

# SISTEMAS WEB

## A1

### Como funciona a programação para WEB

A programação para a web envolve o desenvolvimento de aplicativos e sites que são acessados por meio de navegadores da web. Para compreender o funcionamento de um sistema WEB, leia o texto abaixo.

- **Compreensão do Ambiente Web:**
  - Antes de começar a programar, é essencial entender o ambiente da web. Isso inclui o modelo cliente-servidor, o protocolo HTTP/HTTPS, a estrutura de URLs e o funcionamento básico dos navegadores.
- **Conhecimento de Tecnologias Básicas:**
  - A base da programação web inclui HTML (Hypertext Markup Language) para estruturação de conteúdo, CSS (Cascading Style Sheets) para estilização e JavaScript para interatividade do lado do cliente.
- **Configuração do Ambiente de Desenvolvimento:**
  - Configurar um ambiente de desenvolvimento que inclua um editor de código, como Visual Studio Code, e um navegador para testar o código. Ferramentas adicionais podem incluir Node.js e npm para gerenciamento de pacotes.
- **Desenvolvimento Front-end:**
  - **HTML:** Crie a estrutura básica da página web usando elementos HTML para definir cabeçalhos, parágrafos, listas, links, etc.
  - **CSS:** Adicione estilos à página usando CSS para controlar a aparência, layout e animações.
  - **JavaScript:** Integre interatividade e dinamismo à página usando JavaScript. Isso pode incluir manipulação do DOM (Document Object Model), requisições assíncronas (AJAX) e eventos.
- **Conceitos de Responsividade e Design Responsivo:**
  - Certifique-se de que seu design seja responsivo, ou seja, que se adapte a diferentes tamanhos de tela. Isso é vital para uma experiência do usuário consistente em dispositivos variados.
- **Desenvolvimento Back-end:**
  - Escolha uma linguagem de programação do lado do servidor (como Node.js, Python, **PHP**, Ruby) para processar solicitações do cliente e interagir com bancos de dados.
- **Persistência de Dados:**
  - Use bancos de dados (**MySQL**, PostgreSQL, MongoDB) para armazenar e recuperar informações necessárias para o funcionamento do aplicativo.
- **Implantação:**

- Escolha um serviço de hospedagem (Heroku, AWS, Netlify) para implantar seu aplicativo na web.
- **Testes e Depuração:**
  - Realize testes para garantir que o aplicativo funcione conforme o esperado. Use ferramentas de depuração para identificar e corrigir erros.
- **Otimização:**
  - Otimize o desempenho do seu aplicativo, tanto no lado do cliente quanto no servidor, para garantir tempos de carregamento rápidos e uma experiência eficiente.
- **Manutenção e Atualização:**
  - Mantenha-se atualizado com as melhores práticas, atualize bibliotecas e frameworks e resolva problemas de segurança conforme necessário.

Ao seguir esses passos, você construirá aplicativos web robustos e eficientes. É importante continuar aprendendo e adaptando-se às evoluções tecnológicas na área de desenvolvimento web.

## INTRODUÇÃO AO HTML5

O HTML5, a quinta iteração do padrão HTML, representa um avanço notável no campo do desenvolvimento web. Lançado em 2014 pelo World Wide Web Consortium (W3C) e pela Web Hypertext Application Technology Working Group (WHATWG), o HTML5 trouxe consigo uma série de recursos inovadores que transformaram a paisagem da criação de conteúdo online.

Uma das mudanças mais significativas foi a introdução de tags semânticas, como **<article>**, **<section>**, **<header>**, **<footer>**, que conferem uma estrutura mais significativa ao conteúdo. Essas tags não apenas tornam o código mais legível para os desenvolvedores, mas também melhoram a acessibilidade e a **interpretação dos motores de busca**.

O suporte nativo para áudio e vídeo através das tags **<audio>** e **<video>** foi um divisor de águas, eliminando a dependência de plugins externos. Isso possibilitou uma integração mais fluida de elementos multimídia nas páginas web, enriquecendo a experiência do usuário.

Além disso, o HTML5 trouxe consigo APIs poderosas que ampliaram as capacidades do desenvolvimento web. A API Canvas permitiu a criação de gráficos dinâmicos e animações, enquanto a API WebSockets facilitou a comunicação bidirecional em tempo real entre o navegador e o servidor.

A capacidade de **armazenamento local** através da **API Web Storage** e a execução offline via API de Cache ofereceram aos desenvolvedores ferramentas eficazes para aprimorar a eficiência e a experiência do usuário, especialmente em situações de conectividade intermitente.

Essas inovações, juntamente com a promoção de padronização e interoperabilidade entre navegadores, posicionaram o HTML5 como uma base sólida para o desenvolvimento web

moderno. À medida que a web continua a evoluir, o HTML5 continua a ser uma peça-chave na construção de experiências online envolventes e dinâmicas. Seja na criação de aplicativos interativos, sites responsivos ou conteúdo multimídia, o HTML5 desempenha um papel fundamental no impulsionamento da próxima geração de experiências na web.

## ESCREVENDO CÓDIGOS EM HTML5

O código básico do HTML consiste em uma estrutura simples que define o esqueleto de uma página web. Abaixo está um exemplo básico de um documento HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Minha Página Web</title>
</head>
<body>
  Seu texto aqui.
</body>
</html>
```

Vamos analisar cada parte:

- **<!DOCTYPE html>**: Esta declaração define a versão do HTML sendo utilizada, neste caso, o HTML5.
- **<html>**: Este elemento é a raiz do documento HTML e engloba todo o conteúdo da página.
- **<head>**: Esta seção contém metadados sobre o documento, como o título da página, que é definido pelo elemento **<title>**.
- **<title>**: Dentro da tag **<title>**, você pode inserir o título desejado para a sua página. No exemplo fornecido, o título é "Minha Página Web".
- **<body>**: Esta seção contém o conteúdo visível da página web. No exemplo, há um texto genérico "Seu texto aqui", que pode ser substituído pelo conteúdo desejado.

O código abaixo mostra um exemplo básico do uso do HTML com todas suas tags semânticas.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Minha Página Web</title>
</head>
<body>
  <header>
    <h1>Bem-vindo à Minha Página Web</h1>
  </header>
  <nav>
```

```

<ul>
  <li><a href="#section1">Seção 1</a></li>
  <li><a href="#section2">Seção 2</a></li>
  <li><a href="#section3">Seção 3</a></li>
</ul>
</nav>
<main>
  <section id="secao1">
    <h2>Seção 1</h2>
    <p>Conteúdo da seção 1.</p>
  </section>
  <section id="section2">
    <h2>Seção 2</h2>
    <p>Conteúdo da seção 2.</p>
  </section>
  <section id="section3">
    <h2>Seção 3</h2>
    <p>Conteúdo da seção 3.</p>
  </section>
</main>
<footer>
  <p>Rodapé da página &copy; 2024</p>
</footer>
</body>
</html>

```

Este é um exemplo básico que inclui os elementos essenciais de um documento HTML:

- **<!DOCTYPE html>**: Declara o tipo de documento como HTML5.
- **<html lang="pt-BR">**: Define a raiz do documento HTML com a especificação do idioma (neste caso, Português Brasil).
- **<head>**: Contém metadados, como o conjunto de caracteres usado e a viewport para dispositivos móveis.
- **<title>**: Define o título da página, que aparece na aba do navegador.
- **<body>**: Contém o conteúdo visível da página, como cabeçalhos, parágrafos, links, etc.
- **<header>**, **<nav>**, **<main>**, **<section>**, **<footer>**: Elementos estruturais que ajudam a organizar o conteúdo da página.

Este é um ponto de partida simples para criar uma página web. Conforme você avança no desenvolvimento, pode adicionar estilos (usando CSS), interatividade (usando JavaScript) e outros elementos conforme necessário para atender aos requisitos do seu projeto.

O código foi estilizado para diferenciar cada uma das partes da tela

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Minha Página Web</title>
  <style>
    body {
      background-color: #f0f0f0; /* Cor de fundo da página */
      margin: 0;
      padding: 0;
      font-family: Arial, sans-serif; /* Fonte padrão */
    }
    header {
      background-color: #3399ff; /* Cor de fundo do cabeçalho */
      color: #fff; /* Cor do texto do cabeçalho */
      padding: 20px;
      text-align: center;
    }
    nav {
      background-color: #66cc66; /* Cor de fundo da barra de navegação */
      padding: 10px;
    }
    nav ul {
      list-style-type: none; /* Remover marcadores de lista */
      padding: 0;
      margin: 0;
      text-align: center;
    }
    nav ul li {
      display: inline;
      margin-right: 20px;
    }
    nav ul li a {
      text-decoration: none; /* Remover sublinhado dos links */
      color: #333; /* Cor do texto dos links */
    }
    main {
      padding: 20px;
    }
    section {
      padding: 20px;
```

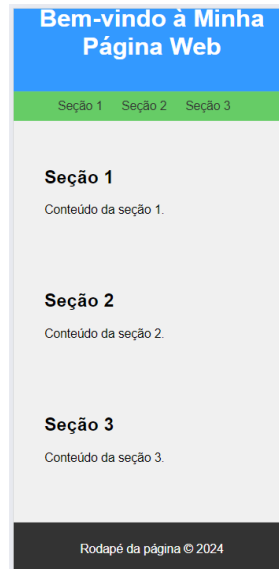
```

        margin-bottom: 20px;
    }
    footer {
        background-color: #333; /* Cor de fundo do rodapé */
        color: #fff; /* Cor do texto do rodapé */
        padding: 10px;
        text-align: center;
    }
</style>
</head>
<body>
    <header>
        <h1>Bem-vindo à Minha Página Web</h1>
    </header>
    <nav>
        <ul>
            <li><a href="#section1">Seção 1</a></li>
            <li><a href="#section2">Seção 2</a></li>
            <li><a href="#section3">Seção 3</a></li>
        </ul>
    </nav>
    <main>
        <section id="section1">
            <h2>Seção 1</h2>
            <p>Conteúdo da seção 1.</p>
        </section>
        <section id="section2">
            <h2>Seção 2</h2>
            <p>Conteúdo da seção 2.</p>
        </section>
        <section id="section3">
            <h2>Seção 3</h2>
            <p>Conteúdo da seção 3.</p>
        </section>
    </main>
    <footer>
        <p>Rodapé da página &copy; 2024</p>
    </footer>
</body>
</html>

```

**Renderizado no navegador**

**Renderizado para mobile**



## TAGS BÁSICAS

O HTML5 introduziu várias tags novas e aprimoradas para melhorar a semântica e oferecer mais flexibilidade ao desenvolvimento web. Abaixo estão algumas das tags básicas e importantes do HTML5:

### Estrutura Básica:

- **<html>**: Define o início e o fim do documento HTML.
- **<head>**: Contém metadados, como **<title>**, **<meta>**, e **<link>**.
- **<title>**: Define o título da página que aparece na aba do navegador.
- **<meta charset="UTF-8">**: Especifica o conjunto de caracteres UTF-8 para suporte internacional.
- **<meta name="viewport" content="width=device-width, initial-scale=1.0">**: Configura a viewport para dispositivos móveis.

### Seções:

- **<header>**: Define o cabeçalho da página ou de uma seção.
- **<nav>**: Representa a navegação, geralmente contendo links.
- **<main>**: Contém o conteúdo principal da página.
- **<section>**: Define uma seção genérica em um documento.
- **<article>**: Representa um conteúdo independente e autossuficiente, como uma postagem de blog.
- **<aside>**: Representa conteúdo relacionado, muitas vezes usado para barras laterais.
- **<footer>**: Define o rodapé da página ou de uma seção.

### Textos e Títulos:

- **<h1>** até **<h6>**: Define títulos de diferentes níveis de importância.
- **<p>**: Define um parágrafo.
- **<a>**: Cria um link.
- **<strong>** e **<em>**: Representam texto forte (negrito) e texto enfatizado (itálico), respectivamente.

### Listas:

- **<ul>**: Define uma lista não ordenada.
- **<ol>**: Define uma lista ordenada.
- **<li>**: Define um item de lista.

#### Multimídia:

- **<img>**: Incorpora uma imagem.
- **<audio>**: Incorpora conteúdo de áudio.
- **<video>**: Incorpora conteúdo de vídeo.

#### Formulários:

- **<form>**: Define um formulário.
- **<input>**: Define um campo de entrada.
- **<textarea>**: Define uma área de texto multilinha.
- **<button>**: Define um botão.

#### Tabelas:

- **<table>**: Define uma tabela.
- **<tr>**: Define uma linha em uma tabela.
- **<td>**: Define uma célula de dados em uma tabela.

Essas são apenas algumas das tags básicas do HTML5. Existem outras tags mais específicas e avançadas que oferecem funcionalidades adicionais e atendem a diferentes propósitos no desenvolvimento web.

## EXERCÍCIO 1

1. **<h1> até <h6>**: Utilize pelo menos três destas tags para organizar títulos de diferentes níveis na página. Por exemplo, **<h1>** para o título principal, **<h2>** para seções principais, e assim por diante.
2. **<p>**: Utilize esta tag para criar parágrafos descritivos sobre você, seus interesses, e projetos.
3. **<a>**: Crie links para suas redes sociais, como LinkedIn e GitHub. Certifique-se de incluir pelo menos dois links na página.
4. **<strong> e <em>**: Utilize essas tags para enfatizar palavras ou frases importantes no texto da página.
5. **<ul> e <li>**: Crie uma lista não ordenada que destaque suas habilidades ou tecnologias que você domina. Use a tag **<li>** para cada item da lista.
6. **<ol> e <li>**: Crie uma lista ordenada com pelo menos três projetos que você tenha realizado. Utilize a tag **<li>** para cada projeto.
7. **<img>**: Adicione imagens dos projetos mencionados na lista. Certifique-se de incluir a tag **<img>** com o atributo **src** apontando para a URL da imagem.
8. **<audio> e <video>**: Experimente adicionar elementos de áudio e vídeo à página, por exemplo, um clipe de áudio que destaque suas habilidades ou um vídeo de um projeto que você tenha concluído.

Como ficará a saída impressa na tela.



## Meu Portfólio

### Sobre Mim

Olá, eu sou [Seu Nome]. Sou um entusiasta de programação e estou apaixonado por desenvolvimento web.

Atualmente, estou explorando várias tecnologias e trabalhando em projetos emocionantes.

### Redes Sociais

Conecte-se comigo através das seguintes redes sociais:

- [LinkedIn](#)
- [GitHub](#)

### Habilidades

- HTML5
- CSS3
- JavaScript

### Projetos

## Projetos

#### 1. Projeto A

Descrição do Projeto A.



#### 2. Projeto B

Descrição do Projeto B.

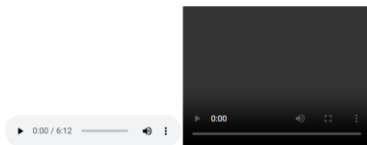


#### 3. Projeto C

Descrição do Projeto C.



### Midia



© 2024 Meu Portfólio

## DICA

No site <https://via.placeholder.com/150> você encontra essas imagens para diagramar sua página. O número 150 diz que a imagem deve ter 150x150 px e você pode alterar este valor para diagramar sua página web.

## Exercício 2

Crie uma página de receitas usando as seguintes instruções:

- **<h1> a <h3>:** Utilize essas tags para organizar os títulos na página. O **<h1>** pode ser o título principal da receita, **<h2>** para seções importantes e **<h3>** para subtítulos dentro das seções.
- **<p>:** Crie parágrafos explicativos sobre a receita, seus ingredientes e passos.
- **<a>:** Adicione um link para uma página de referência da receita ou para um vídeo de demonstração, se aplicável.
- **<strong> e <em>:** Utilize essas tags para destacar ingredientes importantes e etapas críticas na preparação.
- **<ul> e <li>:** Crie uma lista de ingredientes utilizando a tag **<ul>** e uma lista de passos utilizando a tag **<ol>**.

- **<img>**: Inclua imagens dos pratos ou dos ingredientes para ilustrar a receita.  
**Dica:** Você pode escolher uma receita favorita, seja de um prato principal, sobremesa, ou bebida. Certifique-se de incluir todos os elementos mencionados e personalize de acordo com suas preferências.

Lembre-se de manter a estrutura da página organizada, com títulos claros, instruções passo a passo e imagens relevantes. Este exercício permitirá que você pratique a utilização das tags HTML de maneira prática e contextualizada.

### Exercício 3

#### Criando uma Página de Livros Favoritos

Vamos praticar mais o uso das tags HTML para criar uma página que destaca seus livros favoritos. Siga as instruções abaixo:

- **<h1> a <h3>**: Utilize essas tags para organizar os títulos na página. O **<h1>** pode ser o título principal da página, **<h2>** para seções como "Ficção" e "Não Ficção", e **<h3>** para títulos de livros dentro de cada seção.
- **<p>**: Crie parágrafos explicativos sobre cada categoria de livros e, se desejar, adicione uma breve descrição sobre por que esses livros são seus favoritos.
- **<a>**: Inclua links para páginas de resenhas dos livros ou para suas páginas oficiais, se disponíveis.
- **<strong> e <em>**: Utilize essas tags para destacar autores, títulos de livros ou citações que considera marcantes.
- **<ul> e <li>**: Crie listas para os livros dentro de cada categoria. Pode ser uma lista de livros "Para Ler" e outra de livros "Já Lidos".
- **<img>**: Inclua capas de livros como imagens para ilustrar a página.  
**Dica:** Selecione diferentes gêneros de livros (ficção, não ficção, suspense, romance, etc.) e destaque alguns livros representativos de cada categoria. Adicione suas impressões pessoais para tornar a página mais envolvente.

Certifique-se de manter a estrutura da página clara e organizada, proporcionando uma experiência de navegação agradável para quem a acessa. Este exercício ajudará a consolidar o uso das tags HTML enquanto você cria uma página informativa sobre seus livros favoritos.

## A2

HTML5 oferece uma variedade de recursos poderosos para criar formulários interativos e estruturar conteúdo de maneira semântica. Este tutorial abordará a criação de formulários HTML, explorando elementos de entrada e destacando a importância da estrutura semântica com cabeçalhos, parágrafos e listas.

### Parte 1: Estrutura Semântica

## 1.1 Cabeçalhos:

Use as tags **<h1>** a **<h6>** para criar cabeçalhos significativos. Por exemplo:

```
<h1>Meu Formulário</h1>
```

```
<h2>Dados Pessoais</h2>
```

## 1.2 Parágrafos:

Utilize a tag **<p>** para estruturar o texto de forma clara e legível:

```
<p>Complete o formulário abaixo para se cadastrar.</p>
```

## 1.3 Listas:

Use **<ul>** para listas não ordenadas e **<ol>** para listas ordenadas. Exemplo:

```
<ul>
```

```
  <li>Nome</li>
```

```
  <li>Email</li>
```

```
</ul>
```

# Parte 2: Formulários HTML

## 2.1 Elemento <form>:

O elemento **<form>** é a estrutura principal para criar formulários. Adicione atributos como **action** (para o URL de envio) e **method** (GET ou POST).

```
<form action="/enviar" method="post">
```

```
  <!-- Seus elementos de entrada serão adicionados aqui -->
```

```
</form>
```

## 2.2 Elementos de Entrada:

Use tags como **<input>**, **<textarea>**, **<select>**, e outros para coletar dados. Exemplo:

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

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

## 2.3 Tipos de Entrada:

Explore diferentes tipos de entrada, como **text**, **email**, **password**, **checkbox**, **radio**, etc.

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

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

## Conclusão:

Neste tutorial, exploramos a criação de formulários HTML5, destacando elementos de entrada e estruturação semântica. Os exercícios práticos permitirão consolidar o aprendizado. Lembre-se de sempre praticar e experimentar para aprimorar suas habilidades.

## Exercício 1: Construção de Formulário:

- **Crie um Formulário de Contato:**
- Inclua campos para Nome, Email e Mensagem.

Estilização (Opcional):

Experimente adicionar estilos CSS para tornar os formulários visualmente atraentes e amigáveis.

```
form {  
  max-width: 400px;  
  margin: 0 auto;  
  padding: 20px;  
  background: #f4f4f4;  
}
```

## Exercício 2: Formulário de Contato Simples

Crie um formulário de contato simples com os campos:

- Nome (campo de texto)
- Email (campo de texto)
- Mensagem (área de texto)

## Exercício 3: Cadastro de Usuário

Elabore um formulário para cadastro de usuário com os campos:

- Nome (campo de texto)
- Email (campo de texto)
- Senha (campo de senha)
- Confirmação de Senha (campo de senha)

## Exercício 4: Escolha de Gênero

Utilize um **<select>** para criar um formulário que permita aos usuários escolher seu gênero. Inclua as opções:

- Masculino
- Feminino

## Exercício 5: Pesquisa de Satisfação

Crie um formulário de pesquisa de satisfação com perguntas sobre a experiência do usuário. Inclua tipos de entrada diferentes, como caixas de seleção e botões de rádio.

## Exercício 6: Formulário de Login

Desenvolva um formulário de login com os campos:

- Email (campo de texto)
- Senha (campo de senha)
- Lembrar-me (caixa de seleção)

Lembre-se de utilizar rótulos apropriados e estruturar semanticamente os elementos. Este conjunto de exercícios fornecerá uma base sólida para a prática na criação de formulários HTML.

## A3

# Introdução ao CSS - Conceitos Básicos de Estilização

### Introdução:

CSS (Cascading Style Sheets) é uma linguagem de estilização usada para controlar a apresentação visual de páginas web. Este tutorial abordará os conceitos básicos de CSS, incluindo seletores, propriedades e valores, seguido por exercícios práticos para estilizar uma página HTML.

## Parte 1: Conceitos Básicos

### 1.1 Seletores:

Os seletores são padrões que correspondem a elementos HTML que você deseja estilizar. Exemplos comuns incluem:

- Seletor de Tipo: **p** (estiliza todos os parágrafos)
- Seletor de Classe: **.classe** (estiliza elementos com a classe especificada)
- Seletor de ID: **#id** (estiliza o elemento com o ID especificado)

### 1.2 Propriedades e Valores:

As propriedades são características a serem estilizadas, e os valores são as configurações específicas. Exemplos:

- **color: blue;** (define a cor do texto para azul)
- **font-size: 16px;** (define o tamanho da fonte como 16 pixels)
- **background-color: #f0f0f0;** (define a cor de fundo como cinza claro)

## Exercício 1: Estilização de Texto:

Crie um arquivo HTML simples com um parágrafo e estilize o texto. Exemplo de CSS:

```
/* Estiliza todos os parágrafos */
p {
  color: green;
  font-size: 18px;
```

```
font-family: 'Arial', sans-serif;
}
```

### Estilização de Botões:

Adicione um botão à sua página HTML e estilize-o. Exemplo de CSS:

```
/* Estiliza todos os botões */
button {
  background-color: #3498db;
  color: #fff;
  padding: 10px 20px;
  font-size: 16px;
  border: none;
  border-radius: 5px;
}
```

### Layout da Página:

Explore a estilização do layout da página. Exemplo de CSS:

```
/* Estiliza o corpo da página */
body {
  margin: 0;
  padding: 0;
  font-family: 'Helvetica', sans-serif;
}
/* Estiliza o cabeçalho da página */
header {
  background-color: #333;
  color: #fff;
  padding: 10px;
  text-align: center;
}
```

## Exercício 2: Estilizando Títulos

Crie uma página HTML com diferentes títulos (h1 a h6). Utilize CSS para estilizar cada título da seguinte forma:

- h1: Cor vermelha (#ff0000), tamanho de fonte 30 pixels, negrito.
- h2: Cor azul (#0000ff), tamanho de fonte 24 pixels, itálico.
- h3: Cor verde (#008000), tamanho de fonte 20 pixels, sublinhado.
- h4: Cor roxa (#800080), tamanho de fonte 18 pixels, normal.
- h5: Cor laranja (#ffa500), tamanho de fonte 16 pixels, negrito e itálico.
- h6: Cor marrom (#8b4513), tamanho de fonte 14 pixels, normal.

### Exercício 3: Estilizando Links

Desenvolva uma página HTML com uma série de links (a). Utilize CSS para estilizar os links da seguinte forma:

- Cor azul (#3498db) para links normais.
- Cor vermelha (#e74c3c) quando o mouse estiver sobre os links.
- Remova sublinhado dos links.

### Exercício 4: Estilizando uma Lista

Crie uma lista não ordenada ou ordenada em uma página HTML. Utilize CSS para estilizar os itens da lista da seguinte forma:

- Cor verde (#008000) para o texto dos itens.
- Utilize um círculo como marcador.
- Adicione margem de 10 pixels entre os itens.

### Exercício 5: Estilizando Caixas de Texto

Elabore uma página HTML com caixas de texto (input type="text") e áreas de texto (textarea). Utilize CSS para estilizar esses elementos da seguinte forma:

- Bordas arredondadas.
- Cor de fundo cinza claro (#ecf0f1).
- Preenchimento interno de 8 pixels.
- Margem inferior de 10 pixels.

### Exercício 6: Estilizando Botões de Ação

Inclua botões (button) em sua página e utilize CSS para estilizá-los da seguinte forma:

- Cor de fundo verde (#2ecc71).
- Texto branco (fff).
- Preenchimento interno de 10 pixels verticalmente e 20 pixels horizontalmente.
- Borda arredondada de 3 pixels.
- Cor de fundo azul (#27ae60) quando o mouse estiver sobre o botão.

### Exercício 7: Estilizando Imagens

Adicione imagens à sua página HTML e utilize CSS para estilizá-las da seguinte forma:

- Borda sólida de 2 pixels com cor azul (#3498db).
- Sombra sutil de 2 pixels de deslocamento horizontal e vertical com cor de sombra cinza (#888888).

### Exercício 8: Estilizando uma Divisão de Conteúdo

Desenvolva uma divisão (div) de conteúdo em sua página HTML. Utilize CSS para estilizar essa divisão da seguinte forma:

- Cor de fundo cinza claro (#f0f0f0).
- Preenchimento interno de 15 pixels.
- Margem inferior de 15 pixels.

## Exercício 9: Estilizando um Formulário

Crie um formulário simples com campos de texto, áreas de texto e botões. Utilize CSS para estilizar o formulário da seguinte forma:

- Largura máxima de 400 pixels.
- Margem centralizada.
- Campos de texto, áreas de texto e botões com largura de 100%.
- Margem inferior de 10 pixels para os campos.

## Exercício 10: Estilizando uma Barra de Navegação

Implemente uma barra de navegação horizontal (nav) com links. Utilize CSS para estilizar a barra de navegação da seguinte forma:

- Cor de fundo preta (#333).
- Texto branco (#fff) para links normais.
- Cor de fundo mais escura (#555) quando o mouse estiver sobre os links.

## Exercício 11: Estilizando o Layout Geral

Estilize o layout geral de sua página HTML utilizando CSS da seguinte forma:

- Utilize a fonte 'Arial' como fonte principal para o corpo da página.
- Margem e preenchimento do corpo da página zerados.
- Adicione estilos específicos conforme a necessidade para criar uma aparência coesa e agradável.

## Entendendo o Flexbox: Um Guia Abrangente

O modelo de layout Flexbox (Flexible Box) é uma poderosa ferramenta em CSS para criar designs responsivos e flexíveis em páginas web. Ao contrário de modelos de layout mais antigos, como o modelo de bloco e o modelo de linha, o Flexbox simplifica a criação de layouts complexos e a distribuição de espaço entre os itens de forma intuitiva.

## Princípios Básicos do Flexbox

### 1. Container Flex:

O Flexbox opera dentro de um elemento pai, chamado de contêiner flexível (**display: flex;**). Esse contêiner torna seus filhos itens flexíveis, permitindo um controle mais preciso sobre o layout e distribuição do espaço.



## 2. Eixo Principal e Eixo Cruzado:

No Flexbox, há dois eixos principais: o eixo principal e o eixo cruzado. O eixo principal é definido pela propriedade **flex-direction** e determina a direção dos itens flexíveis. O eixo cruzado é perpendicular ao eixo principal.

## 3. Justificação e Alinhamento:

As propriedades **justify-content** e **align-items** controlam, respectivamente, a distribuição dos itens ao longo do eixo principal e cruzado. Essas propriedades oferecem opções como espaçamento uniforme, centralização e alinhamento aos extremos.

## Utilizando Flexbox para Layouts Simples

### 1. Container Flex Simples:

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

### 2. Distribuição no Eixo Principal:

```
.container {  
  
  display: flex;  
  
  justify-content: space-between;  
}
```

### 3. Alinhamento no Eixo Cruzado:

```
.container {  
  
  display: flex;  
  
  align-items: center;  
}
```

## Utilizando Flexbox para Layouts Complexos

### 1. Flexibilidade Dinâmica:

```
.item {  
  
  flex: 1;  
}
```

### 2. Ordem de Exibição:

```
.item:nth-child(2) {  
  
  order: 1;  
}
```

### 3. Layout em Linha ou Coluna:

```
.container {  
  
    flex-direction: column;  
  
}
```

## Vantagens do Flexbox

- **Responsividade Simples:** Facilita a criação de layouts que se adaptam a diferentes tamanhos de tela.
- **Alinhamento Fácil:** Permite alinhar elementos de maneira fácil e eficiente.
- **Reorganização Dinâmica:** Os itens podem ser reorganizados dinamicamente sem modificar a ordem no código HTML.
- **Distribuição Uniforme:** Controla espaçamento e distribuição de espaço automaticamente.

## Estilizando Imagens com CSS

Vamos agora usar as técnicas de estilização de imagens usando CSS. Vamos criar uma interface simples com uma imagem arredondada, acompanhada de um formulário centralizado que possui campos para o nome do usuário e senha.

### 1. Estrutura HTML Básica

Começamos criando o esqueleto HTML da nossa página. Vamos adicionar uma **<div>** para conter a imagem e o formulário.

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <link rel="stylesheet" href="styles.css">  
    <title>Estilizando Imagens</title>  
</head>  
<body>  
    <div class="container">  
          
        <form>  
            <label for="username">Nome de Usuário:</label>  
            <input type="text" id="username" name="username" required>  
            <label for="password">Senha:</label>  
            <input type="password" id="password" name="password" required>  
            <button type="submit">Login</button>
```

```
    </form>
  </div>
</body>
</html>
```

## 2. Estilizando formulário

Vamos agora criar o arquivo **styles.css** para adicionar os estilos.

```
body {
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
  margin: 0;
  font-family: 'Helvetica', Arial, sans-serif;
}
```

```
.container {
  text-align: center;
}
```

```
.rounded-image {
  border-radius: 50%;
  width: 150px;
  height: 150px;
  object-fit: cover;
}
```

```
form {
  margin-top: 20px;
  font-family: 'Helvetica', Arial, sans-serif;
}
```

```
form label {
  display: block;
  margin-bottom: 5px;
}
```

```
form input {
  width: 100%;
  padding: 8px;
  margin-bottom: 10px;
  box-sizing: border-box;
```

```
}
```

```
form button {  
  background-color: #3498db;  
  color: #fff;  
  padding: 10px 20px;  
  border: none;  
  border-radius: 3px;  
  cursor: pointer;  
}
```

```
form button:hover {  
  background-color: #2980b9;  
}
```

## Texto em colunas usando CSS

Vamos criar um exemplo simples de uma página dividida em duas colunas usando CSS. Utilizaremos o texto "Lorem Ipsum" como conteúdo fictício. Aqui está o código HTML (index.html):

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <link rel="stylesheet" href="styles-2.css">  
  <title>Página com Duas Colunas</title>  
</head>  
<body>  
  <div class="container">  
    <div class="column left">  
      <h2>Coluna Esquerda</h2>  
      <p>  
        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed ac libero ac turpis  
tincidunt euismod.  
        Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia  
Curae; Vivamus id  
        nisi at magna tincidunt gravida nec in dui. Quisque vel fermentum tellus. Sed  
bibendum urna vitae  
        libero facilisis, nec tincidunt metus cursus. Curabitur vel risus at dolor tincidunt  
scelerisque.  
      </p>  
    </div>
```

```

<div class="column right">
  <h2>Coluna Direita</h2>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed ac libero ac turpis
    tincidunt euismod.
    Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia
    Curae; Vivamus id
    nisi at magna tincidunt gravida nec in dui. Quisque vel fermentum tellus. Sed
    bibendum urna vitae
    libero facilisis, nec tincidunt metus cursus. Curabitur vel risus at dolor tincidunt
    scelerisque.
  </p>
</div>
</div>
</body>
</html>

```

Agora, o código CSS (**styles-2.css**):

```

body {
  font-family: 'Arial', sans-serif;
  margin: 0;
}
.container {
  display: flex;
  justify-content: space-around;
  padding: 20px;
}
.column {
  width: 45%;
  padding: 15px;
  border: 1px solid #ccc;
}
.left {
  order: 1; /* Muda a ordem de exibição no layout flexível */
}
.right {
  order: 2;
}

```

O arquivo de estilos (**styles.css**) é usado para definir o design e a aparência visual da página HTML. Vamos analisar cada parte do código CSS para entender suas funcionalidades:

```
body {  
  font-family: 'Arial', sans-serif;  
  margin: 0;  
}
```

- **body**: Este seletor aplica estilos ao elemento **<body>** da página.
- **font-family: 'Arial', sans-serif**:: Define a família de fontes para ser usada no corpo da página. Se Arial não estiver disponível, o navegador usará qualquer fonte sans-serif padrão do sistema.
- **margin: 0**:: Remove as margens padrão do corpo da página, garantindo um layout mais consistente.

```
.container {  
  display: flex;  
  justify-content: space-around;  
  padding: 20px;  
}
```

- **.container**: Este seletor aplica estilos à **<div>** com a classe "container".
- **display: flex**:: Usa o modelo de layout Flexbox, permitindo um fácil posicionamento e dimensionamento dos elementos filhos.
- **justify-content: space-around**:: Distribui o espaço em torno dos elementos filhos dentro do contêiner, proporcionando um espaçamento uniforme.
- **padding: 20px**:: Adiciona preenchimento interno ao contêiner para melhorar a estética.

```
.column {  
  width: 45%;  
  padding: 15px;  
  border: 1px solid #ccc;  
}
```

- **.column**: Este seletor aplica estilos a todas as **<div>** com a classe "column".
- **width: 45%**:: Define a largura das colunas como 45% do contêiner, permitindo que ambas se ajustem na tela.
- **padding: 15px**:: Adiciona preenchimento interno às colunas para espaço interno.
- **border: 1px solid #ccc**:: Adiciona uma borda sólida de 1 pixel com a cor cinza claro para destacar visualmente as colunas.

```
.left {  
  order: 1;  
}
```

```
.right {  
  order: 2;  
}
```

- **.left** e **.right**: Estes seletores aplicam estilos às colunas esquerda e direita, respectivamente.

- **order:** Define a ordem de exibição das colunas no layout flexível. Neste caso, a coluna esquerda tem uma ordem de 1 e a coluna direita tem uma ordem de 2.

Em resumo, o arquivo de estilos define a fonte, o layout, o espaçamento e o visual das colunas em uma página HTML. Flexbox é utilizado para criar um layout de duas colunas responsivo, e diversas propriedades são aplicadas para estilizar cada componente da página.

## Exercícios