

Desenvolvimento Web

Cap. 6 - Formulários HTML5

Universidade Estácio de Sá

Análise de Sistemas

Desenvolvimento Web em HTML5, CSS3, Javascript e PHP

Professor: Marlan Külberg

marlan.kulberg@estacio.br

Página de Compra

- Será criada a próxima página da loja
- Mostra os detalhes de um produto após o usuário clicar em um produto na lista da home
- Exibe uma foto grande, opções de cores e preço, descrição do produto e permite a compra
- Deve permitir que o usuário escolha a variação correta do produto
- Quando clicar no botão de comprar, as escolhas são enviadas para o servidor processar a compra
- Será utilizado um <form>
- Feito a partir de um combobox com <select> ou 3 radio buttons para escolher a cor da roupa dentre 3 opções possíveis
- No caso de radio buttons:

```
<form>  
  <input type="radio" name="cor">Verde  
  <input type="radio" name="cor">Rosa  
  <input type="radio" name="cor">Azul  
</form>
```

Formulário de Compra

- Várias falhas de acessibilidade e HTML semântico
- O texto não deve ficar solto na página
- Usa elemento <label> para associar com o respectivo input

```
<form>
  <input type="radio" name="cor" id="verde">
  <label for="verde">Verde</label>
  <input type="radio" name="cor" id="rosa">
  <label for="rosa">Rosa</label>
  <input type="radio" name="cor" id="azul">
  <label for="azul">Azul</label>
</form>
```

- É preciso explicar o que se está escolhendo com esses radio buttons
- Semanticamente, os 3 inputs representam a mesma coisa e devem ser agrupados em um <fieldset> que, por sua vez, recebe um <legend> com a legenda em texto apropriada
- botão de submit para enviar a escolha da compra.

```
<form>
  <fieldset>
    <legend>Escolha a cor</legend>
    <input type="radio" name="cor" id="verde">
    <label for="verde">Verde</label>
    <input type="radio" name="cor" id="rosa">
    <label for="rosa">Rosa</label>
    <input type="radio" name="cor" id="azul">
    <label for="azul">Azul</label>
  </fieldset>
  <button class="comprar">Comprar</button>
</form>
```

- É possível colocar a foto de verdade da roupa em vez da cor
- Usa o atributo alt na imagem para manter sua acessibilidade

```
<form>
  <fieldset>
    <legend>Escolha a cor</legend>
    <input type="radio" name="cor" id="verde">
    <label for="verde">
      
    </label>
    <input type="radio" name="cor" id="rosa">
    <label for="rosa">
      
    </label>
    <input type="radio" name="cor" id="azul">
    <label for="azul">
      
    </label>
  </fieldset>
  <button class="comprar">Comprar</button>
</form>
```

Atividade - Formulário da Página do Produto

1. Vamos implementar nossa página de produto, o primeiro passo é a construção de um HTML semântico. Vamos começar criando a estrutura básica do arquivo. Isto inclui as tags doctype, html, head, body, title.

Crie o arquivo produto.html com uma base parecida com essa:

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <title>Mirror Fashion</title>
    <link rel="stylesheet" href="css/reset.css">
    <link rel="stylesheet" href="css/estilos.css">
  </head>
  <body>
    ...
  </body>
</html>
```

Atividade - Formulário da Página do Produto

2. Como todas as páginas fazem parte da Mirror Fashion, é comum que tenham o mesmo cabeçalho e o mesmo rodapé.

Para termos o mesmo cabeçalho e o mesmo rodapé no produto.html, temos que copiar e colar o código do <header> e do <footer> dentro do <body>

```
<body>
  <header class="container">
    <h1>
      
    </h1>
    <p class="sacola">
      Nenhum item na sacola de compras
    </p>
    <nav class="menu-opcoes">
      <ul>
        <li><a href="#">Sua Conta</a></li>
        <li><a href="#">Lista de Desejos</a></li>
        <li><a href="#">Cartão Fidelidade</a></li>
        <li><a href="sobre.html">Sobre</a></li>
        <li><a href="#">Ajuda</a></li>
      </ul>
    </nav>
  </header>
  <!-- adicionar o conteúdo da página -->
  <footer>
    <div class="container">
      
      <ul class="social">
        <li><a href="http://facebook.com/mirrorfashion">Facebook</a></li>
        <li><a href="http://twitter.com/mirrorfashion">Twitter</a></li>
        <li><a href="http://plus.google.com/mirrorfashion">Google+</a></li>
      </ul>
    </div>
  </footer>
</body>
```

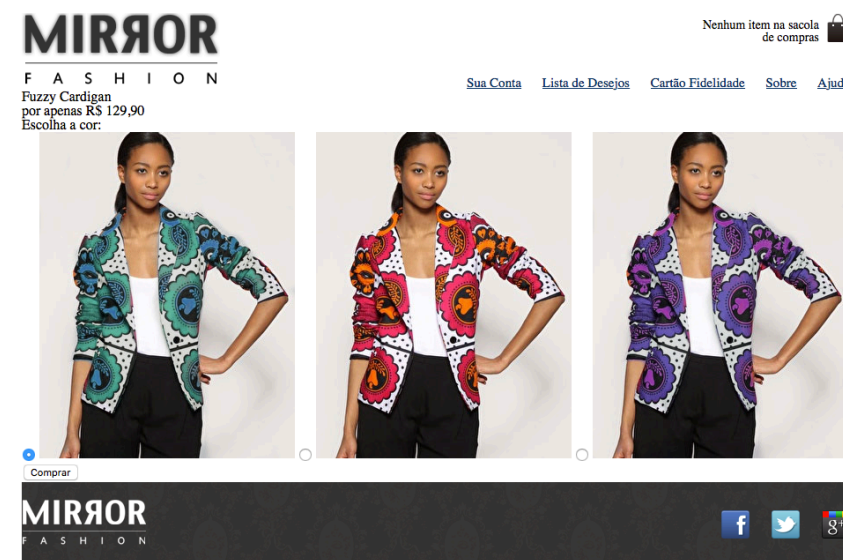
Atividade - Formulário da Página do Produto

3. Edite a página produto.html e, entre as tags <header> e <footer> , adicione um formulário com radios e labels para a escolha da cor. Também usaremos o atributo alt nas imagens para melhorar a acessibilidade:

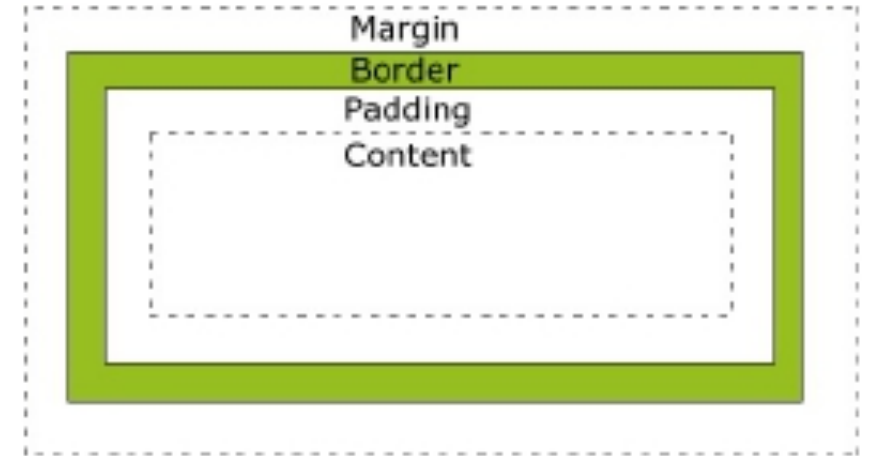
```
<div class="produto">
  <h1>Fuzzy Cardigan</h1>
  <p>por apenas R$ 129,90</p>
  <form>
    <fieldset class="cores">
      <legend>Escolha a cor:</legend>
      <input type="radio" name="cor" value="verde" id="verde" checked>
      <label for="verde">
        
      </label>
      <input type="radio" name="cor" value="rosa" id="rosa">
      <label for="rosa">
        
      </label>
      <input type="radio" name="cor" value="azul" id="azul">
      <label for="azul">
        
      </label>
    </fieldset>
    <button class="comprar">Comprar</button>
  </form>
</div>
```

4. Teste o HTML do exercício anterior.

Veja seu funcionamento sem interferência do CSS e do JS que será feito depois.



Box Model



Box Model padrão do W3C

- Ao alterar propriedades de elementos dentro de uma página, é preciso perceber o comportamento das alterações na presença de outros elementos
- Uma forma é pensar no box model
- O box model é constituído de quatro áreas retangulares que se desenvolvem de dentro para fora
 - conteúdo (content): aquilo que será exibido;
 - espaçamento (padding): distância entre a borda e o conteúdo
 - borda (border): quatro linhas que envolvem a caixa (box)
 - margem (margin): distância que separa um box de outro

Box-Sizing

- No IE6 a Microsoft trocou o box model para que o width significasse o tamanho total até a borda
- Criou problemas de incompatibilidade entre navegadores
- O IE usa o box model oficial como padrão a partir do IE8
- A alteração da Microsoft se transformou no box-sizing do CSS3, que permite trocar o box model a ser utilizado
- Por padrão, todos os elementos têm valor box-sizing: content-box, que indica que o tamanho é definido pelo seu conteúdo
- Mas é possível trocar por box-sizing: border-box, que indica que o tamanho agora considerará até a borda
 - width será a soma do conteúdo com a borda e o padding

Atividade - Página de Produto

Vamos estruturar o CSS para implementar a funcionalidade de troca de cores.

A cada passo, teste no browser para ir acompanhando o resultado.

1. Crie um novo arquivo produto.css na pasta css/.
2. Em produto.html, importe o produto.css após todos os outros CSS's:

```
<link rel="stylesheet" href="css/produto.css">
```

Atividade - Página de Produto

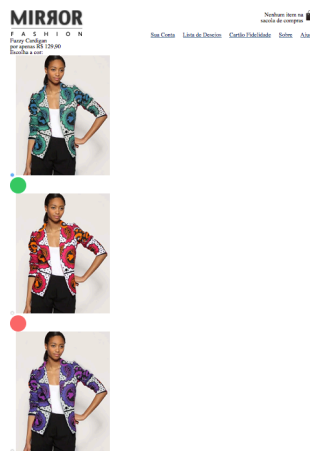
3. Primeiro, vamos desenhar as bolinhas coloridas com pseudo-elementos do CSS3 usando um truque com bordas redondas grandes:

```
.cores label::after {  
  content: " ";  
  display: block;  
  border-radius: 50%;  
  width: 50px;  
  height: 50px;  
}
```

```
label[for=verde]::after {  
  background-color: #33CC66;  
}
```

```
label[for=rosa]::after {  
  background-color: #FF6666;  
}
```

```
label[for=azul]::after {  
  background-color: #6666FF;  
}
```



Próximo passo é estilizar a bolinha selecionada usando pseudo classe :checked :

```
.cores input:checked + label::after {  
  border: 6px solid rgba(0,0,0,0.3);  
}
```

A borda da bolinha selecionada aumenta o tamanho total da bolinha por causa do box model padrão. Uma solução é trocar o box model com a propriedade box-sizing :

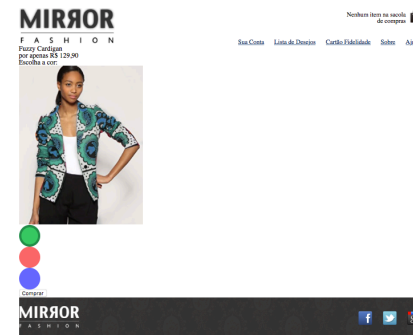
```
.cores label::after {  
  box-sizing: border-box;  
}
```

Agora, a bolinha do input radio é desnecessária. Esconda-a:

```
.cores input[type=radio] {  
  display: none;  
}
```

Para fechar a funcionalidade de escolha das cores, falta exibir apenas a imagem atualmente selecionada. Outra forma de falar isso é que devemos esconder as imagens dos radios não selecionados. Podemos usar um seletor avançado para isso:

```
.cores input:not(:checked) + label img {  
  display: none;  
}
```



Atividade - Página de Produto

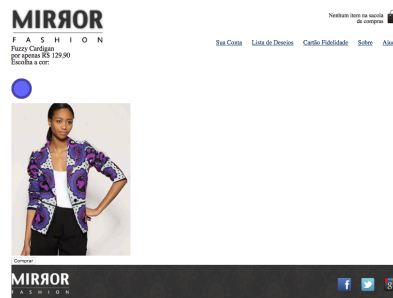
4. A troca de imagens deve estar funcionando, apesar das coisas não estarem ainda posicionadas corretamente.

5. Vamos implementar o posicionamento correto das bolinhas lado a lado. Para isso, use `position: absolute` já que seus tamanhos são conhecidos:

```
.cores label::after {  
  position: absolute;  
  top: 30px;  
}
```

As bolinhas vão ser posicionadas com relação ao `fieldset cores`, então ele precisa estar posicionado. O `padding superior` é para abrir espaço para as bolinhas:

```
.cores {  
  position: relative;  
  padding-top: 90px;  
}
```



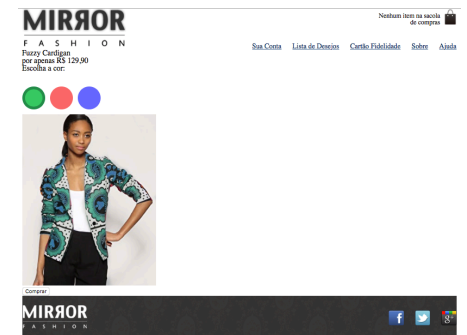
As bolinhas ficaram sobrepostas na esquerda. Para corrigir, basta colocar uma coordenada `left` diferente para cada uma:

```
label[for=verde]::after {  
  left: 0;  
}
```

```
label[for=rosa]::after {  
  left: 60px;  
}
```

```
label[for=azul]::after {  
  left: 120px;  
}
```

Teste o resultado no navegador.



Atividade - Página de Produto

6. Um ponto importante de uma solução responsiva é que as imagens se adaptem ao tamanho da tela.

Às vezes, usamos imagens maiores e, quando a tela é pequena, a imagem fica "vazando" para fora do elemento pai.

Uma forma de corrigir esse problema é garantir que ela nunca passe o tamanho do pai com maxwidth:

```
.cores label img {  
  display: block;  
  max-width: 100%;  
}
```



7. Com toda a parte funcional e de posicionamento pronta, vamos estilizar alguns detalhes visuais da página.

Primeiro, detalhes de tipografia e espaçamento para toda página de produtos:

```
.produto {  
  color: #333;  
  line-height: 1.3;  
  margin-top: 2em;  
}
```

O nome do produto e seu preço também ganham estilo:

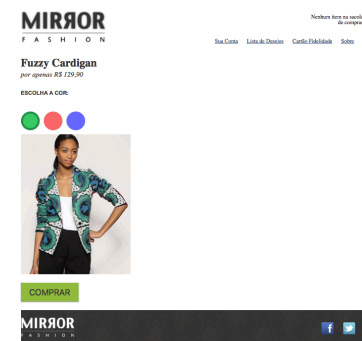
```
.produto h1 {  
  font-size: 1.8em;  
  font-weight: bold;  
}  
  
.produto p {  
  font-size: 1.2em;  
  font-style: italic;  
  margin-bottom: 1em;  
}
```

O <legend> ganha um destaque:

```
.produto legend {  
  display: block;  
  font: bold 0.9em/2.5 Arial;  
  text-transform: uppercase;  
}
```

E o botão de comprar deve ficar em evidência:

```
.comprar {  
  background: #91BD3C;  
  border: none;  
  color: #333;  
  font-size: 1.4em;  
  text-transform: uppercase;  
  box-shadow: 0 1px 3px #777;  
  display: block;  
  padding: 0.5em 1em;  
  margin: 1em 0;  
}
```



Teste e observe o estilo simples da página.

Atividade - Página de Produto

8. (opcional) Quando selecionamos a bolinha, uma borda escura aparece. Use transition para fazer a borda aparecer suavemente. Adicione o seletor no início de produto.css:

```
.cores label::after {  
  border: 6px solid rgba(0,0,0,0);  
  transition: 1s;  
}
```

Podemos colocar também um estilo para quando passar o mouse em cima da bolinha, mostrando uma borda mais leve, adicionar também no início de produto.css:

```
.cores label:hover::after {  
  border: 6px solid rgba(0,0,0,0.1);  
}
```

HTML5 Input Range

- Existem muitas formas corretas e semânticas de implementar seleção de tamanhos no formulário:
 - select
 - radio buttons
 - input range do HTML5
- O `<input type="range">` é um componente que representa um slider numérico
- Ele recebe atributos min e max com o intervalo numérico possível
- Opcionalmente, há o atributo step que indica de quanto em quanto o número deve pular
- Existe também o atributo value que indica o valor inicial do input
- `<input type="range" min="36" max="46" value="42" step="2" name="tamanho">`
- Componente já possui estilo padrão do navegador

Atividade - Seletor de Tamanho

1. Implemente a funcionalidade de escolher o tamanho da roupa usando um input range do HTML5, colocando o código a seguir logo abaixo do fechamento do nosso primeiro <fieldset> no arquivo produto.html:

```
<fieldset class="tamanhos">
  <legend>Escolha o tamanho:</legend>
  <input type="range" min="36" max="46" value="42" step="2" name="tamanho" id="tamanho">
</fieldset>
```

Teste seu funcionamento

2. Temos dois fieldsets, um para escolher cor e outro, tamanho. Na página, podemos posicioná-los lado a lado. Dentro de produto.css, acrescente:

```
fieldset {
  display: inline-block;
  vertical-align: top;
  margin: 1em 0;
  min-width: 240px;
  width: 45%;
}
```



Tabelas

- Tabelas eram muito comuns para a definição de áreas.
- Seu uso se tornou prejudicial pela complexidade da marcação, que dificulta a manutenção das páginas
- Implicação direta na definição de relevância do conteúdo das tabelas para os indexadores de conteúdo por mecanismos de busca
- Ainda assim, hoje, quando se quer exibir uma série de dados tabulares, é indicado o uso da tag de tabela `<table>`

```
<table>
  <tr>
    <th>Título da primeira coluna</th>
    <th>Título da segunda coluna</th>
  </tr>
  <tr>
    <td>Linha 1, coluna 1</td>
    <td>Linha 1, coluna 2</td>
  </tr>
  <tr>
    <td>Linha 2, coluna 1</td>
    <td>Linha 2, coluna 2</td>
  </tr>
</table>
```

- Na primeira linha `<tr>` da tabela, as células são indicadas com a tag `<th>`, para diferenciar seu conteúdo das células de dados
- Existem diversas maneiras de se alterar uma estrutura de uma tabela:
 - Indicar que uma célula `<td>` ou `<th>` ocupa mais de uma linha de altura usando atributo `rowspan`
 - Indicar que ocupa mais de uma coluna de largura usando atributo `colspan`
- Pode-se adicionar título à tabela com a tag `<caption>`
- Ainda existem as tags `<thead>`, `<tfoot>` e `<tbody>`, para agrupar as linhas da tabela
- Dentro do grupo `<thead>` deve haver apenas linhas contendo a tag `<th>`
- como célula

Tabelas

- Tag <colgroup> permite que colunas sejam definidas
- Dentro dessa tag define-se uma tag <col> para cada coluna
- Assim, pode-se adicionar alguns atributos que influenciarão todas as células daquela coluna.

```
<table>
  <caption>Quantidade e preço de camisetas.</caption>
  <colgroup>
    <col width="10%">
    <col width="40%">
    <col width="30%">
    <col width="20%">
  </colgroup>
  <thead>
    <tr>
      <th rowspan="2">
      <th colspan="2">Quantidade de Camisetas</th>
      <th rowspan="2">Preço</th>
    </tr>
    <tr>
      <th>Amarela</th>
      <th>Vermelha</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td>
      <td>Total de camisetas amarelas: 35</td>
      <td>Total de camisetas vermelhas: 34</td>
      <td>Valor total: $45.00</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>Polo</td>
      <td>12</td>
      <td>5</td>
      <td>$30.00</td>
    </tr>
    <tr>
      <td>Regata</td>
      <td>23</td>
      <td>29</td>
      <td>$15.00</td>
    </tr>
  </tbody>
</table>
```

Atividade - Detalhes Finais

1. Crie a seção de detalhes logo abaixo da div com a classe produto , mas ainda dentro do container:

```
<div class="container">
  <div class="produto">
    ...
  </div>
  <div class="detalhes">
    <h2>Detalhes do produto</h2>
    <p>Esse é o melhor casaco de Cardigã que você já viu.
    Excelente material italiano com estampa desenhada
    pelos artesãos da comunidade de Krotor nas ilhas
    gregas. Compre já e receba hoje mesmo pela nossa
    entrega a jato.</p>
  </div>
</div>
```

2. O estilo é bastante simples, apenas para deixar o texto mais bonito. Adicione no final de produto.css:

```
.detalhes {
  padding: 2em 0;
}

.detalhes h2 {
  font-size: 1.5em;
  line-height: 2;
}

.detalhes p {
  margin: 1em 0;
  font-size: 1em;
  line-height: 1.5;
  max-width: 36em;
}
```

Atividade - Detalhes Finais

3. Crie uma tabela com características do produto contendo informações técnicas. Adicione dentro da div detalhes:

```
<table>
  <thead>
    <tr>
      <th>Característica</th>
      <th>Detalhe</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Modelo</td>
      <td>Cardigã 7845</td>
    </tr>
    <tr>
      <td>Material</td>
      <td>Algodão e poliester</td>
    </tr>
    <tr>
      <td>Cores</td>
      <td>Azul, Rosa e Verde</td>
    </tr>
    <tr>
      <td>Lavagem</td>
      <td>Lavar a mão</td>
    </tr>
  </tbody>
</table>
```

Detalhes do produto

Esse é o melhor casaco de Cardigã que você já viu. Excelente material italiano com estampa desenhada pelos artesãos da comunidade de Krotor nas ilhas gregas. Compre já e receba hoje mesmo pela nossa entrega a jato.

Característica	Detalhe
Modelo	Cardigã 7845
Material	Algodão e poliester
Cores	Azul, Rosa e Verde
Lavagem	Lavar a mão

4. Estilize a tabela para deixá-la mais agradável. Use o seletor de filhos múltiplos para um estilo zebrado. Adicione o estilo em produto.css:

```
table {
  border-spacing: 0.2em;
  border-collapse: separate;
}

thead {
  background-color: #999;
}

thead th {
  font-weight: bold;
  padding: 0.3em 1em;
  text-align: center;
}

td {
  padding: 0.3em;
}

tr:nth-child(2n) {
  background-color: #CCC;
}

td:first-child {
  font-style: italic;
}
```

Detalhes do produto

Esse é o melhor casaco de Cardigã que você já viu. Excelente material italiano com estampa desenhada pelos artesãos da comunidade de Krotor nas ilhas gregas. Compre já e receba hoje mesmo pela nossa entrega a jato.

Característica	Detalhe
Modelo	Cardigã 7845
Material	Algodão e poliester
Cores	Azul, Rosa e Verde
Lavagem	Lavar a mão

Atividade - Detalhes Finais

1. Para implementarmos o fundo cinza em tela cheia, vamos precisar de um novo elemento pai para conter todos os elementos da página.

Crie um `<div class="produto-back">` ao redor da `<div class="container">` que tínhamos antes.

Apenas para referência, nesse momento, seu HTML deve estar mais ou menos assim:

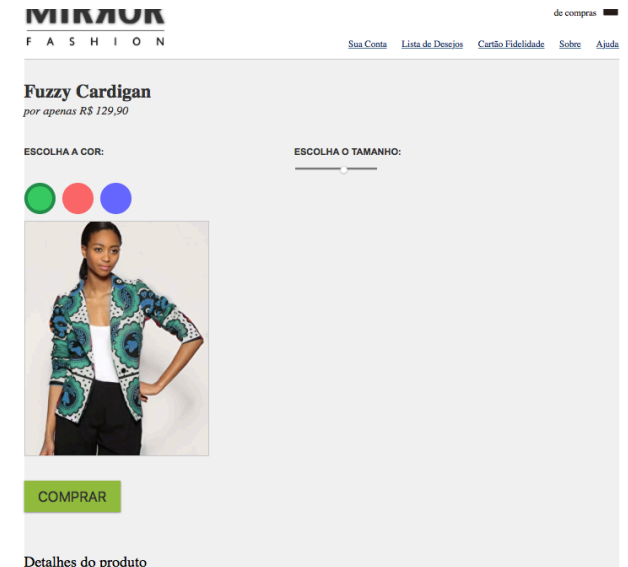
```
<div class="produto-back">
  <div class="container">
    <div class="produto">
      ...
    </div>
    <div class="detalhes">
      ...
    </div>
  </div>
</div>
```

2. O estilo é bastante simples, apenas usando uma cor e bordas sutis para criar um efeito mais elegante:

Adicione o estilo em produto.css:

```
.produto-back {
  background-color: #F2F2F2;
  margin-top: 1em;
  border-top: 2px solid #CCC;
}
```

```
.cores label img {
  border: 2px solid #CCC;
}
```



Bootstrap e Frameworks de CSS

- Ao invés de começar todo projeto do zero, criando todo estilo na mão, frameworks trazem toda uma base construída
- Bootstrap tem maior notoriedade

<http://getbootstrap.com>

- O Bootstrap traz uma série de recursos:
 - Reset CSS
 - Estilo visual base pra maioria das tags
 - Ícones
 - Grids prontos pra uso
 - Componentes CSS
 - Plugins JavaScript

Estilo e Componentes Base

Para usar o Bootstrap, deve incluir seu CSS na página:

```
<link rel="stylesheet" href="link para versão atualizada do bootstrap">
```

Um reset é aplicado
tags ganham estilo e tipografia base

Muitas classes com componentes adicionais podem ser aplicadas na página

Exemplo: criar um título com uma frase de abertura em destaque, usando o Jumbotron:

```
<div class="jumbotron jumbotron-fluid">  
  <div class="container">  
    <h1 class="display-4">Ótima escolha!</h1>  
    <p class="lead">Obrigado por comprar na Mirror Fashion.</p>  
  </div>  
</div>
```

Página de Checkout da Mirror Fashion

- Após escolher o produto desejado, o usuário é encaminhado para a página de efetivação da compra
- Loja foi otimizada pra compra direta, sem carrinho de compras
- O cliente escolhe o produto e compra direto, com um clique
- É preciso coletar os dados do cliente e do pagamento
- O foco da nova página é a coleta de informações para efetivação da compra
- Formulário complexo com os campos necessários.
- Bootstrap será utilizado para desenvolver a página com mais facilidade e rapidez.

Ótima escolha!

Obrigado por comprar na Mirror Fashion! Preencha seus dados para efetivar a compra.

<div>Sua compra</div> <div></div> <div>Produto Cardigan Thelux Basic Cor azul Tamanho 38</div>	<div>Dados pessoais</div> <div>Nome completo</div> <div><input type="text"/></div> <div>E-mail</div> <div><input type="text" value="email@exemplo.com"/></div> <div>CPF</div> <div><input type="text" value="000.000.000-00"/></div> <div><input checked="" type="checkbox"/> Quero receber spam da Mirror Fashion</div>	<div>Cartão de crédito</div> <div>Número - CVV</div> <div><input type="text" value="0000 0000 0000 0000 - 010"/></div> <div>Bandeira</div> <div><input type="text" value="MasterCard"/></div> <div>Validade</div> <div><input type="text"/></div>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Continuar pedido

Atividade: Página de Checkout

1. Crie a página checkout.html com HTML simples.

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <title>Checkout Mirror Fashion</title>
  </head>
  <body>
    <h1>Ótima escolha!</h1>
    <p>Obrigado por comprar na Mirror Fashion!
    Preencha seus dados para efetivar a compra.</p>
    <h2>Sua compra</h2>
    <dl>
      <dt>Produto</dt>
      <dd>Fuzzy Cardigan</dd>
      <dt>Cor</dt>
      <dd>Verde</dd>
      <dt>Tamanho</dt>
      <dd>40</dd>
      <dt>Preço</dt>
      <dd>R$ 129,90</dd>
    </dl>
  </body>
</html>
```

Ótima escolha!

Obrigado por comprar na Mirror Fashion! Preencha seus dados para efetivar a compra.

Sua compra

Produto	Fuzzy Cardigan
Cor	Verde
Tamanho	40
Preço	R\$ 129,90

Atividade: Página de Checkout

2. Abra a página checkout.html no navegador e veja que está com o estilo do navegador. Inclua o arquivo CSS do bootstrap na página. Vai haver uma mudança sutil no estilo da página, principalmente em aspectos tipográficos.

Coloque no <head> da página de checkout o CSS do bootstrap:

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">
```

Teste novamente a página.

Ótima escolha!

Obrigado por comprar na Mirror Fashion! Preencha seus dados para efetivar a compra.

Sua compra

Produto
Fuzzy Cardigan
Cor
Verde
Tamanho
40
Preço
R\$ 129,90

3. O primeiro componente pronto do bootstrap que vamos usar é o jumbotron. É basicamente a abertura do site, contendo sua chamada principal.

Para usá-lo basta criar uma div com a classe jumbotron e a classe jumbotron-fluid

Envolva as chamadas de abertura que já tínhamos com h1 e p em duas <div> . A primeiro div contém class="jumbotron jumbotron-fluid" e a segunda, class="container"

```
<div class="jumbotron jumbotron-fluid">
  <div class="container">
    <!-- h1 e p que já tínhamos, somente adicionar as respectivas classes -->
    <h1 class="display-4">Ótima escolha!</h1>
    <p class="lead">Obrigado por comprar na Mirror Fashion!
    Preencha seus dados para efetivar a compra.</p>
  </div><!-- fim .container dentro do jumbotron -->
</div><!-- fim .jumbotron -->
```

Abra a página e note que um estilo diferente aparece. Teste redimensionar o navegador e veja que o tamanho da fonte e espaçamento do componente se ajustam automaticamente.

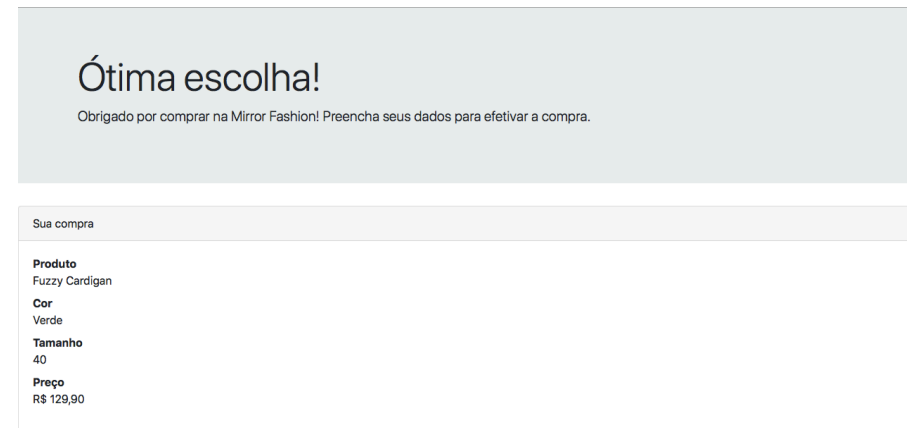
O Bootstrap usa responsive design automaticamente em seus componentes.

Atividade: Página de Checkout

4. Use um outro componente do Bootstrap, o card para organizar a seção que mostra as informações da compra do cliente. Cuidado com os nomes das classes, que podem confundir

Adapte o HTML do h2 "Sua compra" e do dl que temos para se adequar ao componente de card, adicionando o seguinte código após o fechamento da `<div class="jumbotron">` :

```
<div class="card mb-3">
  <div class="card-header">
    Sua compra
  </div><!-- fim .card-header -->
  <div class="card-body">
    <!-- ... aqui vai o <dl> que já temos ... -->
  </div><!-- fim .card-body -->
</div><!-- fim .card -->
```



Repare como os nomes das classes, apesar de serem muitos, fazem sentido para isolar cada parte do cartão

Teste novamente a página no navegador e veja o resultado. Temos um cartão arredondado com título em destaque no topo

Atividade: Página de Checkout

5. Repare no exercício anterior do jumbotron que o `<div class="container">` é responsável por centralizar e dar espaçamento na tela.

Muito parecido com o container criado antes no projeto, mas agora é uma classe do Bootstrap.

Crie uma outra div container pra conter o card criado e veja como ele fica melhor posicionado no centro da tela.

```
<div class="container">  
  <!-- ... card aqui ... -->  
</div><!-- fim .container da pagina -->
```

Ótima escolha!

Obrigado por comprar na Mirror Fashion! Preencha seus dados para efetivar a compra.

Sua compra

Produto
Fuzzy Cardigan
Cor
Verde
Tamanho
40
Preço
R\$ 129,90

Atividade: Página de Checkout

6. Dentro do card-body , logo no topo, acima da lista de definições <dl> , colocar uma foto do produto na cor escolhida

Com Bootstrap, é possível ainda acrescentar algumas classes nessa imagem.

A classe img-thumbnail faz a imagem ficar flexível e nunca estourar o tamanho do pai, também faz a imagem ficar centralizada e com uma borda de destaque.

A classe mb-3 adiciona uma pequena margem na parte inferior da imagem.

Adicione a imagem do produto logo acima da lista <dl> dentro do div card-body :

```

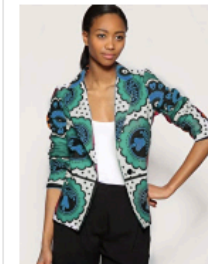
```

Teste novamente a página.

Ótima escolha!

Obrigado por comprar na Mirror Fashion! Preencha seus dados para efetivar a compra.

Sua compra



Produto
Fuzzy Cardigan
Cor
Verde
Tamanho
40
Preço
R\$ 129,90

Formulários

- Atributos do form

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

- O valor do atributo action é o endereço para onde as informações do formulário serão enviadas,
- Valor depende inteiramente de como é feita a aplicação que receberá essas informações no lado do servidor
- O atributo method especifica o método do HTTP pelo qual essa informação será transmitida
- O valor **post** significa que a informação desse formulário será salva de alguma maneira
- O valor **get** é utilizado para obter algo a partir das informações a transmitir

Componentes

somente a marcação da tag <form> não mostra nenhum elemento visível no navegador.

Exemplo

Pegar como informação o nome do visitante do site e guardar em um banco de dados:

```
<form action="/efetivar.html" method="POST">  
  <label for="nome">Nome:</label>  
  <input type="text" name="nome" id="nome">  
  <input type="submit">  
</form>
```

Label

- Esse elemento **label** é uma tag de conteúdo
- Seu texto é exibido de maneira comum dentro do formulário
- A diferença é que essa marcação faz uma ligação com outro elemento no formulário
- No exemplo, label tem o atributo for, que recebe o valor nome
- Ao clicar 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, o elemento vinculado ao label é um campo de texto que declarado com a tag input

Input - text

- Maioria dos elementos utilizados nos formulários para capturar informações dos usuários são da tag <input>
- No exemplo, foram utilizadas duas variações dessa tag
- Os tipos de inputs são determinados pelo valor do seu atributo type

text

- `<input type="text" name="nome_usuario">`
- Tipo mais comum de input
- Utilizado quando se deseja que o usuário envie uma informação textual simples (não permite quebras de linha)
- Ao enviar o formulário, a informação digitada pelo usuário é acessível no lado do servidor por meio do atributo name

textarea

- Para permitir que o usuário insira maior quantidade texto, incluindo quebras de linha, faz-se uso da tag textarea

`<textarea name="texto"></textarea>`

Input - password e checkbox

password

- type="password" é similar ao tipo text , mas exibe uma série de símbolos * ou outros

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

checkbox

- Input do tipo checkbox exibe uma caixa para marcação, utilizado quando há uma opção que pode ser marcada como sim ou não
- Muito utilizado com o valor true, mas 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>
```

Input - Radio

radio

- Quando se deseja que o usuário escolha somente uma entre uma série de opções
- Se há mais de um elemento desse tipo com o mesmo valor no atributo name, apenas um poderá ser selecionado

```
<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>
```

Input - image, file e hidden

image

- É possível substituir o botão de envio do formulário por uma imagem
- CSS3, permite adicionar imagens que não fazem parte do conteúdo, sendo somente estilo

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

file

- Necessário para enviar um arquivo para a aplicação no lado do servidor
- Para o correto envio dos arquivos, as vezes também é necessário adicionar o atributo enctype="multipart/form-data" na tag <form>

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

hidden

- Usado para enviar e receber informações que não serão exibidos no formulário.

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

Input - button, submit e reset

button

- Atributo type="button" renderiza um botão dentro do formulário, mas sem função direta
- Utilizado para disparar eventos para a execução de scripts
- Texto do botão determinado pelo atributo value

```
<input type="button" name="mostra_dialogo" value="Clique aqui!">
```

submit

- Atributo type="submit" é similar ao type="button", mas quando acionado inicia o envio das informações do formulário para o endereço indicado no atributo action do <form>

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

reset

- type="reset" elimina valores digitados anteriormente nos elementos de um formulário

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

A tag <input> dos tipos button, submit e reset pode ser substituída pela tag <button>

texto do botão passa a ser indicado como conteúdo da tag.

É necessário especificar o valor do atributo type, mesmo 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>
```

Select, Option e Optgroup

Para permitir a seleção entre diversas opções, utiliza-se 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>
```

Na configuração padrão, select exibe menu drop-down, permitindo a seleção de apenas uma opção

Atributo multiple permite selecionar mais de uma opção

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

Dependendo do número de opções apresentadas é possível agrupá-las:

```
<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="zl_sp">Zona Leste</option>
    <option value="centro_sp">Centro</option>
  </optgroup>
</select>
```

Novos atributos HTML5 em elementos de Formulário

autofocus

- Indica que aquele campo deve iniciar com foco quando a página for carregada. Usuário pode começar a digitar algo sem nenhum clique

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

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">
```

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>
```

Ícones

- Bootstrap não possui biblioteca de ícones por padrão, mas recomenda algumas bibliotecas
- Para saber mais detalhes: <https://getbootstrap.com/docs/4.1/extend/icons/>
- No curso, será usado o Iconic, de código aberto com 223 ícones nos formatos SVG e fonte customizada
- Necessário baixar um arquivo .css e adicionar no HTML:
`<link rel="stylesheet" href="css/open-iconic-bootstrap.css">`
- Depois disso é muito simples utilizar os ícones:
``
- 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
- Pode-se 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

Atividades - Formulários

1. O formulário de compra possui campos para o cliente digitar informações pessoais e informações sobre o pagamento. Para melhor organização, vamos separar os campos em dois fieldsets.

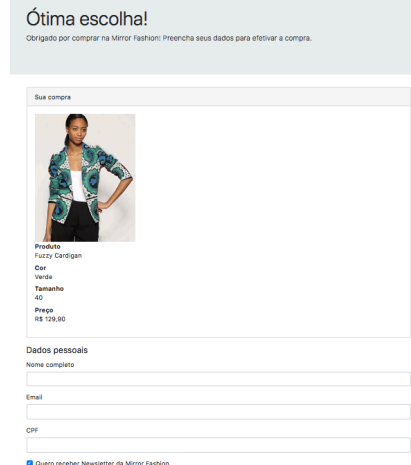
Vamos criar o `<form>` logo depois do card, e ainda dentro do container. Neste form, crie o primeiro `<fieldset>` usando `<legend>` para identificar os Dados Pessoais.

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.

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.

Teste a página no navegador e observe o estilo padrão que ganhamos apenas por usar o Bootstrap.

```
<form>
  <fieldset>
    <legend>Dados pessoais</legend>
    <div class="form-group">
      <label for="nome">Nome completo</label>
      <input type="text" class="form-control" id="nome" name="nome">
    </div>
    <div class="form-group">
      <label for="email">Email</label>
      <input type="email" class="form-control" id="email" name="email">
    </div>
    <div class="form-group">
      <label for="cpf">CPF</label>
      <input type="text" class="form-control" id="cpf" name="cpf">
    </div>
    <div class="form-group custom-control custom-checkbox">
      <input type="checkbox" class="custom-control-input" id="newsletter"
        value="sim" checked>
      <label class="custom-control-label" for="newsletter">
        Quero receber Newsletter da Mirror Fashion
      </label>
    </div>
  </fieldset>
</form>
```




Atividades - Formulários

2. Adicione 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.

Teste novamente no navegador.

```
<fieldset>
  <legend>Cartão de crédito</legend>
  <div class="form-group">
    <label for="numero-cartao">Número - CVV</label>
    <input type="text" class="form-control" id="numero-cartao" name="numero-cartao">
  </div>
  <div class="form-group">
    <label for="bandeira-cartao">Bandeira</label>
    <select class="custom-select" id="bandeira-cartao">
      <option disabled selected>Selecione uma opção...</option>
      <option value="master">MasterCard</option>
      <option value="visa">VISA</option>
      <option value="amex">American Express</option>
    </select>
  </div>
  <div class="form-group">
    <label for="validade-cartao">Validade</label>
    <input type="month" class="form-control" id="validade-cartao" name="validade-cartao">
  </div>
</fieldset>
```



Produto
Fuzzy Cardigan

Cor
Verde

Tamanho
40

Preço
R\$ 129,90

Dados pessoais

Nome completo

Email

CPF

☒ Quero receber Newsletter da Mirror Fashion

Cartão de crédito

Número - CVV

Bandeira

Selecione uma opção...

Validade

Atividades - Formulários

Cartão de crédito

Número - CVV

Bandeira

Selecione uma opção...

Validade

Confirmar Pedido

3. 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).

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

4. 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>
```

Dados pessoais

Nome completo

Email

CPF

☒ Quero receber Newsletter da Mirror Fashion

Atividades - Formulários

5. 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> .

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/open-iconic/1.1.1/font/css/open-iconic-bootstrap.min.css">
```

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

```
<button type="submit" class="btn btn-primary">  
  <span class="oi oi-thumb-up"></span>  
  Confirmar Pedido  
</button>
```



Para saber mais sobre os ícones: <https://useiconic.com/open/>

6. Use outras classes do Bootstrap para ajustar mais detalhes. No botão, adicione a classe btn-lg para deixar o botão maior.

Veja mais opções de botões com Bootstrap: <https://getbootstrap.com.br/docs/4.1/components/buttons/>

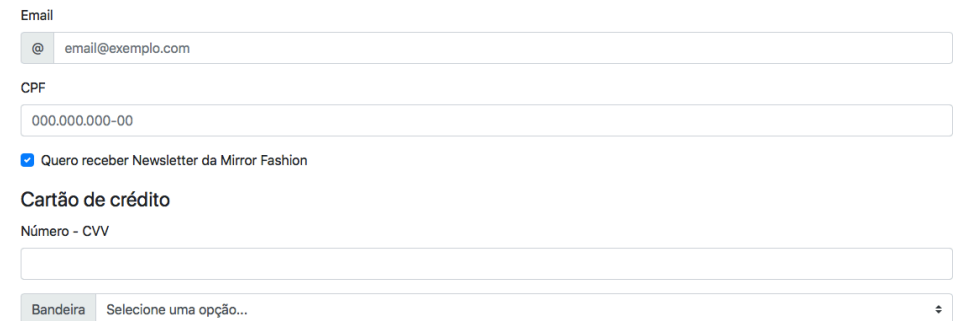
Atividades - Formulários

7. (opcional) O Bootstrap tem outros recursos para formulários, como os input groups. Teste trocando o código do campo email para isso:

```
<div class="form-group">
<label for="email">Email</label>
  <div class="input-group mb-3">
    <div class="input-group-prepend">
      <span class="input-group-text">@</span>
    </div>
    <input type="email" class="form-control" id="email" name="email" placeholder="email@exemplo.com">
  </div>
</div>
```

Veja outro exemplo do uso do input groups. Troque o código do campo Bandeira do cartão de crédito para isso:

```
<div class="form-group">
  <div class="input-group mb-3">
    <div class="input-group-prepend">
      <label class="input-group-text" for="bandeira-cartao">Bandeira</label>
    </div>
    <select class="custom-select" id="bandeira-cartao">
      <option disabled selected>Selecione uma opção...</option>
      <option value="master">MasterCard</option>
      <option value="visa">VISA</option>
      <option value="amex">American Express</option>
    </select>
  </div>
</div>
```



The image shows a visual representation of the Bootstrap form code shown in the previous blocks. It includes an email input field with an '@' icon, a CPF input field with a mask '000.000.000-00', a checkbox for 'Quero receber Newsletter da Mirror Fashion', a credit card number input field, and a dropdown menu for the card brand (Bandeira) with options for MasterCard, VISA, and American Express.

Implemente também em outros campos, inclusive usando ícones. Veja mais opções do Bootstrap para formulários:

<https://getbootstrap.com.br/docs/4.1/components/forms/>

Validação HTML5

- Há no HTML5 uma parte de validação de dados com restrições expressas diretamente no código HTML

required

- Indica na marcação do formulário quando um campo é de preenchimento obrigatório

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

- Esse atributo permite uma validação fraca no lado do cliente

pattern

- Pode-se especificar um formato requerido através do atributo pattern, adicionando uma expressão regular como valor:

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

O atributo pattern também permite uma validação fraca do campo

Validação no CSS

- Maioria dos novos componentes de formulário e atributos que agem como validadores de campos na verdade aplicam uma pseudo-classe (:invalid) no campo que não está atendendo ao requisito especificado
- Pode ser utilizada para dar um retorno visual imediato caso o usuário não esteja atendendo aos requisitos dos campos do formulário

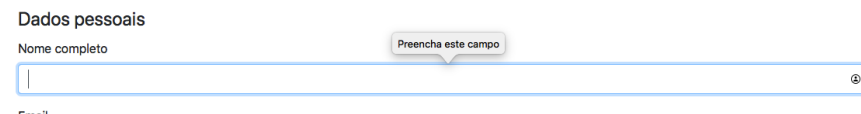
```
:invalid {  
  outline: 1px solid #CC0000;  
}
```

- Validação é fraca
- Não é possível impedir que o usuário envie as informações do formulário
- Mas é possível alterar o botão de submit e deixá-lo desabilitado caso seja possível selecionar algum elemento por essa pseudo-classe no formulário

Atividade - Validação com HTML5

1. Adicione o atributo required nos campos Nome e CPF.

Teste submeter o formulário sem preencher esses campos.

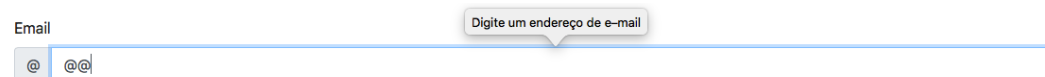


Dados pessoais

Nome completo

Email

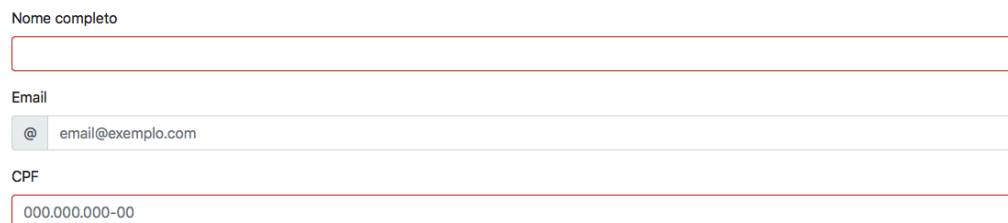
2. Algumas validações já são implícitas apenas por usarmos o input type correto. Por exemplo, tente submeter o formulário preenchendo o email com um valor inválido (com dois @ por exemplo).



Email

3. Podemos estilizar no CSS quando um campo está inválido:

```
.form-control:invalid {  
  border: 1px solid #CC0000;  
}
```



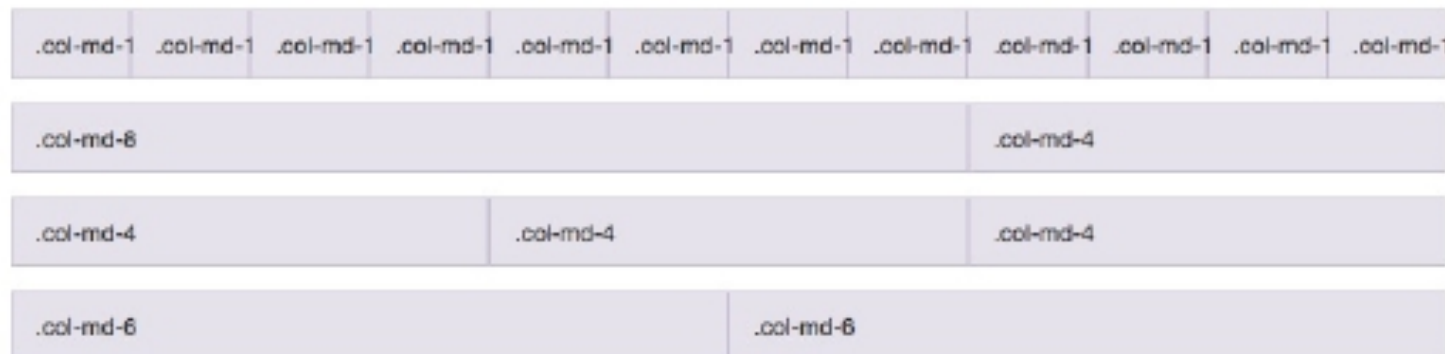
Nome completo

Email

CPF

Grid Responsivo do Bootstrap

- Dificuldade mais comum de um projeto front-end é o posicionamento de elementos, Principalmente em designs multi coluna.
- Solução mais comum é uso de grids
- Divide-se a tela em colunas e encaixa os elementos dentro desse grid
- Todo framework CSS moderno traz um grid pronto para utilização.
- Todo código CSS necessário para correto posicionamento já está escrito, apenas necessitando usar as classes certas.
- O Bootstrap tem um grid pronto e várias classes
- Trabalha com a ideia de 12 colunas mas podendo escolher quantas colunas ocupar



Grid Responsivo do Bootstrap

Essas classes de coluna definem o tamanho de cada elemento na página com base nas 12 partes do grid padrão

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

No código anterior, a primeira <div> ocupa 4/12 da tela e a outra, 8/12. Para o grid funcionar, ao redor das colunas usa uma <div> com a classe row, necessária para que o layout fique correto.

Ainda é possível criar grids dentro de outro grid, sempre obedecendo a divisão de 12 colunas

```
<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

Grid Responsivo do Bootstrap

- Grids são responsivos.
- Pode-se aplicar diferentes layouts de colunas no código ao mesmo tempo e cada um deles vai valer só em determinada situação.
- Nos códigos anteriores foram usadas classes como col-md-6. O md significa que ocupará 6 colunas do grid apenas em telas maiores que 768px de largura.
- Em telas menores, automaticamente o grid será de uma coluna só
- Existem classes pra outros tamanhos de tela
- No Bootstrap existem famílias prontas de classes de grids
 - col- : Extra small < 576px
 - col-sm- : Small >= 576px
 - col-md- : Medium >= 768px
 - col-lg- : Large >= 992px
 - col-xl- : Extra large >= 1200px

É possível 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, o grid divide no meio (6 pra cada lado) em telas muito pequenas mas depois divide em 4 e 8 para telas um pouco maiores.

Atividades - Grids

1. O design pode funcionar na tela de trabalho. Mas aumentando o tela, tudo pode ficar meio grande.

O card e o form esticam 100%, o que é um exagero em telas maiores.

Vamos usar grids do Bootstrap para transformar nosso design em 2 colunas em telas maiores.

Por padrão, o Bootstrap já traz media queries para adaptação em 768px. A ideia é deixar o card ocupar 4/12 e o form ocupar 8/12.

São três alterações necessárias:

- Criar uma `<div>` com classe `row` dentro do `container`;
- Criar uma `<div>` com classe `col-md-4` ao redor do cartão;
- Aplicar a classe `col-md-8` no formulário.

Faça essas alterações e cuidado com o resultado final e os milhões de divs misturados.

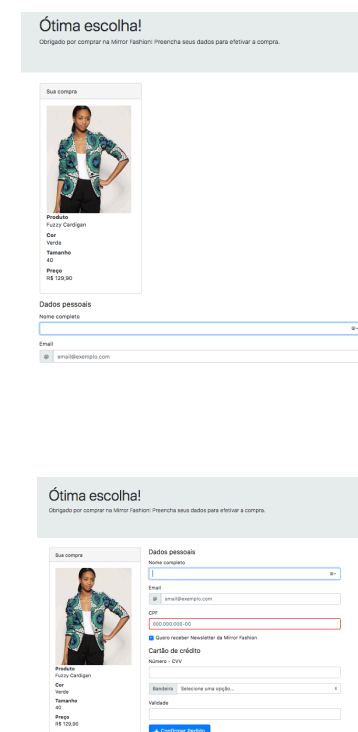
O código deve ficar mais ou menos assim:

```
<div class="container">
  <div class="row">

    <div class="col-md-4">
      <div class="card mb-3">
        <!-- ... cartão todo aqui ... -->
      </div>
    </div>

    <form class="col-md-8">
      <!-- ... todos os campos aqui ... -->
    </form>

  </div> <!-- fim .row -->
</div> <!-- fim .container -->
```



Teste a página para ver o resultado.

Atividades - Grids

2. Repare que o Bootstrap ajusta várias coisas responsivamente de forma automática.

Além de aplicar as classes do grid, repare como os tamanhos e fontes aumentam de acordo com a resolução, sem necessidade de fazer nada.

Faça os testes.

3. Quando aumentamos bastante a tela, tudo ainda se ajusta na proporção de 4 pra 8 que definimos. Mas o formulário fica grande demais. Em telas maiores, talvez seja legal deixar o formulário em 2 colunas.


Vamos usar outras classes do grid do Bootstrap que se aplicam em layouts maiores que 992px. Vamos dividir o formulário em 2 partes iguais, ou seja 6/12 (lembre que o grid do Bootstrap tem 12 partes como base). Conseguimos isso tudo usando a classe col-lg-6.

As mudanças necessárias são:

- Crie uma <div> com classe row ao redor dos 2 fieldsets;
- Aplique a classe col-lg-6 em cada um dos fieldsets.
- No final, a estrutura deve estar parecida com essa:

```
<form ...>
  <div class="row">
    <fieldset class="col-lg-6">
      ...
    </fieldset>
    <fieldset class="col-lg-6">
      ...
    </fieldset>
  </div>
  <button ...>
</form>
```

Ótima escolha!
Obrigado por comprar na Mirror Fashion! Preencha seus dados para efetivar a compra.

Sua compra	Dados pessoais	Cartão de crédito
	Nome completo <input type="text"/>	Número - CVV <input type="text"/>
	Email <input type="text" value="email@exemplo.com"/>	Bandeira Selecione uma opção. ▾
	CPF <input type="text" value="000.000.000-00"/>	Validade <input type="text"/>
	<input checked="" type="checkbox"/> Quero receber Newsletter da Mirror Fashion	
	Confirmar Pedido	
Produto Fuzzy Cardigan Cor Verde Tamanho 40 Preço R\$ 129,90		

Teste a página e redimensione para um tamanho em torno de 992px pra ver o resultado.

Atividades - Grids

4. (opcional) É possível usar mais de uma classe de grid ao mesmo tempo no mesmo elemento.

Por exemplo: dividimos a tela em 4/12 para o cartão e 8/12 para o formulário.

Mas se, em telas maiores, você quiser mudar essa proporção para 3/12 e 9/12, basta adicionar as classes `col-xl-3` e `col-xl-9` em conjunto as que tínhamos antes.

Implemente essa mudança no projeto.

Exemplo:

```
<div class="col-md-4 col-xl-3">
```

```
...
```

```
<form class="col-md-8 col-xl-9">
```

A série `col-xl-` aplica em resoluções acima de 1200px.

Para saber mais sobre os grids do Bootstrap:

<https://getbootstrap.com.br/docs/4.1/layout/grid/>

5. (opcional) Além de alterar o grid nas diferentes resoluções, o Bootstrap também permite esconder/exibir certos elementos apenas em uma resolução específica.

Por exemplo: imagine que, para otimizar o espaço pequeno no design para smartphone, vamos esconder a imagem do produto. Podemos fazer isso adicionando as classes `d-none` e `d-sm-block` na ``.

A classe `d-none` esconde o elemento independente da resolução da tela. E a classe `d-sm-block` faz com que o elemento apareça em telas com resoluções maiores que 576px.

Para saber mais sobre as classes auxiliares para responsivo do Bootstrap:

<https://getbootstrap.com.br/docs/4.1/extend/approach/#responsive>

Componentes JavaScript do Bootstrap

- Existem componentes CSS puro do Bootstrap como card e jumbotron
- Também existem componentes mais avançados que envolvem interatividade e JavaScript
- Há muito material disponível por padrão no Bootstrap
- Componentes como janela modal, galeria de imagens, dropdowns, menus de navegação, etc.

<https://getbootstrap.com.br/docs/4.1/getting-started/javascript/>

- Na atividade, será utilizado o menu superior (navbar)

Atividade - NavBar e JavaScript

1. Um componente muito famoso do Bootstrap é seu menu superior, chamado de navbar. O HTML é um pouco mais complexo pois se trata de um menu completo, mas é relativamente fácil.

Implemente um navbar na página, acima do jumbotron, logo no topo da página:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="index.html">Mirror Fashion</a>

  <ul class="navbar-nav">
    <li class="nav-item active">
      <a class="nav-link" href="sobre.html">Sobre</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Ajuda</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Perguntas frequentes</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Entre em contato</a>
    </li>
  </ul>
</nav>
```

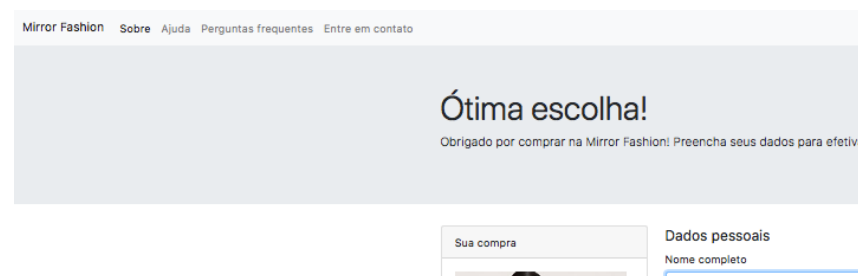
Teste o resultado no navegador.

2. Teste o menu em resoluções menores.

Note que o Bootstrap ajusta automaticamente o navbar em telas menores.

Por padrão, o comportamento é mudar o menu de horizontal para vertical em mobile.

Veja esse comportamento redimensionando o browser.



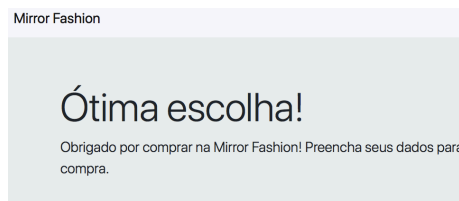
Atividade - NavBar e JavaScript

3. Uma outra solução para menus em telas pequenas é de juntar as opções em uma espécie de dropdown que só abre quando ativado. Isso é, criar um botão para ativar o menu (geralmente com o famoso ícone do sanduíche).

É bem simples fazer isso com Bootstrap, a funcionalidade está toda pronta.

Para fazer o menu colapsar em telas pequenas, basta adicionar uma `<div>` nova em volta do `` e 2 classes nessa `<div>` : a `collapse` e a `navbar-collapse`.

```
<div class="collapse navbar-collapse">
  <ul class="navbar-nav">
    ...
  </ul>
</div>
```



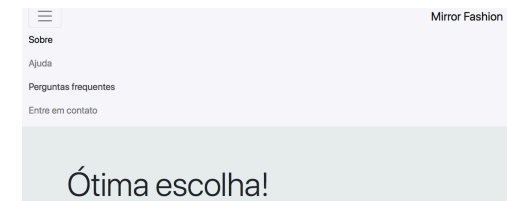
Se você testar agora, vai notar que o menu some nas telas menores. Para exibi-lo, precisamos fazer o próximo passo: criar o ícone que ativa o menu.

Dentro do navbar , logo acima do `<a>` , crie um botão de ativação. Dentro do botão crie um `` com a classe `navbar-toggler-icon`. Esse `` será o nosso ícone hambúrguer.

Para fazer a ligação entre o botão que vamos usar para colapsar a nossa `<div>` e a botão em si, use um `id` na `<div>` .

No caso usamos um `id navbarToggleExternalContent` na `<div>` e um `data-target="#navbarToggleExternalContent"` no button, para indicar que esse button aponta para o `id navbarToggleExternalContent`.

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <button class="navbar-toggler" type="button"
    data-toggle="collapse" data-target="#navbarToggleExternalContent">
    <span class="navbar-toggler-icon"></span>
  </button>
  <a class="navbar-brand" href="index.html">Mirror Fashion</a>
  <div class="collapse navbar-collapse" id="navbarToggleExternalContent">
    <ul class="navbar-nav">
      <li class="nav-item active">
        <a class="nav-link" href="sobre.html">Sobre</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Ajuda</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Perguntas frequentes</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Entre em contato</a>
      </li>
    </ul>
  </div>
</nav>
```



Se testar agora, vai notar que o menu aparece mas não funciona quando clicado. É porque essa funcionalidade no Bootstrap é implementada com JavaScript.

A boa notícia é que não precisamos escrever uma linha de código JS sequer, mas para tudo funcionar precisamos adicionar o JavaScript do Bootstrap.

No fim da página, logo antes de fechar o `</body>` , chame o arquivo do Bootstrap e do jQuery:

```
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/bootstrap.js"></script>
```

Teste novamente e veja o plugin funcionando. Usamos o JavaScript do Bootstrap implicitamente.

Atividade - NavBar e JavaScript

4. Há muitas opções possíveis para o navbar.

Por exemplo, podemos inverter as cores e usar um esquema mais escuro apenas trocando as classes navbar-light e bg-light pelas classes navbardark e bg-dark.

Para saber mais sobre o navbar: <https://getbootstrap.com.br/docs/4.1/components/navbar/>

5. Adicione a classe fixed-top dentro da tag <nav> . Repare que o menu fica fixo no topo mesmo com scroll.

Você talvez queira aplicar um padding-top no body pro conteúdo não ficar por baixo do navbar:

```
body {  
  padding-top: 55px;  
}
```

