

PROGRAMAÇÃO WEB

Prof. Ms. Wilson Lourenço

wilson.slourenco@sp.senac.br



FORMULÁRIOS

Formulários a Fundo

- Quando solicitamos que o usuário informe seu nome, seu endereço de email, se ele quer receber uma newsletter, qualquer informação, precisamos utilizar os elementos corretos.
- Para isso, vamos conhecer os formulários HTML: a tag `<form>` .
- Já usamos alguns antes e agora vamos ver a fundo seus desdobramentos.

Atributos do form

```
<form action="/efetivar.html" method="POST">  
</form>
```

- O formulário exemplificado na tag acima apresenta o atributo obrigatório action.
- O valor desse atributo é o endereço para onde as informações do formulário serão enviadas, e esse valor depende inteiramente de como é feita a aplicação que receberá essas informações no lado do servidor.

- O segundo atributo, `method`, especifica o método do HTTP pelo qual essa informação será transmitida.
- O valor `post`, de maneira simplista, significa que queremos inserir as informações desse formulário, salvá-la de alguma maneira.
- Outro valor possível para esse atributo, o `get`, é utilizado quando queremos obter alguma coisa a partir das informações que estamos transmitindo, por exemplo, um formulário de busca.

Componentes

- Porém, neste exemplo, não temos nenhum elemento para capturar as informações.
- Na verdade, somente a marcação da tag `<form>` não mostra nenhum elemento visível no navegador.

- Vamos supor que precisemos de uma informação como o nome do visitante do nosso site para guardar em um banco de dados.
- Vamos, então, adicionar alguns elementos ao nosso formulário anterior:

Por enquanto vamos apenas entender a tag
`<form>`

Não vamos implementar nada no projeto da
Mirror Fashion até a parte dos exercícios

```
<form action="/efetivar.html" method="POST">
```

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

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

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

```
</form>
```


Label

- Adicionamos a marcação do elemento `<label>`
- Esse elemento é uma tag de conteúdo, e seu texto é exibido de maneira comum dentro do nosso formulário, a única diferença é que essa marcação faz uma ligação com outro elemento qualquer em nosso formulário.
- Note que nosso label tem o atributo `for`, que recebe o valor `nome` .

- Quando clicamos com o mouse sobre o texto marcado com a tag label, o elemento que tem o atributo id com o mesmo valor que o atributo for do label é selecionado para que possamos interagir com ele.
- No exemplo, esse elemento vinculado ao label é um campo de texto que declaramos com a tag input.
- Essa marcação <label> é de extrema importância para a usabilidade e acessibilidade dos nossos formulários.

Input

- A maioria dos elementos que utilizamos nos formulários para capturar informações dos usuários são da tag `<input>`.
- No exemplo anterior, utilizamos duas variações dessa tag.
- Os tipos diferentes de inputs são determinados pelo valor do seu atributo `type`. A seguir, vamos detalhar os tipos aceitos para essa tag.

- **text**

- Provavelmente o tipo mais comum de input é o que tem o atributo `type="text"`, que é utilizado quando queremos que o usuário envie uma informação textual simples, pois esse elemento não permite a entrada de quebras de linha.

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

Nome:

- Ao enviarmos o formulário, a informação digitada pelo usuário é acessível no lado do servidor por meio do atributo name, utilizado para identificar cada informação contida nos parâmetros da requisição.
- Para ter acesso à informação digitada quando tratamos o formulário com algum tipo de script, para validar o conteúdo por exemplo, é necessário obter o conteúdo da propriedade value do objeto no DOM.

- **password**

- O input que recebe o atributo `type="password"` é similar ao anterior, do tipo `text`, com a diferença de que ele não exibe exatamente o texto digitado pelo usuário, e sim uma série de símbolos * ou outro, dependendo do navegador e sistema operacional.

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

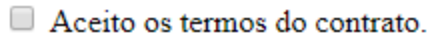
Senha:

- **checkbox**

- O elemento input do tipo checkbox exibe uma caixa para marcação, e é muito utilizado quando temos uma opção que pode ser marcada como sim ou não, por exemplo "Aceito os termos de contrato do usuário", ou "Manter a sessão ativa" em formulários de login.
- Apesar de muito utilizado com o valor true, é possível determinar qualquer valor para o checkbox.

```
<input id="contrato" name="contrato" type="checkbox" value="sim">
```

```
<label for="contrato">Aceito os termos do contrato.</label>
```



- **radio**

- Quando desejamos que o usuário escolha somente uma entre uma série de opções, podemos utilizar elementos input do tipo radio.
- Quando há mais de um elemento desse tipo com o mesmo valor no atributo name, somente um pode ser selecionado.

<p>Selecione a faixa etária:</p>

<input type="radio" name="idade" id="idade5" value="5">

<label for="idade5">Menos de 5 anos</label>

<input type="radio" name="idade" id="idade10" value="10">

<label for="idade10">Menos de 10 anos</label>

<input type="radio" name="idade" id="idade15" value="15">

<label for="idade15">Menos de 15 anos</label>

<input type="radio" name="idade" id="idade20" value="20">

<label for="idade20">Menos de 20 anos</label>

Selecione a faixa etária:

☐ Menos de 5 anos ☐ Menos de 10 anos ☐ Menos de 15 anos ☐ Menos de 20 anos


- **image**

- É possível substituir o botão de envio do formulário por uma imagem, possibilitando criar um visual mais atrativo para o formulário.
- Hoje em dia, com o CSS3, podemos adicionar imagens que não fazem parte do conteúdo, são somente estilo, com Image Replacement no CSS, como fizemos na página index.html.

```
<input type="image" name="botao" src="img_submit.gif" alt="Botão para enviar o formulário" width="20" height="18">
```

Nome:

Sobrenome:



- **file**

- Quando é necessário que o usuário envie um arquivo para a aplicação no lado do servidor é necessário o uso do input do tipo file.
- Para o correto envio dos arquivos, muitas vezes também é necessário adicionar o atributo `enctype="multipart/form-data"` na tag `<form>` .

`<label for="myfile">Selecione o arquivo:</label>

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

`

Selecione o arquivo:

Escolher arquivo

Nenhum arquivo selecionado

- **hidden**

- Muitas vezes precisamos enviar e receber informações que não têm utilidade direta para o usuário e, portanto, não devem ser exibidos no formulário.
- Para essa finalidade, existe o input do tipo hidden, que somente carrega um valor.

```
<input          type="hidden"          name="codigo"  
value="abc012xyz789">
```

- **button**

- O elemento input com o atributo type="button" renderiza um botão dentro do formulário, mas esse botão não tem nenhuma função direta nele e é comumente utilizado para disparar eventos para a execução de scripts.
- O texto do botão é determinado pelo valor do atributo value .

```
<input type="button" name="mostra_dialogo"  
value="Clique aqui!" onclick="msg()">
```

O botão a seguir ativa um comando JavaScript quando clicado

Clique aqui!

Uma página incorporada nesta página diz

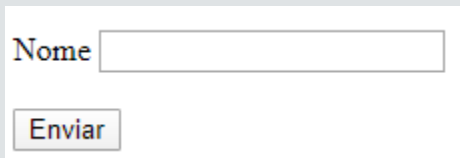
Olá aluno

OK

- **submit**

- O elemento input com o atributo type="submit" é similar ao type="button", mas quando acionado esse elemento inicia a chamada que envia as informações do formulário para o endereço indicado no atributo action do <form>.

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



Nome

- **reset**
 - O input com type="reset" elimina os valores digitados anteriormente nos elementos de um formulário, permitindo que o usuário limpe o mesmo.

```
<input type="reset" name="reset" value="Limpar">
```

- **<INPUT> E <BUTTON>**

- A tag <input> dos tipos button, submit e reset pode ser substituída pela tag <button>. Neste caso, o texto do botão passa a ser indicado como conteúdo da tag. Ainda assim é necessário especificar o valor do atributo type, inclusive se ele for button:

<button type="button" name="mostra_dialogo">Clique aqui!</button>

<button type="submit" name="enviar">Enviar</button>

<button type="reset" name="reset">Limpar</button>

Textarea

- Quando desejamos que o usuário insira uma quantidade grande de informações textuais, incluindo quebras de linha, é necessário o uso da tag textarea

`<textarea name="texto" rows="4" cols="40"> </textarea>`

Qual o seu comentário sobre o produto:

Digite aqui seu comentário!

Select, Optgroup e Option

- Quando desejamos que o usuário selecione entre diversas opções, com a possibilidade de flexibilizar a maneira com que ele interage com o componente do formulário, podemos utilizar a tag `<select>` .

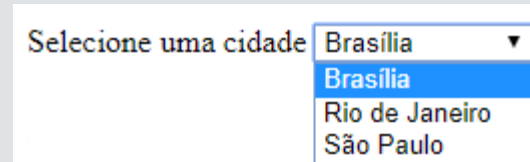
```
<select name="cidades">
```

```
  <option value="bsb">Brasília</option>
```

```
  <option value="rj">Rio de Janeiro</option>
```

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

```
</select>
```



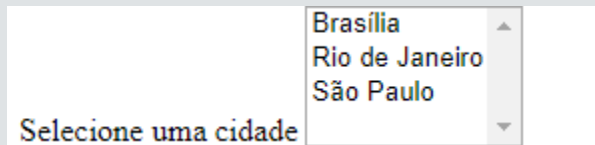
Selecione uma cidade

Brasília	▼
Brasília	
Rio de Janeiro	
São Paulo	

The image shows a web form with a label 'Selecione uma cidade' and a dropdown menu. The dropdown menu is open, showing a list of cities: Brasília, Rio de Janeiro, and São Paulo. The first option, Brasília, is highlighted in blue. The dropdown arrow is visible in the top right corner of the menu box.

- Em sua configuração padrão, o controle select exibe o que conhecemos como menu drop-down, permitindo que somente uma das opções possa ser selecionada.
- Caso seja adicionado o atributo multiple , é possível selecionar mais de uma opção da mesma maneira que selecionamos diversos arquivos no explorador do sistema operacional.

```
<select name="cidades" multiple>  
  <option value="bsb">Brasília</option>  
  <option value="rj">Rio de  
Janeiro</option>  
  <option value="sp">São Paulo</option>  
</select>
```



Selecione uma cidade

- Brasília
- Rio de Janeiro
- São Paulo

- Caso necessário, dependendo do número de opções apresentadas ao usuário, pode ser interessante agrupá-las usando o optgroup:

```
<select name="bairro">
```

```
  <optgroup label="Brasília">
```

```
    <option value="asan_bsb">Asa Norte</option>
```

```
    <option value="asas_bsb">Asa Sul</option>
```

```
  </optgroup>
```

```
  <optgroup label="Rio de Janeiro">
```

```
    <option value="botafogo_rj">Botafogo</option>
```

```
    <option value="centro_rj">Centro</option>
```

```
  </optgroup>
```

```
  <optgroup label="São Paulo">
```

```
    <option value="vlmariana_sp">Vila Mariana</option>
```

```
    <option value="centro_sp">Centro</option>
```

```
  </optgroup>
```

```
</select>
```

Elemento Select

O elemento select é usado para criar uma lista drop-down.

Selecione uma cidade

Asa Norte	▼
Brasília	
Asa Norte	
Asa Sul	
Rio de Janeiro	
Botafogo	
Centro	
São Paulo	
Vila Mariana	
Centro	

Novos Elementos do HTML5

- Com a nova especificação do HTML, é possível utilizar uma série de novos componentes que facilitam bastante o desenvolvimento de formulários.
- Alguns componentes são incompatíveis com os navegadores, mas mostram, na maioria dos casos, um campo de texto permitindo a entrada de qualquer tipo de informação.

- A maioria dos novos tipos de componentes de formulário foram criados para permitir que o navegador adapte o método de entrada para o mais adequado em cada um dos casos.
- Alguns desses componentes já são compatíveis com navegadores de dispositivos móveis.

- **email**

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

- O input do tipo email permite que os dispositivos móveis, principalmente, exibam um teclado adaptado para facilitar esse tipo de entrada.
- Por exemplo, o iPhone exibe um teclado com o caractere @ e com as opções de domínio .com .



- **number**

`<input type="number" max="100" step="5">`

- O input do tipo number, além de exibir um teclado numérico em dispositivos móveis, nos navegadores modernos exibe um controle que permite incrementar ou decrementar o valor do campo clicando em uma seta para cima ou para baixo.
- Além dessa diferença visual, é possível determinar valores mínimos, máximos e se há uma escala de valores válidos.
- No exemplo anterior, o elemento deve aceitar números múltiplos de 5 com o limite do valor "100".

Números de 5 em 5 até 100:

- url

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

- O elemento input com tipo url permite que os dispositivos exibam um teclado com, no exemplo do iPhone, opções como www e .com

- **range**

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

- O elemento input do tipo range exibe um controle deslizante nos navegadores modernos, permitindo uma interação mais agradável quando precisamos de um valor numérico em escala.
- O controle guarda um valor numérico em seu atributo value. Assim como o input do tipo number , é possível especificar um valor mínimo, máximo e uma escala.



- **date, month, week, time, datetime e datetime-local**

```
<input type="date" id="validade" name="validade" value="AAAA-MM-DD">
```

- Os controles de "date picker" são feitos para coletar uma informação de data ou hora. São várias as possibilidades de formato de data ou hora.
- No navegador Opera, quando utilizado esse tipo de controle, o usuário pode selecionar uma data a partir de um calendário. É possível especificar datas mínima e máxima.

Validade:

22/05/2020

maio de 2020

dom	seg	ter	qua	qui	sex	sáb
26	27	28	29	30	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	5	6

No computador

Carrier 5:07 PM

May 18, 1973

`<input type="date">`

Previous Next Clear Done

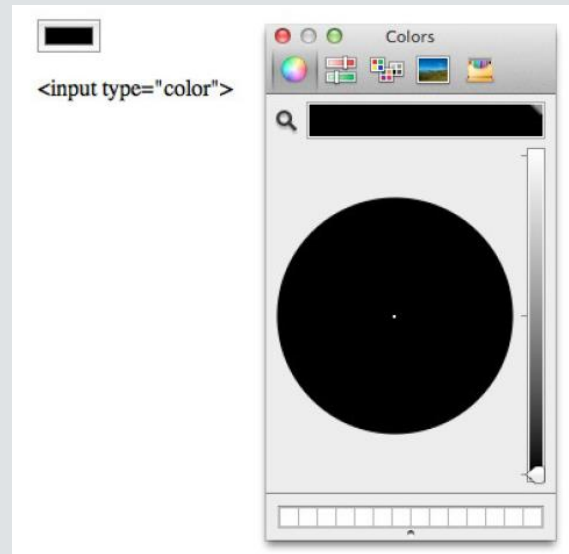
March	16	1971
April	17	1972
May	18	1973
June	19	1974
July	20	1975

No iPhone

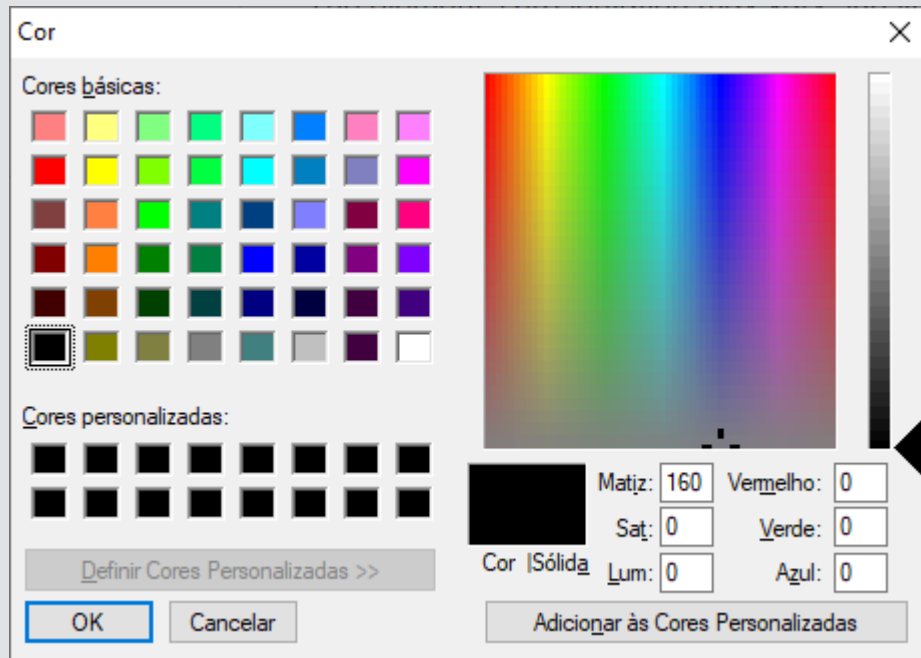
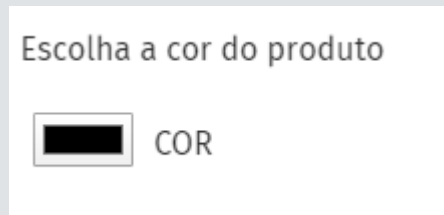
- **color**

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

- O elemento input do tipo color permite que seja exibido um "color picker" para o preenchimento do seu valor.
- O Chrome no Mac, por exemplo, exibe o color picker padrão do sistema:



- No navegador, quando você clica no elemento, aparece a janela de seleção de cor:



- **search**

`<input type="search" results="10">`

- O input do tipo search exibe um campo específico para busca.
- O atributo "results" determina quantas últimas buscas serão armazenadas e lembradas.

- **tel**

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

- O input do tipo tel foi especificado para coletar um número de telefone.
- Em dispositivos com teclados virtuais como smartphones e tablets, é comum o teclado ser adaptado para exibir apenas opções relevantes à entrada de números telefônicos.

Novos Atributos HTML5 em Elementos de Formulário

- Na especificação do HTML5 estão definidos novos atributos para os elementos de formulário, visando implementar algumas necessidades comuns que antes não eram possíveis de serem atendidas puramente com a marcação do formulário.

- **autofocus**
- Sua presença indica que aquele campo deve iniciar com foco quando a página for carregada.
- O usuário já pode começar a digitar algo sem nenhum clique.

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

- **placeholder**
- O atributo placeholder exibe o texto contido em seu valor dentro do elemento do formulário caso o seu valor seja vazio.

```
<input type="text" name="nome" placeholder="Insira seu nome">
```

Nome:

- **autocomplete, list e datalist**
- É possível implementar uma funcionalidade de sugestão de valores com mais facilidade.

```
<input type="text" list="cidades"
autocomplete="on">
```

```
<datalist id="cidades">
```

```
<option value="Brasília">
```

```
<option value="Rio de Janeiro">
```

```
<option value="São Paulo">
```

```
</datalist>
```

Cidade:

- Brasília
- Rio de Janeiro
- São Paulo

- A implementação de autocomplete sem o atributo list no campo, ligando-o a um datalist, vai utilizar os últimos valores utilizados em outros campos ou em outros formulários, dando prioridade a valores adicionados em inputs com o mesmo valor no atributo name .

Ícones

- O Bootstrap não inclui uma biblioteca de ícones por padrão, mas eles recomendam algumas bibliotecas para escolher. As recomendadas são:
 - Font Awesome
 - Iconic
 - Octicons
- Para saber mais detalhes:
<https://getbootstrap.com/docs/4.1/extend/icons/>

- Aqui vamos usar o Iconic, que é um código aberto com 223 ícones nos formatos SVG e fonte customizada.
- Para usarmos essa biblioteca é necessário baixar um arquivo .css (<https://useiconic.com/open/>) (open-iconic-bootstrap.css) e adicionar no HTML.
- Já baixe o arquivo e coloque na pasta em que estão os seus arquivos .css, pois usaremos em breve no nosso projeto.

- Para usarmos essa biblioteca é necessário adicionar o .css no HTML:

```
<link rel="stylesheet" href="css/open-iconic-bootstrap.css">
```

- Depois disso é muito simples utilizar os ícones:

```
<span class="oi oi-thumb-up"></span>
```

- Os ícones são disponibilizados através de uma fonte de texto customizada.
- A vantagem de se usar fontes para ícones é que o desenho fica escalável, como uma letra. Ele não perde qualidade em nenhum tamanho ou resolução por ser vetorial.
- E, assim como uma letra, podemos aplicar efeitos de texto como sombras e cores.
- A desvantagem é que cada ícone só pode ter um path no desenho e uma única cor. Não é possível usar ícones complexos com fontes.

Mais Exercícios de Formulários

- No arquivo checkout.html, o formulário de compra deve ter campos para o cliente digitar informações pessoais e informações sobre o pagamento.
- Para melhor organização, vamos separar os campos em dois fieldsets.

- Vamos criar o `<form>` **LOGO** depois do card, e ainda dentro do container. (atenção aos locais)
- Neste form, crie o primeiro `<fieldset>` usando `<legend>` para identificar os Dados Pessoais.
- Esse primeiro fieldset, dos Dados Pessoais, deve conter os campos Nome, Email, CPF e um checkbox para o usuário optar ou não por receber newsletter.

```
35 </div><!-- fim .card -->
```

```
36 <form>
```

```
37   <fieldset>
```

```
38     <legend>Dados pessoais</legend>
```

```
39     <div class="form-group">
```

```
40       <label for="nome">Nome completo</label>
```

```
41       <input type="text" class="form-control" id="nome" name="nome">
```

```
42     </div>
```

```
43     <div class="form-group">
```

```
44       <label for="email">Email</label>
```

```
45       <input type="email" class="form-control" id="email" name="email">
```

```
46     </div>
```

```
47     <div class="form-group">
```

```
48       <label for="cpf">CPF</label>
```

```
49       <input type="text" class="form-control" id="cpf" name="cpf">
```

```
50     </div>
```

```
51     <div class="form-group custom-control custom-checkbox">
```

```
52       <input type="checkbox" class="custom-control-input" id="newsletter" value="sim" checked>
```

```
53       <label class="custom-control-label" for="newsletter">
```

```
54         Quero receber Newsletter da Mirror Fashion
```

```
55       </label>
```

```
56     </div>
```

```
57   </fieldset>
```

```
58 </form>
```



Pressione **F11** para sair do modo tela cheia

Produto

Fuzzy Cardigan

Cor

Verde

Tamanho

40

Preço

R\$ 129,90

Dados pessoais

Nome completo

Email

CPF

☒ Quero receber Newsletter da Mirror Fashion

- Repare que cada campo possui um input e um label.
- Para agrupá-los, usamos uma div form-group do Bootstrap.
- Cada input deve ter uma classe form-control, atenção que para o input com type="checkbox" é necessário adicionar outras classes do Bootstrap.

- Agora vamos adicionar mais um `<fieldset>` ainda dentro do form (logo após a tag de fechamento do `<fieldset>` do exercício anterior) com os Dados do Cartão.
- Esse fieldset tem três campos: um com código do cartão, outro com a bandeira do cartão e outro com data de validade. Neste último, usaremos o input month do HTML5.

```
60 <fieldset>
61   <legend>Cartão de crédito</legend>
62   <div class="form-group">
63     <label for="numero-cartao">Número - CVV</label>
64     <input type="text" class="form-control" id="numero-cartao" name="numero-cartao">
65   </div>
66   <div class="form-group">
67     <label for="bandeira-cartao">Bandeira</label>
68     <select class="custom-select" id="bandeira-cartao">
69       <option disabled selected>Selecione uma opção...</option>
70       <option value="master">MasterCard</option>
71       <option value="visa">VISA</option>
72       <option value="amex">American Express</option>
73     </select>
74   </div>
75   <div class="form-group">
76     <label for="validade-cartao">Validade</label>
77     <input type="month" class="form-control" id="validade-cartao" name="validade-cartao">
78   </div>
79 </fieldset>
```

Dados pessoais

Nome completo

Email

CPF

☒ Quero receber Newsletter da Mirror Fashion

Cartão de crédito

Número - CVV

Bandeira

Validade

- Para finalizar, é necessário adicionar um botão antes do fechamento da tag form, que cuidará do envio dos dados digitados (vamos usar um btn-primary do Bootstrap).

```
81 | <button type="submit" class="btn btn-primary">  
    | Confirmar Pedido  
83 | </button>
```

Dados pessoais

Nome completo

Email

CPF

☒ Quero receber Newsletter da Mirror Fashion

Cartão de crédito

Número - CVV

Bandeira

Validade

Confirmar Pedido

- Adicione o atributo placeholder do HTML5 nos campos email e CPF com dicas de preenchimento.

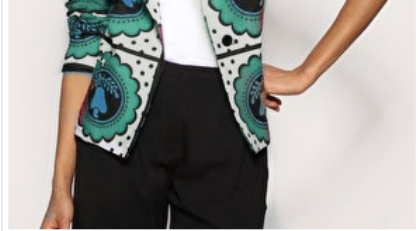
```
<input type="email" class="form-control" id="email" name="email" placeholder="email@exemplo.com">
```

...

```
<input type="text" class="form-control" id="cpf" name="cpf" placeholder="000.000.000-00">
```

- Adicione o atributo autofocus do HTML5 no input nome:

```
<input type="text" class="form-control" id="nome" name="nome" autofocus>
```

**Produto**

Fuzzy Cardigan

Cor

Verde

Tamanho

40

Preço

R\$ 129,90

Dados pessoais

Nome completo

Email

email@exemplo.com

CPF

000.000.000-00

☒ Quero receber Newsletter da Mirror Fashion

- Vamos incentivar o clique no botão de confirmação do pedido com um ícone além do texto.
- Use os Iconic pra isso. Primeiro adicione o link para essa biblioteca dentro da tag `<head>`.

```
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width">
6      <link rel="stylesheet" href="css/bootstrap.css">
7      <link rel="stylesheet" href="css/open-iconic-bootstrap.css">
8      <title>Checkout Mirror Fashion</title>
9  </head>
```


- Em seguida, dentro do botão, apenas adicione a linha que declara o ícone.

```
82 | <button type="submit" class="btn btn-primary">  
83 |   <span class="oi-thumb-up"></span>  
84 |   Confirmar Pedido  
85 | </button>
```

Dados pessoais

Nome completo

Email

CPF

☒ Quero receber Newsletter da Mirror Fashion

Cartão de crédito

Número - CVV

Bandeira

Validade



Confirmar Pedido

Validação no HTML5

- Entre as muitas novidades de formulários que vimos no HTML5, há toda uma parte de validação de dados com restrições expressas diretamente no código HTML.

- **required**
 - Podemos indicar na marcação do formulário quando um campo é de preenchimento obrigatório.
 - Esse atributo permite uma validação fraca no lado do cliente.

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

- **pattern**

- Conseguimos também especificar um formato requerido através do atributo pattern, adicionando uma expressão regular como valor.
- O atributo pattern também permite uma validação fraca do campo.

```
<input      type="text"      pattern="^@\w{2,}"  
name="usuario_twitter">
```

Validação no CSS

- A maioria dos novos componentes de formulário e os atributos que funcionam como validadores de campos na verdade somente aplicam uma pseudo-classe específica no campo que não está atendendo ao padrão ou requisito especificado.

- Essa pseudo-classe é a `:invalid`, e pode ser utilizada para dar um retorno visual imediato caso o usuário não esteja atendendo aos requisitos dos campos do formulário.

```
:invalid {  
    outline: 1px solid #CC0000;  
}
```

- Essa validação é fraca pois de maneira direta não é possível impedir que o usuário envie as informações do formulário, mesmo que incompletas ou incorretas.
- É possível porém alterar o botão de submit e deixá-lo desabilitado caso seja possível selecionar algum elemento por essa pseudo-classe no formulário.
- Essa verificação e alteração do elemento submit pode ser feita por JavaScript .

Exercícios - Validação

- Adicione o atributo required nos campos Nome e CPF e teste submeter o formulário sem preencher esses campos.

```
41 <div class="form-group">  
42   <label for="nome">Nome completo</label>  
43   <input type="text" class="form-control" id="nome" name="nome" autofocus required>  
  </div>
```

```
49 <div class="form-group">  
50   <label for="cpf">CPF</label>  
51   <input type="text" class="form-control" id="cpf" name="cpf" placeholder="000.000.000-00" required>  
52 </div>
```

- Algumas validações já são implícitas apenas por usarmos o input type correto. Por exemplo, tente submeter o formulário preenchendo o email com um valor inválido (com dois @ por exemplo).

Grid Responsivo do Bootstrap

- Uma das dificuldades mais comuns de um projeto front-end é o posicionamento de elementos, sobretudo em designs multi coluna.
- A solução mais comum é uso de grids, uma ideia antiga que veio dos próprios designers.
- Divide-se a tela em colunas e vamos encaixando os elementos dentro desse grid.

- Todo framework CSS moderno traz um grid pronto para utilização.
- Todo código CSS necessário para correto posicionamento já foi escrito e só precisamos usar as classes certas.
- O Bootstrap tem um grid pronto e várias classes para usarmos.
- O grid do Bootstrap trabalha com a ideia de 12 colunas e podemos escrever nosso código escolhendo quantas colunas ocupar.

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4				
.col-md-4				.col-md-4				.col-md-4				
.col-md-6						.col-md-6						

- Essas classes de coluna são as que definem o tamanho de cada elemento na página com base nas 12 partes do grid padrão. Em código:

```
<div class="row">  
  <div class="col-md-4">  
    ...  
  </div>  
  <div class="col-md-8">  
    ...  
  </div>  
</div>
```

- Nesse exemplo, deixamos a primeira `<div>` ocupando 4/12 da tela e a outra, 8/12.
- Repare que, para o grid funcionar, ao redor das colunas usamos uma `<div>` com a classe `row`.
- Ela é necessária para que o layout fique correto.

- Podemos ainda criar grids dentro de outro grid, sempre obedecendo a divisão de 12 colunas em cada. Por exemplo:

```
<div class="row">  
  <div class="col-md-4">  
    ...  
  </div>  
  <div class="col-md-8">  
    <div class="row">  
      <div class="col-md-6">  
        ...  
      </div>  
      <div class="col-md-6">  
        ...  
      </div>  
    </div>  
  </div>  
</div>
```


- Esse exemplo criou um segundo grid dentro da coluna da direita do primeiro grid.
- Nesse segundo grid há duas colunas ocupando metade cada uma (6/12). Mas como um grid está dentro do outro, na prática, ele vai ocupar metade do tamanho da `<div>` que tem 8/12 de tamanho.

- Um dos pontos mais interessantes dos grids é que eles são responsivos. Isso quer dizer que podemos aplicar diferentes layouts de colunas no nosso código ao mesmo tempo e cada um deles vai valer só em determinada situação.
- Nos códigos anteriores, por exemplo, usamos classes como `col-md-6`. O `md` nessa classe significa que vamos ocupar 6 colunas do grid apenas em telas maiores que 768px de largura. Em telas menores, automaticamente nosso grid será de uma coluna só.

- E, claro, temos classes pra outros tamanhos de tela também. No Bootstrap temos essas famílias de classes de grids já prontas:
 - col- : Extra small < 576px
 - col-sm- : Small >= 576px
 - col-md- : Medium >= 768px
 - col-lg- : Large >= 992px
 - col-xl- : Extra large >= 1200px

- Podemos aplicar mais de uma classe ao mesmo tempo no mesmo elemento:

```
<div class="row">  
  <div class="col-xs-6 col-sm-4">  
    ...  
  </div>  
  <div class="col-xs-6 col-sm-8">  
    ...  
  </div>  
</div>
```

- Nesse exemplo, nosso grid divide no meio (6 pra cada lado) em telas muito pequenas mas depois divide em 4 e 8 pra telas um pouco maiores.

Exercícios: Grid

Dicas para Estudo



Seja “CURIOSO”:

Procure revisar o que foi estudado.

Pesquise as referências bibliográficas.



Seja “ANTENADO”:

Leia a próxima aula.



Seja
“COLABORATIVO”:

Traga assuntos relevantes para a sala de aula.

Participe da aula.

Proponha discussões relevantes sobre o conteúdo.



Prof. Ms. Wilson Lourenço



**Dúvidas?
Não mais..**