

	<p>Bacharelado em Ciência da Computação Disciplina: Programação para Web I Professor: Dr. Demetrio Gomes Mestre Período: 2025.1</p>
<p>Nome do Aluno:</p>	

Prova I - Estilização do Projeto de Página de E-commerce

Desenvolva a estilização do projeto de página de e-commerce, seguindo os passos estabelecidos a seguir.

Etapa 1: Crie uma pasta chamada “css” no projeto.

- **Passo 1:** abra e copie o código do normalize.css para a nova pasta criada “css”.
a. [link para o normalize](#).
- **Passo 2:** crie o arquivo style.css dentro da nova pasta criada “css”.
- **Passo 3:** estabeleça a ligação entre as páginas do projeto e as novas folhas de estilos criadas nos passos 1 e 2.

Etapa 2: Defina as seguintes regras no style.css. Procure perceber o que mudou na renderização da página index.html com a adição desta folha de estilo.

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

header {
  background-color: #333;
  color: white;
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 1rem;
}

header h1 {
  margin: 0;
}
```

```
nav a {
  color: white;
  margin: 0 1rem;
  text-decoration: none;
}

.carrinho {
  position: relative;
  cursor: pointer;
}

.carrinho img {
  width: 30px;
  height: 30px;
}
```

```
#carrinho-contagem {  
  position: absolute;  
  top: -10px;  
  right: -10px;  
  background-color: red;  
  color: white;  
  border-radius: 50%;  
  padding: 5px 10px;  
}
```

```
.produtos {  
  display: flex;  
}
```

```
.produto-card {  
  background-color: #f9f9f9;  
  border: 1px solid #ddd;  
  border-radius: 5px;  
  padding: 1rem;  
  text-align: center;
```

```
.produto-card img {  
  height: auto;  
  max-height: 120px;  
  object-fit: contain;  
}
```

```
.produto-card h2 {  
  margin: 0.5rem 0;  
}
```

```
.produto-card p {  
  margin: 0.5rem 0;  
}
```

```
.produto-card button {  
  background-color: #007bff;  
  color: white;  
  border: none;  
  padding: 0.5rem 1rem;  
  cursor: pointer;  
  transition: background-color 0.3s;  
}
```

```
.produto-card button:hover {  
  background-color: #0056b3;  
}
```

```
footer {  
  background-color: #333;  
  color: white;  
  text-align: center;  
  padding: 1rem;  
  position: fixed;  
  bottom: 0;  
  width: 100%;  
}
```

Etapa 3: As imagens não estão aparecendo na página principal nem o código Javascript está sendo carregado. Faça o ajuste apropriado para que as imagens dos produtos apareçam e o código javascript seja carregado, sem alterar a estrutura organizacional dos arquivos do projeto.

Etapa 4: Note que o estilo da classe produtos está definido com display flex. Modifique para que o display da classe produtos no style.css seja definido como estilo grid, mas que ao invés de uma só coluna, haja múltiplas colunas e, que ao ser redimensionada a tela, o grid continue tendo características responsivas. O tamanho de cada card deve ter no mínimo 200 px e no máximo 1fr (use a função *minmax* para aplicar essa definição). Além disso, a propriedade gap deve ser definido para “1 rem” e *padding* para “2 rem”.

Dica: use a propriedade *grid-template-columns* utilizando a função *repeat* para aplicar a regra mesma regra de dimensionamento a todos os cards dentro do grid.

Lembrete: A medida **rem** é o **root font-size**, ou seja, 1rem é igual ao tamanho total do *font-size* do elemento raiz e, na ausência desse, o tamanho inicial da propriedade *font-size*.

Etapa 5: Crie um novo arquivo de estilos, nomeado como styleCarrinho.css, na pasta css e defina as seguintes regras no arquivo styleCarrinho.css. Faça a vinculação de styleCarrinho.css ao carrinho.html. Procure perceber o que mudou na renderização da página carrinho.html com a adição desta folha de estilo.

```
.carrinho-itens {
  padding: 2rem;
}

.carrinho-item {
  display: flex;
  justify-content: space-between;
  align-items: center;
  background-color: #f9f9f9;
  border: 1px solid #ddd;
  padding: 1rem;
  margin-bottom: 1rem;
}

.carrinho-item img {
  width: 50px;
  height: 50px;
  margin-right: 1rem;
}
```

```
.carrinho-item-detahes {
  flex: 1;
}

.carrinho-item-detahes h3 {
  margin: 0;
}

.carrinho-item-detahes p {
  margin: 0.5rem 0;
}

.carrinho-item button {
  background-color: #ff0000;
  color: white;
  border: none;
  padding: 0.5rem 1rem;
  cursor: pointer;
  transition: background-color 0.3s;
}
```

```

.carrinho-item button:hover {
  background-color: #cc0000;
}

.carrinho-total {
  text-align: right;
  margin-top: 2rem;
}

button {
  background-color: #007bff;
  color: white;
  border: none;
  padding: 0.5rem 1rem;
  cursor: pointer;
  transition: background-color 0.3s;
  margin-top: 1rem;
}

button:hover {
  background-color: #0056b3;
}

```

Etapa 6: Opa! Quando o carrinho tem muitos produtos, o botão Limpar Carrinho fica escondido por trás do rodapé. Modifique as regras de CSS para não permitir que isso não aconteça.

Etapa 7: Pense em algo para adicionar no *article* dos produtos notebook e tablet alguma sinalização para mostrar que estes dois produtos estão em promoção.

Etapa 8: Adicione um formulário para realizar pagamentos na página carrinho.html. O formulário deve conter os seguintes campos: bandeira do cartão, nome do titular, número do cartão, data de validade do cartão e dígito verificador do cartão (utilizando o campo protegido de senha).

Etapa 9: Modifique as definições dos elementos que possuem classes definidas e o seletores das regras de css para o projeto utilize o método BEM (Block, Element e Modifier) de organização de CSS.

Dica:

```

.block {
  display: block;
  width: 100%;
}

```

```

.block__element {
  font-size: 16px;
}

```

```

.block__element--modifier {
  color: #ff0000;
}

```