



SHIFT

 FIAP



00000



FRONT-END JOURNEY

DESENVOLVIMENTO WEB COM ANGULAR & REACT

FRONT: **PROGRAMAÇÃO**

HTML - CSS - BOOTSTRAP - SASS





ISRAEL MARQUES JÚNIOR

PROFESSOR

- Israel é pós-graduado em Engenharia Web e trabalha com educação há 27 anos. Trabalhou no desenvolvimento de sistemas para desktop e, em seguida, migrou para a criação de aplicações para a Internet.
- Na FIAP, é professor nos cursos de: Sistemas de Informação, Análise e Desenvolvimento de Sistemas, Sistemas para Internet e Jogos Digitais para as disciplinas focadas em front end.

✉ profisrael.copi@fiap.com.br



AGENDA

1

AULA 1

CONTEÚDO – Baixando o Editor + Introdução HTML + CSS

2

AULA 2

CONTEÚDO – Listas + Links + Imagens + Divs + Posicionamento

3

AULA 3

CONTEÚDO – Semântica – Background – Flexbox

4

AULA 4

CONTEÚDO – Tabelas – Formulários

5

AULA 5

CONTEÚDO – Design Responsivo + Media Queries + Mobile First

AGENDA

6

AULA 6

CONTEÚDO – BootStrap: Introdução + Containers + Utilitários

7

AULA 7

CONTEÚDO – BootStrap: Grid + NavBar + Cards + Carousel

8

AULA 8

CONTEÚDO – BootStrap: Accordion + Modal + Formulários

9

AULA 9

CONTEÚDO – Sass: Introdução + Instalação + Conceitos Iniciais

10

AULA 10

CONTEÚDO – Sass: Estilização + Variáveis + Funcionalidades

AULA 4

TABELAS - **FORMULÁRIOS**

TABELAS

TABELAS

- Criar tabelas com HTML é uma tarefa muito fácil, pois existem tags para todos os elementos que a compõem: títulos, linhas e colunas.
- Tabelas são usadas para exibir dados de uma forma organizada e fácil para o usuário entender o que eles representam.
- A representação dos dados deve seguir uma organização que torne as informações bem claras para o usuário.

<table>

- Para criarmos nossas tabelas em HTML, usamos a tag: **<table></table>**.
- Todo o conteúdo que formará a tabela deverá ficar dentro dessa tag.
- Esse conteúdo deverá estar dentro das linhas e de suas respectivas colunas.

<table>

Conteúdo da tabela...

Conteúdo da tabela...

Conteúdo da tabela...

</table>

<caption>

- A tag **<caption></caption>** será usada para inserir o título da tabela.
- Esse título poderá ser utilizado para identificar o conjunto de dados que a tabela representa.

<table>

<caption> Título da tabela </caption>

Conteúdo da tabela...

Conteúdo da tabela...

</table>

<tr>

- A tag **<tr></tr>** será usada para a criação das linhas existentes na tabela.
- As colunas da tabela, com os seus respectivos conteúdos, serão inseridas dentro das linhas.

<table>

<caption> Título da tabela </caption>

<tr> Linha da tabela...</tr>

<tr> Linha da tabela...</tr>

<tr> Linha da tabela...</tr>

</table>

<td>

- A tag **<td></td>** será usada para a criação das colunas existentes nas linhas da tabela.
- Os dados deverão ficar armazenados dentro da tag **<td>**.

<table>

<caption> Título da tabela </caption>

<tr>

<td> Conteúdo da coluna </td>

<td> Conteúdo da coluna </td>

</tr>

</table>

<th>

- A primeira linha de dados da tabela é considerada uma das mais importantes, porque é nela que identificamos o conjunto de informações que está armazenado em cada coluna.
- Dessa forma, devemos substituir a tag **<td>** pela tag **<th>**: a diferença visual é que isso deixará o nome da coluna centralizado e em negrito.

```
<table>
```

```
<caption> Título da tabela </caption>
```

```
<tr>
```

```
<th> Nome da coluna </th>
```

```
<th> Nome da coluna </th>
```

```
</tr>
```

```
</table>
```


<thead>

- A tag **<thead></thead>** é usada para identificar a primeira linha da tabela.
- Em tabelas muito grandes, e que forem impressas, essa linha sempre será repetida no início de uma nova página impressa, facilitando, assim, a identificação dos dados ali exibidos.

<table>

<caption> Título da tabela </caption>

<thead>

<tr>

<th> Nome da coluna </th>

<th> Nome da coluna </th>

</tr>

</thead>

</table>

<tbody>

- A tag **<tbody></tbody>** é usada para armazenar todas as linhas de conteúdo existentes na tabela, representando, assim, o corpo da tabela.
- A ideia é organizar os dados agrupando-os em uma área específica da tabela.

<table>

<tbody>

<tr>

<td> Conteúdo da coluna </td>

<td> Conteúdo da coluna </td>

</tr>

</tbody>

</table>

<tfoot>

- A tag `<tfoot></tfoot>` é usada para representar a linha de rodapé da sua tabela.
- Na maioria das vezes, usamos o `<tfoot>` como uma linha para alguma observação sobre os dados: um comentário, uma marca de rodapé, algo nesse sentido.
- Seu uso é opcional, e você encontrará tabelas sem esse elemento.

```
<table>
```

```
<tfoot>
```

```
<tr>
```

```
<td> Nota de rodapé </td>
```

```
</tr>
```

```
</tfoot>
```

```
</table>
```

FORMULÁRIOS

FORMULÁRIOS

Um formulário é uma página HTML com uma série de componentes (campos) que podem ser preenchidos ou selecionados pelo usuário e, ao final, ele poderá clicar em um botão que enviará as informações digitadas para um servidor web.

É preciso ressaltar que sozinho o HTML não conseguirá fazer esse envio. Ele precisa da ajuda de uma linguagem de programação para conseguir realizar essa tarefa. Lembre-se: o HTML é apenas uma linguagem de marcação.

<form>

- Para criarmos um formulário HTML, usaremos a tag **<form></form>**, e será em seu interior onde definiremos os componentes que deverão ser vistos e utilizados pelos usuários da nossa aplicação.
- A tag <form> possui alguns atributos que identificam o formulário e controlam como as informações digitadas serão enviadas.

<form>

Campos do formulário

</form>

ATRIBUTO NAME

Define um nome a ser atribuído ao formulário. Os campos do formulário também podem receber esse atributo, visto que algumas linguagens de programação farão uso dele para recuperar o que foi selecionado ou digitado pelo usuário.

ATRIBUTO ID

Define um identificador para o formulário, lembrando que esse valor não poderá ser repetido na página. Os campos do formulário também podem receber esse atributo, já que algumas linguagens de programação farão uso dele para recuperar o que foi selecionado ou digitado pelo usuário.

ATRIBUTO ACTION

- Define o que acontecerá com os dados digitados quando o usuário pressionar o botão de envio, ou seja, submeter o formulário. Basicamente eles poderão ser enviados para uma nova página, um e-mail, uma tabela em algum banco de dados ou algum servidor web, onde serão processados. Isso tudo depende do tipo de aplicação ao qual o formulário pertence.
- É comum inserir como valor do atributo ****action**** o endereço de algum script que irá recuperar as informações digitadas e efetuar o seu respectivo envio.
- **Lembre-se:** mesmo que você deixe o valor desse atributo em branco, o browser tentará submeter o formulário. Como ele não saberá o que fazer, seguirá o fluxo normal, atualizando a página e, dessa forma, os campos preenchidos terão seus valores apagados.

ATRIBUTO METHOD

Esse atributo define qual tipo de método será usado para enviar os dados digitados.

Possuímos o método **GET** e o método **POST**.

- O método GET é o padrão, ele utiliza a URL: local onde você digita o endereço de um site no browser para enviar as informações digitadas ao servidor. É usado para pesquisas ou para passar as informações de um produto, paginação etc.
- O método POST envia as informações incluídas (encapsuladas) no corpo da mensagem, evitando, assim, a sua visualização. É usado para formulários de contato, formulários de login, envio de arquivos etc.

TAG FORM COMPLETA

```
<form name="nome" id="identificador" method="post" action="endereço do script">
```

Campos do formulário

```
</form>
```

<fieldset> + <legend>

A tag **<fieldset>** pode ser usada para hierarquizar um conjunto de dados que tenham algum tipo de relação entre si. Insere uma borda envolvendo todos os seus elementos descendentes.

O uso do **<fieldset>** não é obrigatório, use apenas se quiser agrupar informações em seu formulário.

A tag **<legend>** poderá ser usada como um identificador para o grupo de campos.

```
<fieldset>
```

```
    <legend>Dados Pessoais</legend>
```

```
    <p>Campos do formulário</p>
```

```
</fieldset>
```

<label>

Insere um identificador para um respectivo campo do formulário, podendo ser uma palavra ou frase.

Possui o atributo `for`, que permitirá a criação de um vínculo entre o identificador e o campo desejado. Se o usuário clicar neste identificador, o campo linkado ganhará foco automaticamente, melhorando a sua usabilidade.

Para isso acontecer, será necessário que o atributo **for** possua o mesmo valor que será atribuído ao **id** do campo.

```
<label for="nome">Nome do usuário:  
</label>
```

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

<input>

- Essa tag é responsável pela inserção da maioria dos componentes do formulário. Ela possui vários atributos, entre eles o **type**, que receberá como valor o tipo de campo que desejamos inserir no formulário.

```
<input type="tipo do campo" id="nome">
```


<input type= “text”>

- Insere um campo de texto para digitação dos dados pelo usuário. Nesse campo ele poderá digitar letras, números e caracteres especiais.
- É um dos componentes mais comuns nos formulários.

```
<input type=“text” id=“nome”>
```

<input type= “password”>

- Insere um campo para digitação de uma senha pelo usuário. Ele poderá digitar letras, números e caracteres especiais.
- Toda informação digitada será mascarada e os caracteres não serão exibidos.

```
<input type=“password” id=“senha”>
```

<input type="date">

- Insere um campo para digitação ou seleção de datas. Assim que ele for clicado, aparecerá um calendário, e o usuário poderá fazer a escolha da data desejada.
- Os navegadores possuem padrões diferentes para renderização desse componente. Seria muito interessante você testá-lo em diferentes browsers.

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

<input type="time">

- Insere um campo para digitação ou seleção da hora. Assim que ele for clicado, aparecerá uma caixa para o usuário selecionar a hora e os minutos.
- Os navegadores possuem padrões diferentes para renderização desse componente. Seria muito interessante você testá-lo em diferentes browsers.

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

<input type="datetime-local">

- Insere um campo para digitação ou seleção da data e da hora, ou seja, os dois campos anteriores configurados em apenas um.
- Os navegadores possuem padrões diferentes para renderização desse componente. Seria muito interessante você testá-lo em diferentes browsers.

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

<input type="month">

- Insere um campo para digitação ou seleção de um mês e ano.
- Os navegadores possuem padrões diferentes para renderização desse componente. Seria muito interessante você testá-lo em diferentes browsers.

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

<input type="week">

- Insere um campo para digitação ou seleção de uma determinada semana do ano.
- Os navegadores possuem padrões diferentes para renderização desse componente. Seria muito interessante você testá-lo em diferentes browsers.

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


<input type="email">

- Insere um campo para digitação de um endereço de e-mail. Basicamente é um campo do tipo text com algumas vantagens:
- Ele tem validação própria: não permite que o formulário seja enviado caso o conteúdo digitado nesse campo não seja um endereço de e-mail válido.
- Os smartphones reconhecem esse campo e exibem o teclado já com a opção **.com** e o símbolo de **arroba(@)**.

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

<input type="number">

- Insere um campo para digitação de um número. Na parte final do campo, você encontrará setas para aumentar ou diminuir o valor digitado.
- Os navegadores possuem padrões diferentes para renderização desse componente. Seria muito interessante você testá-lo em diferentes browsers.

```
<input type="number" id="numero">
```

<input type="range">

- Insere uma barra deslizante para receber um valor de um intervalo previamente determinado.
- O intervalo padrão é de 0 a 100, podendo ser alterado conforme suas necessidades.
- Os navegadores possuem padrões diferentes para renderização desse componente. Seria muito interessante você testá-lo em diferentes browsers.

```
<input type="range" id="escala">
```

<input type="color">

- Insere um campo que permitirá ao usuário a seleção de uma cor em uma paleta de cores.
- Com a utilização de alguma linguagem de programação, poderemos utilizar a cor selecionada em qualquer parte da página.
- Um exemplo comum seria estilizar algum elemento ou container com a cor escolhida.

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

<input type="radio">

- Insere campos de seleção que permitirá ao usuário a escolha de uma única opção entre as apresentadas.
- O usuário poderá mudar a seleção efetuada, mas, mesmo assim, ficará apenas com uma opção selecionada.
- Para esse componente funcionar de forma correta, deveremos usar o atributo name contendo o mesmo valor para todos os elementos inseridos.

```
<input type="radio" name="radio1"  
value="1">
```

```
<label for="first">Valor 1</label>
```

```
<input type="radio" name="radio1"  
value="2">
```

```
<label for="second">Valor 2</label>
```

```
<input type="radio" name="radio1"  
value="3">
```

```
<label for="master">Valor 3</label>
```

<input type="checkbox">

- Insere campos de seleção que permitirá ao usuário a escolha de quantas opções ele quiser.
- Ele poderá também alterar as opções selecionadas quantas vezes forem necessárias.

```
<input type="checkbox" name="opcao1" value="1">
```

```
<label for="first">Valor 1</label>
```

```
<input type="checkbox" name="opcao2" value="2">
```

```
<label for="second">Valor 2</label>
```

```
<input type="checkbox" name="opcao3" value="3">
```

```
<label for="master">Valor 3</label>
```

<input type="file">

- Insere um campo que permite o envio de arquivos.
- É composto por um botão e uma área onde o usuário pode selecionar o arquivo que deseja enviar.
- O envio do arquivo só será concretizado com a utilização de alguma linguagem de programação.

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

<input type="submit">

- Permite a inserção de um botão que será usado para enviar os dados digitados pelo usuário.
- Assim que esse botão for pressionado, o browser procurará o atributo **action** inserido na tag **<form>** e verificará para onde as informações deverão ser enviadas.
- O botão inserido usará o atributo **value** como rótulo, o texto que estará visível ao usuário.

```
<input type="submit" id="enviar"  
      value="Enviar dados">
```


<input type="reset">

- Permite a inserção de um botão que será usado para apagar todos os dados digitados no formulário.
- Ele também usará o atributo **value** como rótulo para o botão.
- Cuidado! Uma vez que os dados forem apagados não poderão ser recuperados.

```
<input type="reset" id="enviar"  
value="Limpar dados">
```

OUTROS ATRIBUTOS PARA CAMPOS

O atributo **type** não é o único existente para os campos do formulário. Podemos fazer uso de uma gama diversificada de outras opções que podem diminuir a quantidade de código e melhorar a experiência do usuário.

REQUIRED + VALUE + READONLY + DISABLED

Required: define que o campo deve ter o preenchimento obrigatório.

Value: define o valor inicial que será atribuído ao campo. Algumas linguagens de programação farão uso desse atributo para entender o que foi selecionado ou digitado pelo usuário.

Readonly: define que o valor do campo seja apenas de leitura, não permitindo a alteração por parte do usuário.

Disabled: define que campo fique desabilitado, não permitindo a inserção de dados.

AUTOFOCUS + MINLENGTH + MAXLENGTH + PLACEHOLDER

Autofocus: define que algum campo ganhe foco.

Minlength: define o tamanho mínimo de caracteres permitidos no campo.

Maxlength: define o tamanho máximo de caracteres permitidos no campo.

Placeholder: especifica uma palavra ou frase que, na maioria das vezes, indicará ao usuário como preencher um determinado campo. Ela ficará visível até que o usuário digite a primeira letra, quando isso ocorrer, ele desaparecerá e dará lugar ao conteúdo digitado. Caso o conteúdo inserido seja totalmente apagado, a indicação voltará.

<select> + <option>

A tag **<select>** **</select>** cria um campo de seleção onde o usuário poderá selecionar uma das opções que serão listadas.

As opções serão criadas pela tag **<option>** **</option>** e cada uma delas deverá possuir o seu respectivo atributo value.

```
<select id="atuacao" required>  
  <option value="">Selecione</option>  
  <option value="1">Front End</option>  
  <option value="2">Back End</option>  
  <option value="3">Full Stack</option>  
</select>
```

<datalist> + <option>

A tag **<datalist>** cria uma lista de opções que poderá ser associada ao elemento `<input>`.

Para isso, devemos atribuir o elemento `id` à tag `<datalist>`, já na tag `<input>`, utilizaremos o atributo **link** recebendo o mesmo valor do **id**.

```
<input list="areas" id="atua"
name="atua">
```

```
<datalist id="areas">
```

```
<option value="Diretoria">
```

```
<option value="Financeiro">
```

```
<option value="Tecnologia">
```

```
<option value="Projetos">
```

```
</datalist>
```

<textarea>

Insere um campo de texto de múltiplas linhas. É muito comum encontrar esse tipo de campo nos formulários, pois os usuários o utilizam para escrever alguma mensagem, fazer algum comentário, uma sugestão etc.

```
<textarea name="mensagem"  
id="mensagem"></textarea>
```

<button>

Insere um botão na página. O tipo de botão a ser inserido será definido no atributo type.

- **Submit:** quando clicado, executa a mesma função do input type="submit".
- **Reset:** quando clicado, executa a mesma função do input type="reset".
- **Button:** quando clicado, pode executar alguma nova função definida por alguma linguagem de programação. O JavaScript adora essa opção.

```
<button type="submit">Enviar dados</button>
```

```
<button type="reset">Limpar dados </button>
```

```
<button type="button">Alguma função</button>
```


BIBLIOGRAFIA BÁSICA

- **DUCKETT**, Jon. HTML&CSS projete e construa websites. 2014.
- **SILVA**, Maurício Samy. HTML5 A linguagem de marcação que revolucionou a Web. 2011.
- **SILVA**, Maurício Samy. CSS3: Desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3. 2012.
- **TERUEL**, Evandro Carlos. HTML5 guia prático. 2011.
- **SANDERS**, Bill. Smashing Html5. 2012.
- **CASTRO**, Elizabeth, HYSLOP, Bruce. HTML5 e CSS3: Guia Prático e Visual. 2013.

OBRIGADO



FIAP

Copyright © 2021 | Professor Israel Marques Cajai Júnior

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.



SHIFT

 FIAP



00000