

# Exercícios de Front-End

## Exercício 1: Transformar o currículo em uma página da web

**Objetivo:** Criar uma página HTML simples para representar o seu currículo. O foco é aprender HTML e CSS básicos.

### 1. Descrição:

- a. Pegue o seu currículo (em formato texto ou PDF) e transforme-o em uma página web estática.
- b. A página deve incluir:
  - i. Um título com o seu nome.
  - ii. Uma seção para informações pessoais (nome, e-mail, telefone).
  - iii. Uma seção para experiências profissionais.
  - iv. Uma seção para formação acadêmica.
  - v. Uma seção para habilidades.
- c. Adicione estilos básicos com CSS para melhorar a aparência.

### 2. Exemplo de estrutura básica:

Unset

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Currículo</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      line-height: 1.6;
    }
    .section {
      margin-bottom: 20px;
    }
  </style>
</head>
<body>
  <h1>Seu Nome</h1>
```

```
<div class="section">
  <h2>Informações Pessoais</h2>
  <p>Email: seuemail@email.com</p>
  <p>Telefone: (XX) XXXXX-XXXX</p>
</div>
<div class="section">
  <h2>Experiências Profissionais</h2>
  <p>...</p>
</div>
<div class="section">
  <h2>Formação Acadêmica</h2>
  <p>...</p>
</div>
<div class="section">
  <h2>Habilidades</h2>
  <p>...</p>
</div>
</body>
</html>
```

## Exercício 2: Aplicar uma biblioteca de JS por CDN

**Objetivo:** Ensinar como usar bibliotecas de JavaScript via CDN e integrá-las ao currículo.

### 1. Descrição:

- Adicione interatividade à página do currículo criada no **Exercício 1**.
- Use uma biblioteca JS simples, como [Lottie](#), [Moment.js](#) ou [DataTables](#), via CDN.
- Exemplo: Adicione animações com Lottie ou formate datas com Moment.js.

## 2. Passo a passo:

- a. Escolha a biblioteca e adicione o script CDN no HTML:

Unset

```
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.29.4/moment.min.js"></script>
```

- b. Use a biblioteca no documento HTML para interagir com algum elemento.

## 3. Exemplo com Moment.js: Adicione a data formatada dinamicamente na página.

Unset

```
<div id="current-date"></div>  
<script>  
    document.getElementById("current-date").textContent =  
        "Data atual formatada: " + moment().format('DD/MM/YYYY');  
</script>
```

## 4. Exemplo com Lottie: Adicione uma animação JSON com Lottie:

Unset

```
<div id="lottie-container" style="width: 300px; height:  
300px;"></div>  
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/lottie-web/5.10.1/lottie.min.js"></script>  
<script>  
    var animation = lottie.loadAnimation({  
        container: document.getElementById('lottie-container'),  
        renderer: 'svg',  
        loop: true,  
        autoplay: true,  
        path:  
        'https://assets7.lottiefiles.com/packages/lf20_t9gkxhz4.json' //  
        URL do arquivo JSON da animação  
    });  
</script>
```

## 5. Exemplo com DataTables: Adicione uma tabela interativa usando DataTables:

Unset

```
<table id="example" class="display" style="width:100%">
  <thead>
    <tr>
      <th>Nome</th>
      <th>Idade</th>
      <th>Cargo</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>João</td>
      <td>30</td>
      <td>Desenvolvedor</td>
    </tr>
    <tr>
      <td>Maria</td>
      <td>25</td>
      <td>Designer</td>
    </tr>
  </tbody>
</table>
<script
src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script
src="https://cdn.datatables.net/1.13.5/js/jquery.dataTables.min.js"></script>
<link rel="stylesheet"
href="https://cdn.datatables.net/1.13.5/css/jquery.dataTables.min.css">
<script>
  $(document).ready(function() {
    $('#example').DataTable();
  });</script>
```

## Exercício 3: Criar uma galeria de imagens dinâmica

**Objetivo:** Ensiná-lo a manipular elementos HTML e adicionar conteúdo dinamicamente à página utilizando JavaScript.

### 1. Descrição:

- a. Crie uma galeria de imagens gerada dinamicamente a partir de um array de URLs de imagens.
- b. A página deve ter:
  - i. Um título.
  - ii. Um botão (ex: "Mostrar Galeria").
  - iii. Uma área vazia onde as imagens aparecerão.

### 2. Passo a passo:

- a. No JavaScript, crie um array com URLs de imagens (você pode usar URLs reais ou placeholders como <https://via.placeholder.com/150>).
- b. Quando o botão for clicado:
  - i. Use o JavaScript para adicionar cada imagem ao elemento HTML correspondente.
  - ii. As imagens devem ser estilizadas em forma de grade.

### 3. Dicas:

- a. Use `document.createElement` para criar elementos `<img>`.
- b. Use `appendChild` para adicioná-los à área vazia na página.
- c. Adicione classes ou estilos inline para organizar as imagens.

### 4. Estrutura básica da página:

Unset

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Galeria de Imagens</title>
  <style>
    .gallery {
      display: grid;
```

```

        grid-template-columns: repeat(auto-fill,
minmax(150px, 1fr));
        gap: 10px;
    }
    .gallery img {
        width: 100%;
        border: 1px solid #ddd;
        border-radius: 5px;
    }
</style>
</head>
<body>
    <h1>Minha Galeria de Imagens</h1>
    <button id="show-gallery">Mostrar Galeria</button>
    <div id="gallery" class="gallery"></div>
    <script>
        // Adicione o código aqui!
    </script>
</body>
</html>

```

## Exercício 4: Web Scraping com JS

**Objetivo:** Usar JavaScript para acessar e manipular elementos HTML, simulando um web scraping.

### 1. Descrição:

- a. Crie uma página com uma lista de links ou use uma página existente.
- b. Use o seguinte código para extrair links da página e salvá-los em um arquivo `.txt`:

JavaScript

```
let elements = // Pegue os elementos da página que contém os
links para outras partes da documentação na side-bar

let elementsArr = // Transforme a HTML Collection em um array.

let links = // Itere sobre os elementos do seu array extraindo o
href (link em sí) pesquise sobre métodos de array como map e
forEach.

let linksString = // Converta seu array pra string, pesquise
sobre JSON.stringify.

let blob = // Transforme sua string de links em um BLOB (Binary
Large Object) que vai representar o seu arquivo .txt armazenado
em memória.

let txtFileDownloadLink = // Crie um elemento <a></a> que
receberá o link de download pra sua blob.

txtFileDownloadLink.href = // adicione a URL pro download do seu
blob.

txtFileDownloadLink.download = // Nome do arquivo que será
baixado

txtFileDownloadLink.método; // Evento que simula um click no seu
link (nome do evento não é método, pesquise qual elemento simula
esse comportamento).
```

## 2. Passo a passo:

- a. Analise a página de documentação da [Jitterbit](#) procurando os elementos que contém links para outras partes do site na side-bar e encontre uma maneira de seleccionar o elemento ou HTML Collection (bem mais fácil do um elemento de cada vez).
- b. Explique o que cada parte do código faz.