos objetos em JavaScript:



Vamos agora escrever uma página HTML bem simples para que possamos ver o DOM em ação. Crie uma nova página HTML, digitando o código seguinte no seu editor favorito. Veja como a página ficará.

Lembre-se de criar as imagens que farão parte da página.

Demonstração do DOM em JavaScript



Para receber nossa newsletter digite o seu nome:


```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Usando DOM</title>

  </head>
  <body align="center">
    <h3>Demonstração do DOM em JavaScript</h3>
    
    <a href="http://ead.projetojovem.seduc.ce.gov.br/ejovem/"></a>
    <p>Para receber nossa newsletter digite o seu nome:</p>
    <form name="form1">
      <input type="text" name="text1">
      <input type="button" value="Enviar"><br><br>
    </form>
  </body>
</html>
```

E aqui está o código para a página:

Observe que a partir de agora, sempre que você criar um formulário, deverá dar nome a ele. Isso é fácil de entender. Como todos os elementos de uma página seguem o modelo de hierarquia, você deve segui-la para conseguir acessar as propriedades dos objetos. Vamos ver algum exemplo como isso é possível.

Depois de criada a página apresentada, salve-a em uma pasta. É essa página que usaremos para os nossos testes.

O primeiro script que vamos criar vai obter o título da página e exibi-lo em uma caixa de mensagem.

Abra a nossa página de testes e digite o código seguinte dentro das tags <head></head>.

```
<script type="text/javascript">
  window.alert(document.title);
</script>
```

Observe que tudo que fizemos aqui foi criar uma mensagem alert e definir que a mensagem a ser exibida será o título da página. Para acessar a propriedade title do objeto document, você só precisa usar a linha de código:

```
document.title;
```

Como este objeto está logo abaixo do objeto window na hierarquia, você pode perfeitamente acessar a propriedade acima, usando a linha de código seguinte:

```
window.document.title;
```

Por padrão, não é necessário usar o nome do objeto window sempre que for acessar o objeto document. Por esta razão nós o omitimos no script.

O nosso próximo script é um pouco mais avançado. Agora vamos criar uma função que altera o título da página em tempo de execução. Pense como uma visitante ficaria satisfeito se você exibisse o nome dele na barra de títulos do navegador.

Ainda com a nossa página de testes, digite o código abaixo na parte <body></body> da página para criarmos um botão. Para separar esse botão do outro conteúdo da página, digite algumas quebras de linhas

 para colocá-lo mais abaixo na página:

```
<form name="form1">
  <input type="text" name="text1">
  <input type="button" value="Enviar"><br><br>
  <input type="button" value="Alterar o Título da página" onClick="alterarTitulo()">
</form>
```

```
<script type="text/javascript">
  window.alert(document.title);
  function alterarTitulo () {
    window.document.title = "Seja bem vindo Samia";
  }
</script>
```

Agora, com o botão devidamente criado, digite o código seguinte na parte <head></head> da página:

Nossa página ficará dessa forma.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Usando DOM</title>
    <script type="text/javascript">
      window.alert(document.title);
      function alterarTitulo () {
        window.document.title = "Seja bem vindo Samia";
      }
    </script>
  </head>
  <body align="center">
    <h3>Demonstração do DOM em JavaScript</h3>
    
    <a href="http://ead.projetojovem.seduc.ce.gov.br/ejovem/"></a>
    <p>Para receber nossa newsletter digite o seu nome:</p>
    <form name="form1">
      <input type="text" name="text1">
      <input type="button" value="Enviar"><br><br>
      <input type="button" value="Alterar o Título da página" onclick="alterarTitulo()">
    </form>
  </body>
</html>
```

Salve a página e execute.

Você vai perceber que, assim que clica no botão, ele chama a função alterarTitulo() e o título da página é alterado. Legal, não?

Agora, que tal colocar o nome do visitante na barra de títulos? Vamos ao nosso script:

O código seguinte pede que o visitante digite o seu nome e depois o exibe na barra de títulos juntamente com o título da página. Digite o código seguinte na parte <body></body> da página de teste:

```
<script type="text/javascript">
  var nome = window.prompt("Qual é o seu nome?", "Seu nome");
  var titulo = document.title + " - Visitante: " + nome;
  document.title = titulo;
</script>
```

Que tal usar isso em alguma página? Tenho certeza de que os seus visitantes ficarão deslumbrados.

Vamos entender um pouco como o código foi feito.

Primeiro nós criamos uma variável chamada nome e atribuímos a ela uma mensagem prompt, perguntando o nome do visitante. Criamos também uma variável chamada "titulo" que receberá o texto do título do documento mais o valor da variável "nome", que será informada pelo visitante. A partir daí é só alterar o título por meio da linha de código:

```
document.title = titulo;
```

Fácil, não?

Fico imaginando o que seria de nós, programadores em JavaScript, se não pudéssemos acessar os objetos por meio da hierarquia do DOM do JavaScript.

O script seguinte mostra como acessar uma caixa de texto e pegar o valor que foi digitado pelo usuário. Observe que sempre que quiser realizar tal tarefa, terá que dar nomes tanto ao formulário quanto aos seus elementos. Se você ainda não tem um conhecimento muito aprofundado de HTML, veja como dar nomes aos elementos:

```
<form name="meu_form">
  <input type="text" name="texto1">
```

Quando você dá nomes aos elementos, pode acessá-los e manipular suas propriedades do jeito que quiser. Você pode dar qualquer nome aos seus elementos. Apenas tente dar nomes sugestivos que faça com que você se lembre mais facilmente na hora de acessar cada um deles.

Voltando à nossa página de teste, você pode ver que temos um formulário chamado form1, uma caixa de texto chamada text1 e um botão. Observe que aqui não demos um nome ao botão, mas se quisermos acessar suas propriedades (faremos isso no próximo script), temos que dar um nome a ele.

O script que vamos escrever agora exibe uma mensagem assim que o usuário digita o nome na caixa de texto e clica no botão Enviar.

O texto a ser exibido usa o valor digitado na caixa de texto. Vamos ao código: Digite este código na parte <head></head> de sua página:


```
function enviar () {  
    var nome = document.form1.text1.value;  
    window.alert("Você digitou: " + nome);  
}
```

Agora, precisamos chamar esta função a partir do botão Enviar. Para tanto, vá até o botão e digite a seguinte linha de código:

```
onclick="enviar()"
```

O seu código completo para o botão deve ficar assim:

```
<input type="button" value="Alterar o Título da página" onclick="alterarTitulo()">
```

Salve a página, execute, digite o seu nome na caixa de texto e clique o botão Enviar. Se você realizou todos os procedimentos corretamente, verá uma caixa de mensagem exibindo o texto Você digitou "seu nome". Como nos exemplos anteriores, pratique bastante alterando valores e mixando estes scripts com as técnicas aprendidas anteriormente.

Exercícios Propostos

Crie uma função que seja capaz de receber a informação de 2 campos de texto e em seguida mostre no HTML os valores digitados.

No mesmo formulário melhore a função verificando se existe ou não algo digitado em um dos campos, se não tiver nada digitado, o html deverá mostrar uma mensagem dizendo campo vazio.

Crie um formulário para que irá receber 3 valores, converta-os para valores numéricos usando Objeto Number() e imprima a média dos valores digitados.

Crie um formulário com um campo de texto para uma nota e um combobox que terá as matérias: Física, matemática e português. Mostrei abaixo qual nota foi digitada e qual curso foi selecionado.