

Multimídia

Prof. Rodrigo Martins

rodrigo.martins@francomontoro.com.br

Cronograma da Aula

Formulários

Formulários

Um formulário é um componente de uma página web que possui controles, como campos de texto, botões, caixas de seleção ou de combinação, listas etc., para permitir a interação do usuário com o site.

Criação de Formulários

Os formulários podem ser criados por meio do elemento `<form>`. No interior desse elemento pode ser utilizado um conjunto de outros elementos para gerar os controles (legendas, campos, botões e agrupamentos) do formulário. Os elementos para criar os controles do formulário são `<input>`, `<textarea>`, `<legend>`, `<label>`, `<fieldset>`, `<datalist>`, `<keygen>`, `<output>`, `<select>`, `<option>` e `<optgroup>`.

A tag `<form>`

Todo formulário começa com o elemento `<form>`. Em seu interior, são colocados os campos, na maioria das vezes representados pelo elemento `<input>` que, se não tiver atributos, por padrão, exibe na tela um campo de texto. Para identificar um controle, como, por exemplo, um campo texto criado pelo elemento `<input>`, utiliza-se o elemento `<label>`.

Cada parte de um formulário é considerado um parágrafo, e é normalmente separado do resto do formulário por meio do elemento `<p>`.

A tag <form> – Exemplo

```
<html>
  <head>
    <title>Exemplo de formulário</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form>
      <p>
        <label>
          Nome:<input>
        </label>
      </p>
    </form>
  </body>
</html>
```

Nome:

A tag <input>

O elemento <input> gera um campo em que o usuário pode selecionar ou entrar com dados de várias maneiras. O que define como será a entrada é o atributo type.

```
<form>
  <p>
    <label>
      Nome:<input type="text">
    </label>
  </p>
  <p>
    <label>
      Senha:<input type="password">
    </label>
  </p>
</form>
```

Nome:

Senha:

Campo do tipo telefone

Para definir um campo de entrada de telefone, é necessário que o elemento `<input>` possua o atributo `type` com valor “tel”.

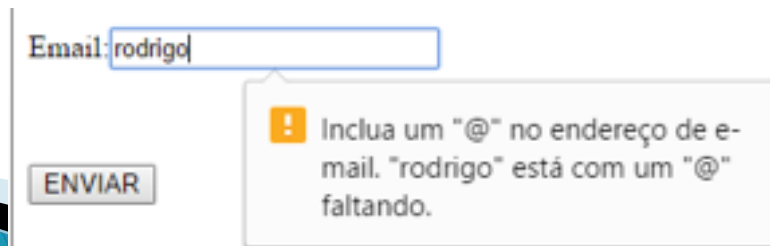
```
<form>  
  <p>  
    <label>  
      Telefone:<input type="tel" required>  
    </label>  
  </p>
```

Telefone:

Campo do tipo email

Para definir um campo de entrada de email, é necessário que o elemento `<input>` possua o atributo `type` com valor "email".

```
<form method="post">  
  <p>  
    <label>  
      Email:<input type="email">  
    </label>  
  </p><br />  
  <p>  
    <input type="submit" value="ENVIAR">  
  </p>  
</form>
```



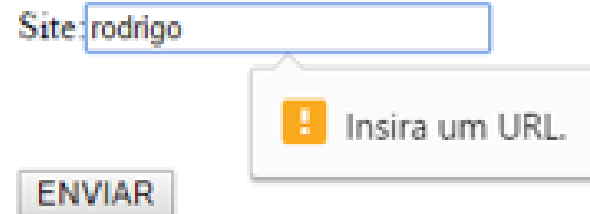
Email:

! Inclua um "@" no endereço de e-mail. "rodrigo" está com um "@" faltando.


Campo do tipo url

Para definir um campo de entrada de endereço web, é necessário que o elemento `<input>` possua o atributo `type` com valor “url”.

```
<form method="post">
  <p>
    <label>
      Site:<input type="url">
    </label>
  </p><br />
  <p>
    <input type="submit" value="ENVIAR">
  </p>
</form>
```



Site:

 Insira um URL.

Campo do tipo number

O campo “number” é utilizado nos elementos `<input>` para entrada de valores numéricos. É possível definir restrições de entrada para esse tipo de campo.

```
<form method="post">
```

```
<p>
```

```
<label>
```

```
Nota:<input type="number" name="nota"
           min="0" max="10" step="0.5">
```

```
</label>
```

```
<p>
```

```
<input type="submit" value="ENVIAR">
```

```
</p>
```

```
</form>
```

Nota: 1,5

ENVIAR

Nota: 1,15

ENVIAR



Insira um valor válido. Os dois valores válidos mais próximos são 1 e 1,5.

Campo do tipo range

O tipo “range” é usado para campos de entrada que devem receber um valor numérico contido em uma faixa de valores numéricos.

```
<form method="post">
  <p>
    <label>
      Volume:<input type="range" name="volume"
                    min="0" max="10" step="2">
    </label>
  <p>
    <input type="submit" value="ENVIAR">
  </p>
</form>
```

Volume:

ENVIAR

Campos para entrada de data e hora

Date – Permite selecionar o dia, mês e ano

```
<form method="post">  
  <p>  
    <label>  
      Data:<input type="date" name="data">  
    </label>  
  <p>  
    <input type="submit" value="ENVIAR">  
  </p>  
</form>
```

Data:

Campos para entrada de data e hora

Month – Permite selecionar mês e ano

```
<form method="post">  
  <p>  
    <label>  
      Mês:<input type="month" name="mes">  
    </label>  
  <p>  
    <input type="submit" value="ENVIAR">  
  </p>  
</form>
```

Mês: maio de 2016

ENVIAR

Campos para entrada de data e hora

Week – Selecciona dia da semana e ano

```
-  
<form method="post">  
  <p>  
    <label>  
      Semana:<input type="week" name="semana">  
    </label>  
  <p>  
    <input type="submit" value="ENVIAR">  
  </p>  
</form>
```

Semana: Semana 04, 2016

ENVIAR

Campos para entrada de data e hora

Time – Permite seleccionar hora e minuto

```
<form method="post">  
  <p>  
    <label>  
      Hora:<input type="time" name="hora">  
    </label>  
  <p>  
    <input type="submit" value="ENVIAR">  
  </p>  
</form>
```

Hora: --:--

ENVIAR

Campos para entrada de data e hora

datetime-local – Permite selecionar dia, mês, ano, hora e minuto.

```
-  
<form method="post">  
  <p>  
    <label>  
      Data e Hora:<input type="datetime-local" name="datahora">  
    </label>  
  <p>  
    <input type="submit" value="ENVIAR">  
  </p>  
</form>
```

Data e Hora:

ENVIAR

Validação de entrada de valores de data e hora

Se for inserido um valor não válido em campos que manipulam data e hora, será exibida a mensagem indicando que o valor é inválido.

```
<form method="post">
  <p>
    <label>
      Tempo de entrega do Produto:<input type="time" min="19:00"
                                         max="23:00" step="300" name="hora">
    </label>
  </p>
  <input type="submit" value="ENVIAR">
</form>
```

Tempo de entrega do Produto: 19:02 x

ENVIAR



Insira um valor válido. Os dois valores válidos mais próximos são 19:00 e 19:05.

Validação de entrada de valores de data e hora (cont.)

```
<form method="post">
  <p>
    <label>
      Data:<input type="date"
        min="2016-01-01" max="2016-12-31" value="2016-05-03">
    </label>
  <p>
    <input type="submit" value="ENVIAR">
  </p>
</form>
```

Data: 03/05/2016 x ▾

ENV maio de 2016 ▾ ◀ ● ▶

dom	seg	ter	qua	qui	sex	sáb
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Campo do tipo hidden

O elemento `<input>` com o atributo `type` definido como “hidden” indica que o campo não aparece na tela para o usuário. Esse tipo de campo é utilizado para transportar valores do cliente para o servidor para a identificação do formulário, ou outros valores que a aplicação que executa no servidor precisa para executar algum procedimento.

Campo do tipo hidden (cont)

Quando o servidor receber o formulário, a aplicação pode capturar o conteúdo do atributo value por meio do conteúdo do atributo name e identificar que o formulário é de contato.

```
<label>  
  <input type="hidden" name="formulario" value="contato">  
</label>
```

Campo do tipo file

Quando o elemento `<input>` possui o atributo `type` configurado com o valor "file", o campo permite a seleção de um arquivo para upload.

```
<html>
  <head>
    <title>Upload</title>
    <script type="text/javascript">
      function atualizarNomeArquivo(caminho) {
        document.getElementById("imagem").src = caminho;
      }
    </script>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form>
      <p>
        <input type="file" name="imagem"
          accept="image/*" onchange="atualizarNomeArquivo(this.value)">
      </p>
    </form>
    <p>
      
    </p>
  </body>
</html>
```

Escolher arquivo Nenhum arquivo selecionado



Botões dos tipos submit, button e reset

Quando o elemento `<input>` possui o atributo `type` configurado com o valor "file", o campo permite a seleção de um arquivo para upload.

```
<!DOCTYPE html>|
<html>
  <head>
    <title>Cadastro</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script type="text/javascript">
      function validar() {
        var n = document.formulario.pnome.value;
        if (n=="") {
          alert("O campo nome está vazio");
          document.formulario.pnome.focus;
        }
      }
    </script>
  </head>
  <body>
```

Botões dos tipos submit, button e reset (cont)

```
<body>
  <form name="formulario" action="teste.jsp" method="post">
    <p>
      <label>
        Nome:<input type="text" name="pname">
      </label>
    </p>
    <p>
      <label>
        Sobrenome:<input type="text" name="sname">
      </label>
    </p>
    <p>
      <input type="submit" value="Enviar">
      <input type="reset" value="Limpar">
      <input type="button" value="Validar Nome" onclick="validar();">
    </p>
  </form>
</body>
</html>
```


Botões dos tipos submit, button e reset (cont)

Nome:

Sobrenome:

localhost:8383 diz:

O campo nome está vazio

OK

Botões dos tipo image

Quando o elemento `<input>` possui o atributo `type` configurado com o valor "image", o botão pode ser uma imagem que será utilizada para submeter o formulário .

```
<form name="formulario" action="teste.jsp" method="post">
  <p>
    <label>
      Nome:<input type="text" name="pnome">
    </label>
  </p>
  <p>
    <label>
      Sobrenome:<input type="text" name="snome">
    </label>
  </p>
  <p>
    <input type="image" src="botao.jpg" alt="botao">
  </p>
</form>
```

Nome:

Sobrenome:



Botão do tipo radio

Estado Civil:

- ☐ Casado
- ☐ Solteiro
- ☐ Divorciado

Campos radio (radio buttons) servem para permitir que o usuário selecione uma (e somente uma) dentre muitas opções de escolha disponibilizadas em um formulário. Campos radio são criados pelo elemento `<input>` contendo o atributo `type` com o valor "radio".

```
<body>
  <fieldset>
    <legend>Estado Civil:</legend>
    <p>
      <label>
        <input type="radio" name="estadoCivil" value="Casado">Casado
      </label>
    </p>
    <p>
      <label>
        <input type="radio" name="estadoCivil" value="Solteiro">Solteiro
      </label>
    </p>
    <p>
      <label>
        <input type="radio" name="estadoCivil" value="Divorciado">Divorciado
      </label>
    </p>
  </fieldset>
</body>
```

Botão do tipo checkbox

Produtos

☐ Computador

☐ Impressora

☐ Monitor

Para permitir que o usuário selecione várias das muitas opções disponíveis em uma sequência de itens, pode-se utilizar elementos `<input>` com o atributo `type` contendo o valor "checkbox".

```
<body>
  <fieldset>
    <legend> Produtos </legend>
    <p>
      <label>
        <input type="checkbox" name="computador" value="Computador">Computador
      </label>
    </p>
    <p>
      <label>
        <input type="checkbox" name="impressora" value="Impressora">Impressora
      </label>
    </p>
    <p>
      <label>
        <input type="checkbox" name="monitor" value="Monitor">Monitor
      </label>
    </p>
  </fieldset>
</body>
```

Campo do tipo search

O tipo search é usado para campos utilizados em pesquisas, como uma busca no site, ou de pesquisa nos buscadores web.


```
<p>  
  <label>  
    O que deseja pesquisar?  
    <input type="search" name="pesquisa">  
  </label>  
</p>
```

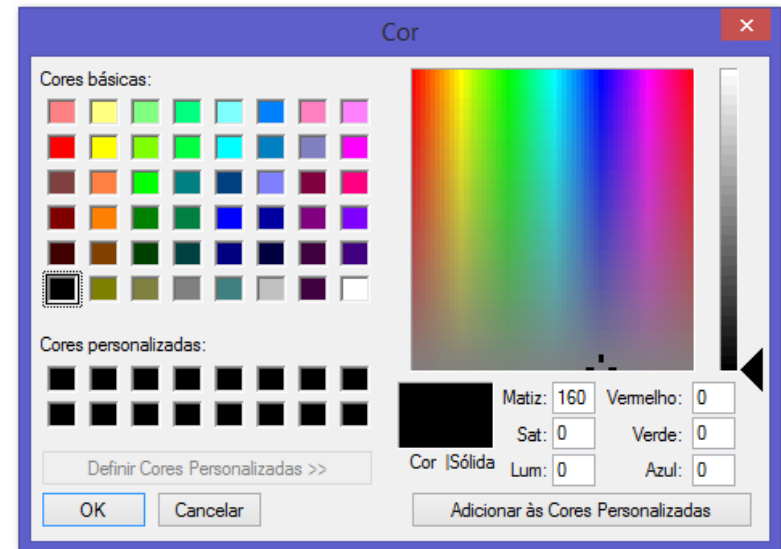
O que deseja pesquisar? x

Campo do tipo color

Campos do tipo color são usados para entrada de cores. Nos campos para seleção de cores o atributo type deve receber o valor “color”.

```
<p>  
  <label>  
    Cor:  
    <input type="color" name="cor">  
  </label>  
</p>
```

Cor: 



O elemento <button>

O elemento <button> é utilizado para criar botões independentes ou não dos formulários. No interior do elemento <button> você pode colocar conteúdo, como texto ou imagens. Esta é a grande diferença entre o elemento <button> e o elemento <input> com o atributo type igual a "button".

```
<body>
  <p>
    <button type="button">
      Clique aqui
    </button>
  </p>
  <p>
    <button type="button">
      
    </button>
  </p>
</body>
```



O elemento <button> (cont)

```
<form name="formulario" action="teste.php" method="post">
  <p>
    <label>
      Nome:<input type="text" name="pnome">
    </label>
  </p>
  <p>
    <button type="button" onclick="alert('Botão Clicado.')">
      Clique aqui
    </button>
    <button type="submit">
      Enviar
    </button>
    <button type="reset">
      Limpar
    </button>
  </p>
</form>
```

Nome:

Clique aqui

Enviar

Limpar

localhost:8383 diz:

Botão Clicado.

☐ Impedir que esta página crie caixas de diálogo adicionais.

OK

O elemento <textarea>

Campos criados com o elemento <textarea> permitem a digitação de um texto com múltiplas linhas.

```
<p>
  <label>
    Deixe sua sugestão:<br />
    <textarea cols="30" rows="5"></textarea>
  </label>
</p>
```

Deixe sua sugestão:

Os dados do produto
disponíveis nessa página não
são suficientes.

O elemento <datalist>

O elemento <datalist> especifica uma lista oculta de opções sugeridas (não obrigatórias) para um campo de entrada. A lista é criada com elementos <option> dentro do <datalist>.

```
<p>
  <label>
    Site:<input type="url" list="sites" name="link">
  </label>
</p>
<datalist id="sites">
  <option label="Ig"
    value="http://www.ig.com.br">
</option>
  <option label="Google"
    value="http://www.google.com">
</option>
  <option label="Microsoft"
    value="http://www.microsoft.com.br">
</option>
</datalist>
```

Site:

- <http://www.ig.com.br> Ig
- <http://www.google.com> Google
- <http://www.microsoft.com.br> Microsoft

O elemento <output>

O elemento <output> é usado para diferentes tipos de saída, como cálculos ou saída do script.

```
<html>
  <head>
    <title>OUTPUT</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <p>Multiplicação de dois valores:</p>
    <form onsubmit="return false"
      oninput="resultado.value=valor1.valueAsNumber*valor2.valueAsNumber">
      <input name="valor1" type="number">*
      <input name="valor2" type="number">=
      <output name="resultado"></output>
    </form>
  </body>
</html>
```

Multiplicação de dois valores:

3 * 10 = 30

O elemento <select>

O elemento <select> é usado para criar uma lista drop-down. Os <option> dentro do elemento <select> definir as opções disponíveis na lista.

14			Marca de Carro:
15	-		<select>
16			<option value="volvo">Volvo</option>
17			<option value="saab">Jeep</option>
18			<option value="mercedes">Mercedes</option>
19			<option value="audi">Audi</option>
20			</select>

Os atributos da tag <select>

autofocus: Especifica que a lista suspensa deve automaticamente obter foco quando a página é carregada.

disabled: O atributo disabled é um atributo booleano. Quando presente, ele especifica que a lista drop-down deve ser desativado.

multiple: O atributo múltiplo é um atributo booleano. Quando presente, ele especifica que várias opções podem ser selecionadas de cada vez.

name: O atributo name especifica o nome de uma lista drop-down. O atributo nome é usado para fazer referência a elementos em um JavaScript, ou fazer referência a dados do formulário depois que um formulário é enviado.

required: O atributo obrigatório é um atributo booleano. Quando presente, ele especifica o usuário é obrigado a selecionar um valor antes de enviar o formulário.

size: O atributo de tamanho especifica o número de opções visíveis em uma lista drop-down. Se o valor do atributo de tamanho é maior do que 1, mas menor do que o número total de opções na lista, o navegador irá adicionar uma barra de rolagem para indicar que existem mais opções para ver.

Os atributos da tag <select>

```
14  
15   
16  
17  
18  
19  
20  
21
```

Marca de Carro:

```
<select>  
  <option value="volvo">Volvo</option>  
  <option value="saab">Jeep</option>  
  <option value="mercedes">Mercedes</option>  
  <option value="audi">Audi</option>  
</select>  
<br>
```

Os atributos da tag <select>

22
23
24
25
26
27
28
29
30
31

Exemplo com autofocus e size


```
<select autofocus size="2">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Jeep</option>  
  <option value="opel">Opel</option>  
  <option value="audi">Audi</option>  
</select>  
<br>
```

Os atributos da tag <select>

```
32      Exemplo com disabled
33      <select disabled>
34          <option value="volvo">Volvo</option>
35          <option value="saab">Jeep</option>
36          <option value="mercedes">Mercedes</option>
37          <option value="audi">Audi</option>
38      </select>
39      <br>
40
41      Exemplo com multiple, name e required<br>
42      <select name="carros" multiple required>
43          <option value="volvo">Volvo</option>
44          <option value="saab">Jeep</option>
45          <option value="opel">Opel</option>
46          <option value="audi">Audi</option>
47      </select>
48  </body>
49  </html>
```


Outros atributos da tag <form>

O atributo **action** especifica para onde enviar o formulário de dados quando um formulário é enviado.

O atributo **accept-charset** especifica as codificações de caracteres que estão a ser utilizados para o envio do formulário.

- UTF-8: Codificação de caracteres para Unicode.
- ISO-8859-1: Codificação de caracteres para o alfabeto latino

```
49 <form action="cadastrar.php" method="get" accept-charset="ISO-8859-1">
50     Primeiro nome: <input type="text" name="pNome"><br>
51     Último nome: <input type="text" name="uNome"><br>
52     <input type="submit" value="ENVIAR">
53 </form>
```

Outros atributos da tag <form>

O atributo `method` especifica como enviar os dados (o formulário de dados são enviados para a página especificada no atributo `action`). O formulário de dados podem ser enviados como variáveis de URL (com `method = "get"`) ou como pós HTTP transação (com `method = "post"`).

Notas sobre GET:

- ▶ Acrescenta form-data para o URL em pares nome / valor
- ▶ O comprimento de um URL é limitado (cerca de 3000 caracteres)
- ▶ Nunca use GET para enviar dados que requer extrema segurança(Será visível na URL)
- ▶ Útil para envios de formulários onde um usuário deseja marcar o resultado
- ▶ GET é melhor para dados não-seguros, como sequências de consulta no Google

Notas sobre POST:

- ▶ Acrescenta-forma de dados no interior do corpo do pedido HTTP (dados não é mostrada é em URL)
- ▶ Não tem limitações de tamanho envios de formulários com POST não podem ser marcadas

Outros atributos da tag <form>

O atributo **autocomplete** especifica se um formulário deve ter autocomplete ligado ou desligado.

Quando o preenchimento automático está ligado, o navegador irá completar automaticamente com base nos valores que o usuário tenha entrado antes.

O valor padrão é on.

```
<form action="cadastrar.php" name="frmCadastrar" method="get"
      autocomplete="off" accept-charset="ISO-8859-1">
  Primeiro nome: <input type="text" name="pNome"><br>
  Último nome: <input type="text" name="uNome"><br>
  <input type="submit" value="ENVIAR">
</form>
```

Referências Bibliográficas

–Teruel, Evandro Carlos

HTML 5: Guia Prático 1ed. São Paulo: Érica, 2011

Fim

