

Formulários HTML5

Alisson Chiquitto¹

¹Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Cianorte, 2014

- 1 Introdução
- 2 Criando um formulário
 - Formulários
- 3 Elementos para entrada de dados do usuário
 - Elemento input
 - Listas
 - Elemento de texto com múltiplas linhas
- 4 Recursos extras de formulários
- 5 Outros elementos
- 6 Atividades

- 1 Introdução
- 2 Criando um formulário
 - Formulários
- 3 Elementos para entrada de dados do usuário
 - Elemento input
 - Listas
 - Elemento de texto com múltiplas linhas
- 4 Recursos extras de formulários
- 5 Outros elementos
- 6 Atividades

- Um formulário é utilizado para passar dados para o servidor;
- Um formulário *HTML* pode conter elementos *input* como campos de texto, *checkboxes*, *radio buttons*, botões de submissão, listas de seleção, *textareas*, etc;
- Para criar um formulário HTML use a tag `<form>`;

Formulários HTML

☒ Pessoa Física ☐ Pessoa Jurídica

Nome completo *

CPF *

Sexo *

Data de Nascimento *

Telefone Preferencial * -

Telefone Alternativo -

Figura: Um formulário de exemplo

- 1 Introdução
- 2 Criando um formulário
 - Formulários
- 3 Elementos para entrada de dados do usuário
 - Elemento input
 - Listas
 - Elemento de texto com múltiplas linhas
- 4 Recursos extras de formulários
- 5 Outros elementos
- 6 Atividades

- 1 Introdução
- 2 Criando um formulário
 - Formulários
- 3 Elementos para entrada de dados do usuário
 - Elemento input
 - Listas
 - Elemento de texto com múltiplas linhas
- 4 Recursos extras de formulários
- 5 Outros elementos
- 6 Atividades

O elemento *form*

- “O elemento `<form>` representa uma coleção de elementos associados ao formulário, alguns dos quais podem representar valores editáveis que podem ser submetidos a um servidor para processamento.”
- Para criar um formulário *HTML* usa-se a tag `<form>`;

Atributos do elemento *form*

name

Representa o nome do formulário. O valor não pode ser uma *string* vazia.

action

Representa a *URL* do serviço que irá manipular os dados do formulário.

Criação de um formulário com HTML

```
<form name="form1" action="processar.php">  
  <!-- Elementos do form -->  
</form>
```

Figura: Criação de um formulário com HTML

- 1 Introdução
- 2 Criando um formulário
 - Formulários
- 3 Elementos para entrada de dados do usuário
 - Elemento input
 - Listas
 - Elemento de texto com múltiplas linhas
- 4 Recursos extras de formulários
- 5 Outros elementos
- 6 Atividades

- 1 Introdução
- 2 Criando um formulário
 - Formulários
- 3 Elementos para entrada de dados do usuário
 - **Elemento input**
 - Listas
 - Elemento de texto com múltiplas linhas
- 4 Recursos extras de formulários
- 5 Outros elementos
- 6 Atividades

O elemento *input*

- Um dos elementos de formulário mais importante que existe no HTML;
- O elemento `<input>` pode variar de muitas maneiras, dependendo do seu atributo *type*;
- Um elemento `<input>` pode ser um campo de texto, *checkboxes*, senha, *radio button*, botões de submissão, etc;

Atributos principais do elemento *input*

name

Representa o nome do elemento. Ao submeter um formulário o nome do elemento será relacionado com o seu respectivo valor.

value

Especifica o valor do elemento. O valor informado pelo usuário será atribuído (automaticamente) a este atributo;

type

Define o formato, tipo de dado e controle associados ao elemento;

O atributo *type*

- O atributo *type* controla o tipo de dado (e controles associados) ao elemento;
- Ele é um atributo enumeado, e os seguintes valores podem ser atribuídos a ele: *hidden*, *text*, *search*, *tel*, *url*, *email*, *password*, *datetime*, *date*, *month*, *week*, *time*, *datetime-local*, *number*, *range*, *color*, *checkbox*, *radio*, *file*, *submit*, *image*, *reset*, *button*;
- O valor *default* deste atributo é *text*;

O elemento *input type=hidden*

- O elemento *input* com o atributo *type="hidden"* representa um valor que não será visualizado ou manipulado pelo usuário;
- <http://dev.w3.org/html5/markup/input.hidden.html>

```
<input  
type="hidden"  
name="elementoX"  
value="Valor oculto do usuário"  
>
```

Figura: Criação de um elemento *hidden*

O elemento *input type=text*

- O elemento *input* com o atributo *type="text"* representa um campo de texto simples de uma linha;
- <http://dev.w3.org/html5/markup/input.text.html>

```
<input  
type="text"  
name="usuario"  
value="Nome do usuário"  
>
```

Figura: Criação de um elemento *text*

O elemento *input type=search*

- O elemento *input* com o atributo *type="search"* representa um campo de texto para entrada de um ou mais termo de busca;
- <http://dev.w3.org/html5/markup/input.search.html>

```
<input  
type="search"  
name="q"  
value="Digite sua pesquisa"  
>
```

Figura: Criação de um elemento *search*

O elemento *input type=tel*

- O elemento *input* com o atributo *type="tel"* representa um campo de texto para entrada de um número de telefone;
- <http://dev.w3.org/html5/markup/input.tel.html>

```
<input  
type="tel"  
name="fone"  
>
```

Figura: Criação de um elemento *tel*

O elemento *input type=url*

- O elemento *input* com o atributo *type="url"* representa um controle para a entrada de uma *URL* absoluta;
- <http://dev.w3.org/html5/markup/input.url.html>

```
<input  
type="url"  
name="site"  
>
```

Figura: Criação de um elemento *url*

O elemento *input type=email*

- O elemento *input* com o atributo *type="email"* representa um controle para a entrada de um ou vários endereços de email;
- <http://dev.w3.org/html5/markup/input.email.html>

```
<input  
type="email"  
name="email"  
>
```

Figura: Criação de um elemento *email*

O elemento *input type=password*

- O elemento *input* com o atributo *type="password"* representa um controle para a entrada de uma senha;
- <http://dev.w3.org/html5/markup/input.password.html>

```
<input  
type="password"  
name="senha"  
>
```

Figura: Criação de um elemento *password*

O elemento *input type=datetime*

- O elemento *input* com o atributo *type="datetime"* representa um controle para a entrada de uma *string* representando uma data e hora global;
- <http://dev.w3.org/html5/markup/input.datetime.html>

```
<input  
type="datetime"  
name="datahora"  
>
```

Figura: Criação de um elemento *datetime*

O elemento *input type=date*

- O elemento *input* com o atributo *type="date"* representa um controle para a entrada de uma *string* representando uma data global;
- <http://dev.w3.org/html5/markup/input.date.html>

```
<input  
type="date"  
name="data"  
>
```

Figura: Criação de um elemento *date*

O elemento *input type=month*

- O elemento *input* com o atributo *type="month"* representa um controle para a entrada de uma *string* representando um mês;
- <http://dev.w3.org/html5/markup/input.month.html>

```
<input  
type="month"  
name="mes"  
>
```

Figura: Criação de um elemento *month*

O elemento *input type=week*

- O elemento *input* com o atributo *type=“week”* representa um controle para a entrada de uma *string* representando uma semana;
- <http://dev.w3.org/html5/markup/input.week.html>

```
<input  
  type="week"  
  name="semana"  
>
```

Figura: Criação de um elemento *week*

O elemento *input type=time*

- O elemento *input* com o atributo *type="time"* representa um controle para a entrada de uma *string* representando uma hora;
- <http://dev.w3.org/html5/markup/input.time.html>

```
<input  
type="time"  
name="hora"  
>
```

Figura: Criação de um elemento *time*

O elemento *input type=number*

- O elemento *input* com o atributo *type=“number”* representa um controle para a entrada de uma *string* representando um número;
- <http://dev.w3.org/html5/markup/input.number.html>

```
<input  
type="number"  
name="n"  
>
```

Figura: Criação de um elemento *number*

O elemento *input type=range*

- O elemento *input* com o atributo *type="range"* representa um controle para a entrada de uma *string* representando um número. Diferente do elemento *type=number*, aqui o valor exato não tem importancia para o usuário;
- <http://dev.w3.org/html5/markup/input.range.html>

```
<input  
type="range"  
name="r"  
>
```

Figura: Criação de um elemento *range*

O elemento *input type=color*

- O elemento *input* com o atributo *type="color"* representa um controle para a entrada de uma *string* representando uma cor hexadecimal;
- <http://dev.w3.org/html5/markup/input.color.html>

```
<input  
type="color"  
name="cor"  
>
```

Figura: Criação de um elemento *color*

O elemento *input type=checkbox*

- O elemento *input* com o atributo *type=“checkbox”* representa um estado ou uma opção que pode ser alternado;
- <http://dev.w3.org/html5/mkuptype/input.checkbox.html>

```
<input  
type="checkbox"  
name="ativo"  
>
```

Figura: Criação de um elemento *checkbox*

O elemento *input type=radio*

- O elemento *input* com o atributo *type="radio"* representa uma seleção de um item de uma lista de itens;
- <http://dev.w3.org/html5/markup/input.radio.html>

```
<input type="radio" name="uf" value="PR">  
<input type="radio" name="uf" value="SC">  
<input type="radio" name="uf" value="RS">
```

Figura: Criação de um elemento *radio*

O elemento *input type=file*

- O elemento *input* com o atributo *type="file"* representa uma lista com zero ou mais arquivos;
- <http://dev.w3.org/html5/mkup/input.file.html>

```
<input  
type="file"  
name="arquivo"  
>
```

Figura: Criação de um elemento *file*

O elemento *input type=submit*

- O elemento *input* com o atributo *type=“submit”* representa um botão para a submissão de um formulário;
- <http://dev.w3.org/html5/markup/input.submit.html>

```
<input  
  type="submit"  
  name="enviar"  
  value="Enviar"  
>
```

Figura: Criação de um elemento *submit*

- 1 Introdução
- 2 Criando um formulário
 - Formulários
- 3 Elementos para entrada de dados do usuário
 - Elemento input
 - **Listas**
 - Elemento de texto com múltiplas linhas
- 4 Recursos extras de formulários
- 5 Outros elementos
- 6 Atividades

Lista simples

- O elemento `<select>` representa um controle que permite o usuário selecionar opções dentre várias opções;
- O elemento `<option>` representa uma opção em um elemento `<select>` ou parte de uma lista de sugestões;
- Na lista simples, o usuário deve selecionar **sempre** uma opção;

```
<select name="ufs">
  <option value="PR">Paraná</option>
  <option value="BA">Bahia</option>
  <option value="AC">Acre</option>
</select>
```

Figura: Uma lista simples

Lista com múltipla seleção

- Na lista com múltipla seleção o usuário pode selecionar desde zero até todas as opções;
- Utilize o atributo *multiple* para mudar a lista simples para uma lista de múltipla seleção;

```
<select name="ufs" multiple>
  <option value="PR">Paraná</option>
  <option value="BA">Bahia</option>
  <option value="AC">Acre</option>
</select>
```

Figura: Uma lista de múltipla seleção

Lista de sugestões

- O elemento `<datalist>` um conjunto de elementos `<option>` que representa opções pré-definidas para outros controladores;
- Este elemento pode ser utilizado como uma lista de sugestões para elementos `<input>`;

```
<input name="sexo" list="sexos">  
<datalist id="sexos">  
  <option value="Feminino">  
  <option value="Masculino">  
</datalist>
```

Figura: Uma lista de sugestões

- 1 Introdução
- 2 Criando um formulário
 - Formulários
- 3 Elementos para entrada de dados do usuário
 - Elemento input
 - Listas
 - Elemento de texto com múltiplas linhas
- 4 Recursos extras de formulários
- 5 Outros elementos
- 6 Atividades

Elemento de texto com múltiplas linhas

- O elemento `<textarea>` representa um controlador de texto com multiplas linhas;
- O conteúdo do elemento representa o seu valor;

```
<textarea name="comentarios">Texto</textarea>
```

Figura: Um editor de texto com múltiplas linhas

- 1 Introdução
- 2 Criando um formulário
 - Formulários
- 3 Elementos para entrada de dados do usuário
 - Elemento input
 - Listas
 - Elemento de texto com múltiplas linhas
- 4 Recursos extras de formulários
- 5 Outros elementos
- 6 Atividades

Habilitando e desabilitando elementos de formulários

- O atributo booleano *disable* habilita/desabilita elementos de formulários;
- Quando presente, o elemento esta desabilitado;
- Elementos desabilitados não são processados durante a submissão do formulário

```
<input  
type="text"  
name="usuario"  
value="Nome do usuário"  
disabled  
>
```

Figura: Um elemento desabilitado

- O atributo booleano *readonly* controla se o valor de um campo poderá ser alterado/editado pelo usuário;
- Quando presente, informa que o valor do elemento não pode ser selecionado pelo usuário;

```
<input  
type="text"  
name="usuario"  
value="Nome do usuário"  
readonly  
>
```

Figura: Um elemento somente leitura

Checando elementos ou opções

- O atributo booleano *checked* controla se um elemento está selecionado;
- Quando presente, informa que o elemento deve ser selecionado;
- Aplicável a elementos *radio* e *checkbox* apenas;
- Para selecionar uma opção de um elemento `<select>`, use o atributo *selected* no `<option>` que deve ser selecionado;

```
<input  
type="checkbox"  
name="ativo"  
checked  
>
```

Figura: Um elemento selecionado

Elementos com preenchimento obrigatório

- O atributo booleano *required* especifica se o preenchimento do elemento é obrigatório;
- Quando presente, informa que o preenchimento do elemento é obrigatório;

```
<input  
type="text"  
name="usuario"  
required  
>
```

Figura: Um elemento de preenchimento obrigatório

Forçando um padrão para preenchimento

- O atributo *pattern* valida o valor de um elemento com relação a uma expressão regular;
- Você pode utilizar atributo *title* para dar ao usuário uma descrição do padrão esperado;
- Pode ser aplicado a elementos do tipo: *text*, *search*, *url*, *tel*, *email* e *password*;

```
<input  
type="text"  
name="usuario"  
required  
pattern="[A-Z]{2}"  
title="Sigla do UF"  
>
```

Figura: Definindo um padrão para o preenchimento

Restrições para valores numéricos

- O atributo *min* define o valor mínimo do elemento, enquanto o atributo *max* define o valor máximo;
- O atributo *step* indica a granularidade do valor do elemento;
- Pode ser aplicado a elementos do tipo: *number* e *range*;

```
<input  
type="number"  
name="n"  
required  
min="0"  
max="100"  
step="5"  
>
```

Figura: Definindo um padrão para o preenchimento

Breve descrição para os elementos

- O atributo *placeholder* define uma “dica” que descreve o valor de entrada do elemento, como um valor exemplo ou uma breve descrição do valor;

```
<input  
  type="text"  
  name="usuario"  
  placeholder="Nome completo"  
>
```

Figura: Um elemento de texto com uma dica de preenchimento

- 1 Introdução
- 2 Criando um formulário
 - Formulários
- 3 Elementos para entrada de dados do usuário
 - Elemento input
 - Listas
 - Elemento de texto com múltiplas linhas
- 4 Recursos extras de formulários
- 5 Outros elementos
- 6 Atividades

Elemento *label*

- O elemento *label* representa uma legenda de algum elemento de formulário;
- Ela pode ser associada com um elemento através do atributo *for*, e do atributo *id* do elemento;

```
<label id="idUsuario">
  Usuario
</label>
<input
  type="text"
  name="usuario"
  id="idUsuario"
>
```

Figura: O elemento *label*

Agrupando elementos

- O elemento *fieldset* representa um conjunto de elementos de formulário opcionalmente agrupados sobre um nome;
- O elemento *legend* representa uma legenda para um elemento *fieldset*;

```
<fieldset>
  <legend>Dados pessoais</legend>

  <input type="text" name="nome">
  <input type="email" name="email">
</fieldset>
```

Figura: Um grupo de elementos

Agrupando opções de listas

- O elemento *optgroup* representa um grupo de elementos *option* sobre uma mesma legenda;

```
<select name="ufs">
  <optgroup label="Sul">
    <option value="PR">Paraná</option>
    <option value="RS">R. G. Sul</option>
  </optgroup>
  <optgroup label="Sudeste">
    <option value="SP">São Paulo</option>
    <option value="MG">Minas Gerais</option>
  </optgroup>
</select>
```

Figura: Um grupo de opções

Outline

- 1 Introdução
- 2 Criando um formulário
 - Formulários
- 3 Elementos para entrada de dados do usuário
 - Elemento input
 - Listas
 - Elemento de texto com múltiplas linhas
- 4 Recursos extras de formulários
- 5 Outros elementos
- 6 Atividades

Atividade 1

1. Criar um arquivo chamado **form1.html**;
2. Montar um formulário para a reserva de quarto em um hotel com os seguintes elementos:
 - 2.1 Nome completo;
 - 2.2 Endereço de email;
 - 2.3 Confirmar endereço de email;
 - 2.4 Data de chegada;
 - 2.5 Número de *per-noites*;
 - 2.6 Número de hospedes;
 - 2.7 Código promocional;
 - 2.8 Botão “Solicitar reserva”;

Atividade 2

1. Criar um arquivo chamado **form2.html**;
2. Montar um formulário para um usuário poder entrar em contato com o suporte de uma rede de supermercados:
 - 2.1 Nome completo;
 - 2.2 Endereço de email;
 - 2.3 Endereço para correspondencia;
 - 2.4 Cidade;
 - 2.5 UF (lista de opções);
 - 2.6 Setor no qual o usuário deseja entrar em contato (uma lista com as opções devolução de mercadorias, financeiro e entrega);
 - 2.7 Mensagem;
 - 2.8 Receber novidades (o usuário irá marcar se ele deseja receber);
 - 2.9 Botão “Enviar contato”;

Atividade 3

1. Criar um arquivo chamado **form3.html**;
2. Montar um formulário que permita que usuários se cadastrem em uma loja virtual;
 - 2.1 Nome completo;
 - 2.2 CPF (use o atributo *pattern*);
 - 2.3 Data de nascimento;
 - 2.4 Sexo;
 - 2.5 Cidade;
 - 2.6 UF (lista de opções);
 - 2.7 Endereço;
 - 2.8 Cep (use o atributo *pattern*);
 - 2.9 Email;
 - 2.10 Senha (use o atributo *pattern*);
 - 2.11 Receber promoções e novidades (o usuário irá marcar se ele deseja receber);
 - 2.12 Botão “Cadastrar”;

Referências I



HTML5 - A vocabulary and associated APIs for HTML and XHTML.
W3C.

Disponível em <http://www.w3.org/TR/html5/forms.html>



Empty attribute syntax.
W3C - *HTML5 Markups*.

Disponível em <http://dev.w3.org/html5/marking/syntax.html#syntax-attr-empty>



Regular-Expressions.info - Regex Tutorial, Examples and Reference -
Regex Patterns.

Disponível em <http://www.regular-penalty.com/hskip/zskip/discretionary{-}{ }{\ }\\penalty\@M\hskip\z@skipexpressions.info/>