

formulários

Desenvolvimento WEB I

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
Prof. Felipe Scheidt – IFPR – Campus Foz do Iguaçu
2023

Formulários

- Em páginas web é comum que sejam solicitadas informações do usuário.
- Exemplos: pesquisa de opinião, fazer login, fazer cadastro, ...
- Formulário é um elemento do HTML que permite a entrada de dados e seu envio para o servidor de um site.

<form>

- Um formulário é definido pela tag **<form>**
- O escopo da tag <form> </form> delimita o início e o fim do formulário, ou seja, todas as entradas de dados devem estar contidas dentro da tag form.
- Para enviar os dados do formulário, deve existir um botão que dispara o envio.

`<form>` html

`<form>`

Email: `<input type="text" name="email">`

`<button>enviar</button>`

`</form>`

Email:

<form> - atributos

A tag form possui dois atributos que precisam ser definidos para envio correto dos dados:

- **action**: define o *destino* para onde os dados serão enviados
- **method**: define o método de envio [get/post]

```
<form action="/endpoint" method="POST">
```

```
<!-- inputs do formulario -->
```

```
<button>enviar</button>
```

```
</form>
```

<input>

- É uma tag que cria um "campo", onde é permitido ao usuário entrar com uma informação.
- Uma input pode ter diferentes **tipos** (texto, número, data,...)
- O tipo é definido pelo atributo type.

<input> - atributos

```
<form>
```

```
  <label>Nome</label>
```

```
  <input type="text" name="nome">
```

```
  <label>Peso</label>
```

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

```
</form>
```

Nome

Peso

Há dois atributos importantes que precisam ser definidos quando usamos a tag input em um formulário: type e name

type define o tipo (ex.: text, number)

name atribui um nome para a input (ex.: nome, peso)

Além disso é comum adicionarmos um label para apresentar ao usuário um rótulo sobre a entrada esperada. A tag **label** acaba sendo usada com frequência junto com a input em formulários.

<input> - tipo

Exemplo de tipos configurado no atributo type

- **text**: aceita caracteres alfanuméricos
- **radio**: é um tipo de botão, onde apenas uma opção pode ser selecionada
- **checkbox**: permite selecionar várias opções.
- **submit**: um botão para envio do formulário
- **password**: um campo cria uma máscara nos caracteres digitados.
- **button**: cria um botão.
- **date**: define um campo que aceita apenas datas.

Form duas colunas (css)

Duas colunas é um padrão de layout comum quando se trabalha com formulários. Esse layout pode ser criado no CSS através da propriedade **display grid** e da **grid-template-columns** que especifica o número de colunas. Exemplo:

```
.grid {  
  display: grid;  
  grid-template-columns: max-content max-content;  
  align-items: center;  
  gap: 10px;  
}  
  
.grid label{  
  justify-self: right;  
}
```

Form duas colunas (html)

Para aplicar o grid de duas colunas, basta definir o atributo class do form:

```
<form class="grid">  
  <label>Nome</label>  
  <input type="text" name="nome">  
  <label>CPF</label>  
  <input type="text" name="cpf">  
  <label>Endereço</label>  
  <input type="text" name="endereco">  
</form>
```

Nome

CPF

Endereço

Arquivo CSS externo

Para criar um link para um arquivo css externo, usamos a tag `<link>` e o atributo `href` (*caminho para o arquivo*)

```
<head>  
  <meta charset="UTF-8">  
  <link href="filename.css" rel="stylesheet">  
  <title>Page title</title>  
</head>
```



Maneiras vincular CSS ao HTML

3º – Externo

url

```
<link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap/dist/css/bootstrap.min.css" >
```

local

```
<link href="filename.css" rel="stylesheet">
```

2º – Na head

```
<head>
  <meta charset="UTF-8">
  <style>
    main {
      width: 600px;
      margin: auto;
    }
  </style>
</head>
```

1º – Inline

```
<div style="color: ■ red;font-family: 'Hack';">
  <h3>jungle</h3>
  <img src='../imgs/jungle01.jpg' />
</div>
```

Validação de tamanho

maxlength: quantidade máxima de caracteres aceitos na input

minlength: quantidade mínima de caracteres.

```
<form class="grid">
  <label>Cidade</label>
  <input type="text" name="nome" minlength="3">
  <label>Estado</label>
  <input type="text" name="estado" maxlength="2">
</form>
```

Cidade	<input type="text" value="Foz do Iguaçu"/>
Estado	<input type="text" value="PR"/>

radio

Radio limita a seleção de apenas uma opção dentre várias. O **name** das inputs devem ser iguais.

```
<form class="grid1">
  <h2>Qualidade do serviço?</h2>
  <label>
    <input type="radio" name="pesquisa" value="3">
    Excelente
  </label>
  <label>
    <input type="radio" name="pesquisa" value="2">
    Bom
  </label>
  <label>
    <input type="radio" name="pesquisa" value="1">
    Péssimo
  </label>
</form>
```

Qualidade do serviço?

- ☐ Excelente
- ☐ Bom
- ☐ Péssimo

checkbox

Permite selecionar várias opções

```
<form>
  <h2>Tipos de séries favoritas</h2>
  <label>
    <input type="checkbox" name="favoritos" value="1">
    Aventura
  </label>
  <label>
    <input type="checkbox" name="favoritos" value="2">
    Drama
  </label>
  <label>
    <input type="checkbox" name="favoritos" value="3">
    Ficção científica
  </label>
</form>
```

Tipos de séries favoritas

- ☒ Aventura
- ☐ Drama
- ☒ Ficção científica

<select>

A select é semelhante ao input radio, ou seja, permite a seleção de apenas uma opção. É mais prática de trabalhar quando se tem uma lista com muitas opções (lista de cidades, estados, ...).

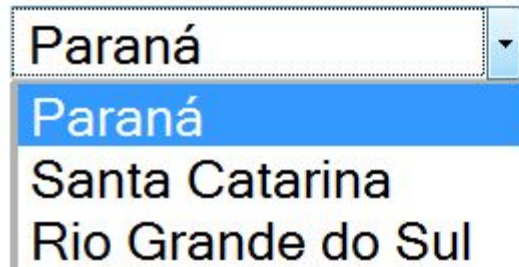
normal

Selecione o estado onde mora?

A normal select dropdown menu with a light blue border and a small downward arrow on the right. The text 'Paraná' is displayed inside the box.

clicado

Selecione o estado onde mora?

A select dropdown menu with the list of options open. The menu has a light blue border. The text 'Paraná' is displayed inside the box. Below the box, a list of options is shown: 'Paraná', 'Santa Catarina', and 'Rio Grande do Sul'. The 'Paraná' option is highlighted with a blue background.

<select>

Exemplo do select em um formulário

Em qual estado você mora?

```
<form>
```

```
<select name="estados">
```

```
  <option value="AM">AMAZONAS</option>
```

```
  <option value="PR">Paraná</option>
```

```
  <option value="SC">Santa Catarina</option>
```

```
  <option value="SP">São Paulo</option>
```

```
</select>
```

```
</form>
```

<select>

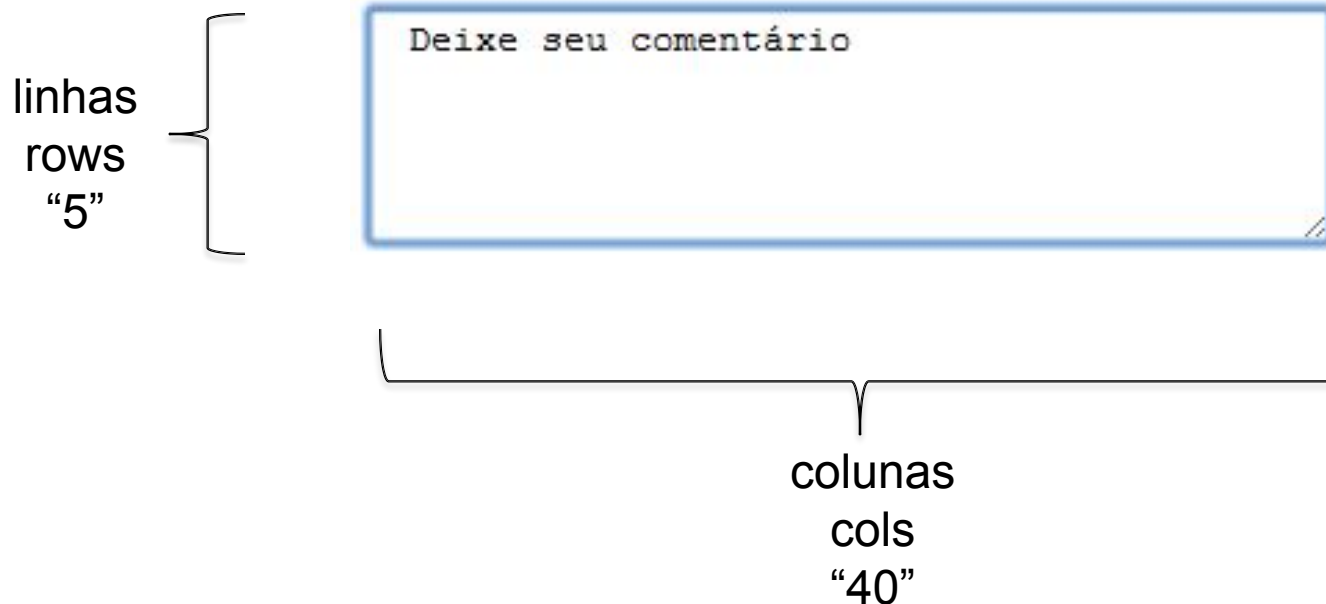
- O atributo **selected** permite configurar que uma opção seja selecionada por padrão. No exemplo abaixo, Santa Catarina será a opção padrão exibida quando a página for carregada.

```
<select name="estados">
  <option value="AM">AMAZONAS</option>
  <option value="PR">Paraná</option>
  <option selected value="SC">Santa Catarina</option>
  <option value="SP">São Paulo</option>
</select>
```

<textarea>

- textarea permite a entrada de uma quantidade maior de texto, por exemplo, em campos que solicitam comentários, motivo, feedback, etc...
- Para controlar a quantidade de linhas e largura da textarea, definimos os seguintes atributos:
 - **rows**: a quantidade de linhas da textarea
 - **cols**: a quantidade de colunas

<textarea>



```
<textarea cols="40" rows="5" name="comentario">  
    Deixe seu comentário  
</textarea>
```

+ tipos de inputs

`<input type="email">`

input que aceita somente emails

`<input type="number">`

define uma input do tipo numérico

`<input type="range">`

define uma input do tipo intervalo

`<input type="date">`

define uma input tipo data (calendário)

`<input type="time">`

uma input para trabalhar com horário

`<input type="datetime-local">`

input que aceita data e horário

+ atributos da input

ATRIBUTO	VALORES	FUNÇÃO
autocomplete	on, off	define se o campo aceita autocomplete
autofocus		define que a input receberá o foco quando a página terminar de carregar
disabled	disabled	define que a tag está desabilitada
placeholder	texto	define uma dica sobre os valores que o campo aceita
readonly	readonly	campo somente leitura
required	required	define um campo obrigatório
min	número ou data	valor mínimo que o input aceita
max	número ou data	valor máximo que o input aceita

Formatar validação da input

Exemplo de 4 regras CSS para formatar o estilo da input no de uma entrada inválida ou válida.

```
/* regra para valores inválidos*/  
input:required:invalid {  
}  
/* regra para valores válidos */  
input:required:valid {  
}  
/* regra para valores inválidos  
quando input recebe o foco*/  
input:focus:invalid {  
}  
/* regra para valores validos  
quando input recebe o foco*/  
input:focus:valid {  
}
```

Exercício I

Usando os conceitos vistos em aula, faça o código HTML e CSS do formulário abaixo usando grid Layout.

Data de nascimento:	<input type="text" value="dd/mm/aaaa"/>
Escolaridade:	<input type="text" value="Ensino Fundamental ▼"/>
Situação da Escolaridade:	<input type="radio"/> Completo <input type="radio"/> Incompleto
Nacionalidade:	<input type="text" value="Brasil ▼"/>
UF onde nasceu:	<input type="text" value="Paraná ▼"/>
<input type="button" value="enviar"/>	

Exercício II

Nome

Email

Password

Possui conhecimento?

☐

JavaScript

☐

HTML

☐

CSS

Curso



Turno

☐

Tarde

☐

Noite

Motivo matrícula

Opções em Curso:

- Análise e Desenvolvimento
- Física
- Matemática

Exercício II (com validação)

Nome	<input type="text" value="La"/>
Email	<input type="text" value="user@example.org"/>
Password	<input type="password" value="..."/>

Continuando o exercício II, adicione validação no formulário. Os campos abaixo devem ser **obrigatórios**:

- Nome
- Email
- Password

Nome possui no mínimo 3 caracteres.

Password deve ter no mínimo 6 caracteres e no máximo 10 caracteres.

Adicione uma regra css que define a cor da fonte para vermelho quando a entrada digitada é inválida

Exercício III

Cadastro usuário

Nome

Email

Nascimento



Pais



Idade

DDD

Telefone