

Você está em

**Guia de HTML » Praticando**

## 12. Criando Formulários HTML



Voltar para guia



Marcar como concluído



Tire sua dúvida

Documentos HTML podem ser interativos e a principal forma de interação entre um usuário e um site ou sistema web é através de formulários.

04:45 min

[Ir para o código](#)

## Documentação

A tag `<form>` indica ao HTML que ali começa um formulário:



2053



Você está em

Guia de HTML » Praticando

## Tag <input />

Nosso formulário precisa de caixas de texto para interação do usuário e para este fim, utilizamos a tag `<input />`.

O padrão desta tag é de texto comum, mas sua forma e comportamento podem variar se utilizarmos o atributo `type`.

## Atributo type

O atributo `type` possui muitas variações disponíveis. Com o HTML5, este controle ganhou muitas outras funcionalidades. Os valores possíveis para `type` são: `text`, `password`, `submit`, `reset`, `radio`, `checkbox`, `file`, `image`, `color`, `date`, `number`, `range` e `step`, `email`, `tel`, `time`, `url`.

Vamos nos concentrar nos tipos que importam para nós no formulário de login: `text`, `password` e `submit`.

## type="text"

Obviamente, o tipo `text` define um input do tipo texto. Este tipo é ideal para nosso campo login:

```
1 | <input type="text" name="login" />
```

Apesar de `text` ser o padrão da tag `<input />`, evite definir um input sem `type` definido.

## type="password"

Este tipo nos provê um input adequado para senhas. Quando definimos um `<input />` com `type="password"` obtemos a renderização de um input em que seu conteúdo fica ofuscado por algum símbolo:

```
1 | <input type="password" name="senha" />
```

Veja como fica a renderização na **Figura 1**.

Login:  Senha:

**Figura 1.** O campo senha possui o atributo `type="password"`. Isso ofusca seu conteúdo.

## type="submit"

Quando definimos um `<input />` do tipo `submit`, este comporta como um **botão**. A ação deste botão será a de



2053



Você está em

Guia de HTML » Praticando



## Atributo value

O atributo `value` possui funções e comportamento diferente em cada caso em que é utilizado. Como vimos anteriormente, em um `<input />` do tipo `submit`, o `value` serve para definir um texto para o botão.

Para os demais `<input />`, assim como em opções de uma caixa de seleção `<select>`, o atributo `value` define o valor inicial do elemento. Veja:

```
1 | <label>Nome</label>
2 | <input type="text" name="nome" value="Insira o nome" />
```

A renderização do código é apresentada na **Figura 2**.

Nome

**Figura 2.** Definindo via código um conteúdo para o atributo `value` de um `<input/>` já o deixa preenchido.

Observe o uso do atributo `value` em um `input` do tipo `submit`:

```
1 | <input type="submit" value="Entrar" />
```

Você está em

Guia de HTML » Praticando

Aqui, o atributo `value` nos permite definir um texto para o botão de submissão do formulário. Porém, sua função vai muito, além disso.

## Atributo Placeholder

O atributo `value` nos ajuda a preencher campos de formulário por meio de programação. Isso é interessante quando vamos modificar um cadastro feito em um site, por exemplo, onde é possível editar o conteúdo que já vem escrito.

Porém, quando estamos fazendo um novo cadastro e precisamos, de algum modo, facilitar ao usuário o entendimento sobre o preenchimento de um campo, utilizar o `value` não é uma boa prática. A situação apresentada na **Figura 2** não está correta, do ponto de vista semântico do HTML.

O `<input />` possui um atributo para isso. O valor colocado em `placeholder` é utilizado para criar um texto de exemplo que oriente o usuário sobre como ele pode preencher o campo:

```
1 <label>Nome: </label>
2 <input type="text" name="nome" placeholder="Informe seu nome" />
```

Veja o exemplo a seguir na **Figura 4**.

Nome:

**Figura 4.** Usando o atributo `placeholder` faz com que o conteúdo do campo seja exibido de com um visual mais claro além de sumir quando o campo recebe o foco do cursor.

Note que, na renderização padrão do navegador, o texto dentro do `input` fica mais claro, indicando que é um conteúdo provisório.

## Outros controles

E quando precisamos de um cadastro complexo, como o cadastro de um cliente em um banco de dados? O HTML nos fornece outros controles para formulários de todos os tipos.

### `<textarea>`

Textos maiores, com múltiplas linhas não podem ser colocados em `inputs` comuns de texto. Para isto utilizamos a

Você está em

**Guia de HTML » Praticando**veja como o `textarea` é renderizado na **Figura 5**.

Mensagem

**Figura 5.** Renderização de um campo `<textarea>`.

## `<select>`

O controle `<select>` é utilizado para criar caixas de seleção e suas opções são representadas por elementos `<option>`. Observe o exemplo:

```
1 <select>
2   <option value="RJ">Rio de Janeiro</option>
3   <option value="SP">São Paulo</option>
4   <option value="MG">Minas Gerais</option>
5   <option value="ES">Espírito Santo</option>
6 </select>
```

Veja o resultado na **Figura 6**.

Estado:

**Figura 6.** Renderização de um campo `<select>`.Observe a **Figura 7** e veja como fica se clicarmos.

Estado: 

Rio de Janeiro

Rio de Janeiro

São Paulo

Minas Gerais

Espírito Santo

**Figura 7.** Como o `<select>` é exibido quando se clica

Note que o atributo `value` é utilizado para definir o valor da `option`, mas este valor não é mostrado dentro da caixa de

Você está em

Guia de HTML » Praticando

## Atributo selected

Utilizando o atributo `selected` podemos dizer ao navegador que um determinado `<option>` deve vir selecionado por padrão:

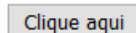
```
1 | <option value="SP" selected>São Paulo</option>
```

## <button>

O controle `<button>` cria um botão clicável sem uma ação pré-definida:

```
1 | <button type="button">Clique aqui</button>
```

Veja como fica na **Figura 8**.



**Figura 8.** Um elemento `<button>` é renderizado como um `<input>` do tipo submit.

É uma boa prática definir sempre um valor para o atributo **type**. Navegadores diferentes possuem formas diferentes de definir um valor padrão para um elemento `<button>`.

## Outros tipos de input

Como vimos anteriormente, a tag `<input />` possui muitos tipos além de **text**, **password** e **submit**. Vamos analisar mais a fundo alguns de seus principais tipos.

### checkbox

**Checkbox** é uma caixa de marcação, muito utilizada para selecionarmos múltiplas opções em um formulário:

```
1 | <label>Checkbox 1</label>
2 | <input type="checkbox" name="check1" value="1" />
3 |
4 | <label>Checkbox 2</label>
5 | <input type="checkbox" name="check2" value="2" />
6 |
7 | <label>Checkbox 3</label>
8 | <input type="checkbox" name="check3" value="3" />
```



2053



Você está em

Guia de HTML » Praticando

## Radio

O controle `radio` funciona de forma similar ao `checkbox`, mas permite apenas a seleção de um único item entre as opções disponíveis. Veja:

```
1 <label>Radio 1</label>
2 <input type="radio" name="radio" value="1" />
3
4 <label>Radio 2</label>
5 <input type="radio" name="radio" value="2" />
6
7 <label>Radio 3</label>
8 <input type="radio" name="radio" value="3" />
```

Observe o resultado na **Figura 10**.

Radio 1 ☐ Radio 2 ☐ Radio 3 ☐

**Figura 10.** `<input>` do tipo `radio` renderizados no navegador

## Acessibilidade com inputs checkbox e radio


Interagir com checkboxes pode ser difícil, pois devemos acertar o mouse bem em cima do quadradinho que ele nos fornece. Nesse sentido, podemos incluir um pouco de acessibilidade a estes elementos utilizando o atributo `for` da tag `<label>` em seu rótulo juntamente com o atributo `id` do input. Confira:


```
1 <label for="radio1">Radio 1</label>
2 <input type="radio" id="radio1" name="radio" />
```

Dessa forma, ao clicar no label do radio ou checkbox, automaticamente a caixinha é marcada.

## Guia

- Guia HTML

 Ver código

 Marcar como concluído

Próxima aula →

Você está em

**Guia de HTML » Praticando**

## Suporte ao aluno - Tire a sua dúvida.



Poste aqui a sua dúvida, nessa seção só você e o consultor podem ver os seus comentários.

[Enviar dúvida](#)

Filtrar comentários por:

Curso

