

# Centro Universitário Presidente Antônio Carlos Programação para Internet

## HTML e CSS

Felipe Roncalli de Paula Carneiro  
felipecarneiro@unipac.br

# O que vamos aprender nessa aula

- HTML
- Estrutura básica de uma página HTML
- Tags HTML mais comuns
- Atributos HTML mais comuns
- CSS
- Seletores
- Propriedades
- Estilos

# HTML

- Linguagem de **marcação** baseada em tags;
- Bloco de construção mais **básico** da web. Define o significado e a estrutura do conteúdo da web;
- Quando um **navegador** recebe uma página HTML, ele a interpreta e renderiza para que os usuários possam visualizá-la.

# HTML 5

Em resumo, HTML é a linguagem de marcação básica usada para criar páginas da web, enquanto HTML5 é a versão mais recente e avançada do HTML, oferecendo recursos e capacidades adicionais para criar experiências web mais ricas e interativas.

# HTML 5

- **Semântica Melhorada:** O HTML5 introduziu novas tags semânticas, como `<header>`, `<nav>`, `<main>`, `<article>`, `<section>` e `<footer>`, que ajudam a estruturar o conteúdo de forma mais clara e significativa.
- **Áudio e Vídeo Integrados:** Com as tags `<audio>` e `<video>`, o HTML5 permite incorporar áudio e vídeo diretamente nas páginas da web, eliminando a necessidade de plugins de terceiros.

# HTML 5

- **Canvas:** A tag `<canvas>` permite a criação de gráficos, animações e renderização de imagens em tempo real usando JavaScript.
- **Geolocalização:** A API de Geolocalização do HTML5 permite que os navegadores obtenham a localização do usuário, desde que a permissão seja concedida, o que pode ser útil para serviços baseados em localização.

# HTML 5

- **Armazenamento Local:** As APIs localStorage e sessionStorage permitem que os desenvolvedores armazenem dados localmente no navegador do usuário, permitindo a criação de aplicativos web mais avançados e interativos.
- **Formulários Avançados:** O HTML5 introduziu novos tipos de entrada de formulário, como `<input type="email">`, `<input type="url">`, `<input type="date">`, e muito mais, tornando a criação de formulários mais rica e amigável.

# HTML 5

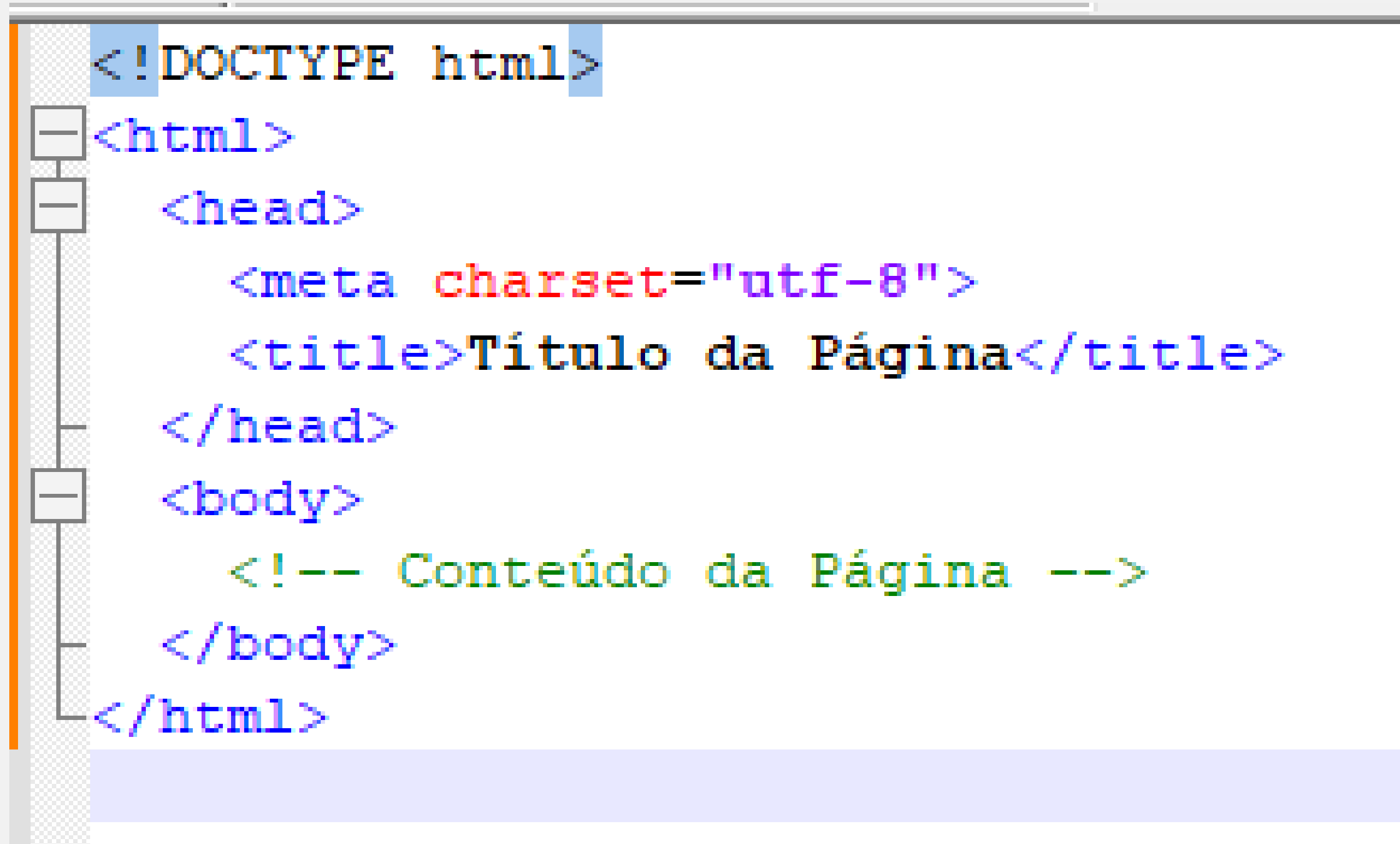
- **WebSockets:** A API de WebSockets permite comunicação bidirecional em tempo real entre o cliente e o servidor, possibilitando a criação de aplicativos web interativos e dinâmicos.
- **SVG:** Embora não seja exclusivo do HTML5, o suporte a SVG (Scalable Vector Graphics) foi aprimorado, permitindo a criação de gráficos vetoriais escaláveis diretamente no código HTML.



# HTML 5

- **Web Workers:** A API de Web Workers permite a execução de scripts em segundo plano, ajudando a melhorar a performance e a responsividade dos aplicativos web.
- **API de Arrastar e Soltar:** Com a API de Arrastar e Soltar, é possível criar interfaces mais intuitivas e interativas, permitindo que os usuários arrastem elementos na página.

# Estrutura básica de uma página HTML



# HTML

- **<!DOCTYPE html>**: Declaração do tipo de documento, que indica que o documento é uma página HTML5.
- **<html>**: Elemento raiz da página HTML.
- **<head>**: Elemento que contém informações sobre a página, como o título, metadados e referências a arquivos externos.
- **<meta charset="utf-8">**: Meta tag que especifica o conjunto de caracteres utilizado na página. O valor "utf-8" é recomendado para suportar diversos idiomas.
- **<title>**: Elemento que define o título da página, que é exibido na aba do navegador.
- **<body>**: Elemento que contém o conteúdo da página, como textos, imagens, links e formulários.

# HTML - Tags mais comuns

**<h1>,<h2>,<h3>,<h4>,<h5>,<h6>**: Definem títulos e subtítulos.

**<p>**: Define um parágrafo de texto.

**<a>**: Define um link para outra página ou recurso.

**<img>**: Define uma imagem para ser exibida na página.

**<ul> e <li>**: Define uma lista não ordenada.

**<ol> e <li>**: Define uma lista ordenada.

**<table>, <thead>, <tbody>, <tr>, <th>, <td>**: Define uma tabela e suas células.

**<form>, <input>, <button>, <select>, <option>, <textarea>**: Define um formulário e seus campos.

**<div> e <span>**: São usados para agrupar elementos e aplicar estilos CSS.

# HTML - Atributos mais comuns

- **id**: Define um identificador único para um elemento, permitindo que ele seja estilizado ou manipulado via JavaScript.
- **class**: Define uma classe para um elemento, permitindo que você aplique estilos a vários elementos usando a mesma classe.
- **src**: Especifica o URL da origem de conteúdo, como imagens ou scripts.
- **href**: Define o URL de destino para links, âncoras e áreas de mapa de imagem.
- **alt**: Fornece um texto alternativo para imagens, útil para acessibilidade e quando a imagem não pode ser exibida.
- **title**: Define um texto de dica de ferramenta que aparece quando o cursor paira sobre um elemento.

# HTML - Atributos mais comuns

- **style:** Permite a aplicação de estilos CSS inline a um elemento.
- **width e height:** Especificam as dimensões de uma imagem ou de um elemento.
- **disabled:** Desativa a interação com um elemento, como botões ou campos de entrada.
- **value:** Define o valor de um elemento interativo, como um campo de entrada ou um botão.
- **placeholder:** Exibe um texto temporário em um campo de entrada antes que o usuário insira dados.
- **required:** Indica que um campo de entrada deve ser preenchido antes de enviar um formulário.

# HTML - Atributos mais comuns

- **checked:** Marca uma caixa de seleção ou um botão de opção como selecionado por padrão.
- **readonly:** Impede a edição de campos de entrada, permitindo apenas leitura.
- **target:** Define onde o link deve ser aberto, como em uma nova janela (\_blank) ou na mesma janela (\_self).
- **alt:** Especifica texto alternativo para elementos de mídia, como imagens e áudio, para fins de acessibilidade.
- **aria-\*:** Atributos de acessibilidade usados para fornecer informações adicionais para leitores de tela.

# Exercícios - HTML

Crie um formulário de cadastro de cliente com os campos apropriados: Nome, Telefone, Data de Nascimento, Rua, Bairro, Cidade, Estado, CEP, E-mail e um botão "Enviar". Quando o usuário clicar no botão "Enviar", uma mensagem de sucesso deve aparecer indicando que o cadastro foi realizado com sucesso.



# CSS

- CSS (Cascading Style Sheets), em português "Folhas de Estilo em Cascata", é uma linguagem de estilo utilizada para definir a apresentação e formatação de documentos HTML e XML.
- Ela permite separar o conteúdo da estrutura e do estilo de um documento, o que ajuda na criação de layouts atraentes e consistentes para páginas web.

# CSS - Principais Características

- **Seletor:** Define os elementos HTML aos quais a regra de estilo se aplicará.
- **Propriedades:** Descrevem como os elementos selecionados devem ser exibidos, como cores, fontes, margens, tamanhos, entre outros.
- **Valor:** Especifica o valor de uma propriedade, como uma cor específica, tamanho da fonte, etc.
- **Cascata:** Refere-se à maneira como as regras de estilo são aplicadas e podem ser anuladas por outras regras de estilo.
- **Especificidade:** Define a prioridade das regras quando múltiplas regras conflitam.

# CSS - Principais Características

- **Herança:** Alguns estilos definidos em um elemento podem ser herdados por seus elementos filhos.
- **Box Model:** Descreve o dimensionamento e o posicionamento dos elementos HTML, incluindo margens, bordas e preenchimento.
- **Media Queries:** Permite a adaptação da aparência do site a diferentes dispositivos e tamanhos de tela.
- **Animações e Transições:** CSS permite criar animações e transições suaves entre os estados dos elementos.

# CSS - Seletores

- **Seletor de Tipo:** Este é o seletor mais básico e simples. Ele seleciona todos os elementos de um determinado tipo. Por exemplo:

```
1 p {  
2     color: blue;  
3 }
```

Neste caso, todos os parágrafos (<p>) terão o texto em azul.

# CSS - Seletores

- **Seletor de Classe:** Este seletor seleciona elementos que possuem uma determinada classe. Classes são definidas usando o atributo class no HTML e permitem que você estilize vários elementos de maneira semelhante.

```
5  .destaque {  
6      background-color: yellow;  
7  }
```

Isso aplicará um fundo amarelo a todos os elementos com a classe "destaque".

# CSS - Seletores

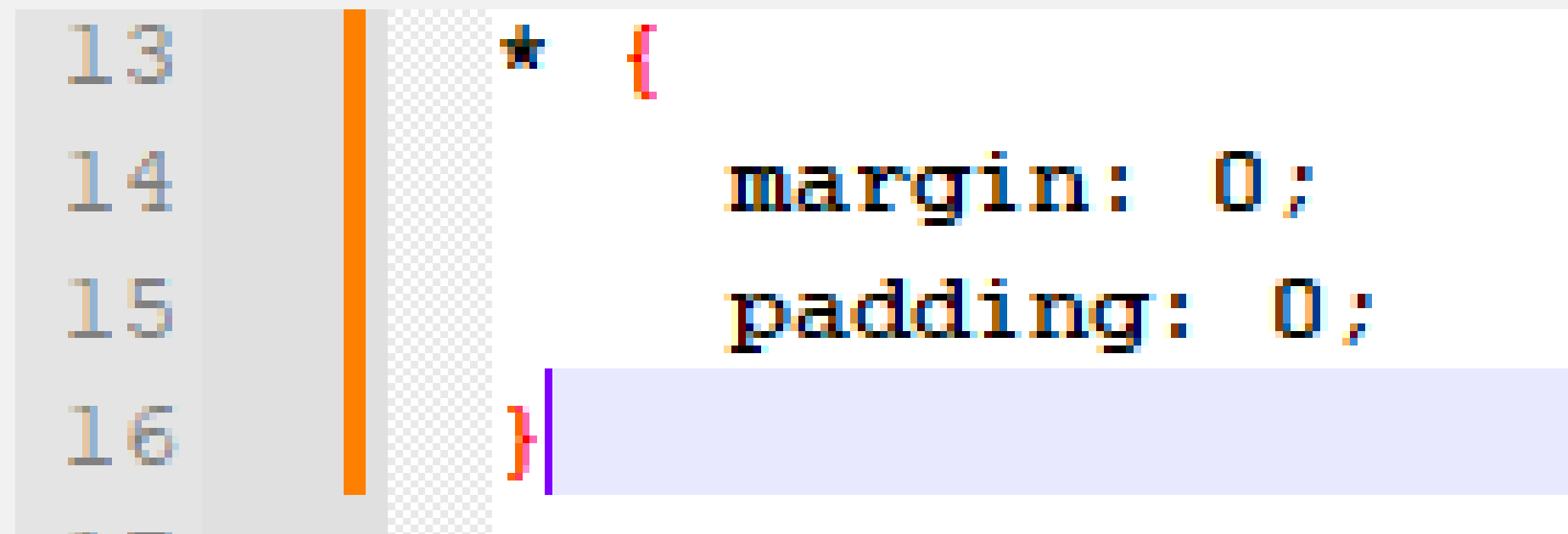
- **Seletor de ID:** Este seletor seleciona um elemento com um ID específico. IDs são definidos usando o atributo **id** no HTML e devem ser exclusivos em toda a página.

```
9  #cabecalho {  
10     font-size: 24px;  
11 }
```

Isso alterará o tamanho da fonte do elemento com o ID "cabecalho".

# CSS - Seletores

- **Seletor Universal:** Este seletor seleciona todos os elementos na página.

A screenshot of a code editor with a light gray background. On the left, there is a vertical scrollbar and a list of line numbers: 13, 14, 15, 16. The main area shows CSS code. Line 13 contains the universal selector `* {`. Line 14 contains `margin: 0;`. Line 15 contains `padding: 0;`. Line 16 contains `}`. The code is color-coded: the asterisk is blue, the opening and closing braces are red, and the property names and values are in a monospace font. A light blue horizontal bar highlights the closing brace on line 16.

```
13 * {  
14     margin: 0;  
15     padding: 0;  
16 }
```

Isso define margens e preenchimento zero para todos os elementos.

# CSS - Seletores

- **Seletor de Descendente:** Este seletor seleciona elementos que são descendentes de um elemento específico.

```
18  div p {  
19      font-style: italic;  
20  }
```

Isso aplicará itálico a todos os elementos `<p>` que são descendentes de um `<div>`.



# CSS - Propriedades

- **color:** Define a cor do texto.
- **font-size:** Define o tamanho da fonte do texto.
- **margin:** Define as margens externas de um elemento.
- **padding:** Define o preenchimento interno de um elemento.
- **background-color:** Define a cor de fundo de um elemento.
- **border:** Define a borda de um elemento.
- **width e height:** Define as dimensões de um elemento.
- **position:** Define o posicionamento de um elemento.
- **display:** Define como um elemento é exibido (por exemplo, block, inline, flex, etc.).

# CSS - Propriedades

- **text-align:** Define o alinhamento horizontal do texto dentro de um elemento.
- **float:** Define o posicionamento de elementos em relação aos outros (usado principalmente para layout de colunas).
- **border-radius:** Define o raio dos cantos arredondados de um elemento.

Estes são apenas alguns exemplos das muitas propriedades CSS disponíveis para personalizar o estilo e o layout dos elementos em uma página web.

# CSS - Estilos

O termo "**estilos**" geralmente se refere a um conjunto ou combinação de propriedades CSS que são aplicadas a um ou mais elementos para alcançar um determinado design ou aparência visual. Em outras palavras, um estilo é uma coleção de propriedades CSS que trabalham juntas para definir como um elemento ou conjunto de elementos deve ser exibido

# CSS - Estilos

O termo "**estilos**" geralmente se refere a um conjunto ou combinação de propriedades CSS que são aplicadas a um ou mais elementos para alcançar um determinado design ou aparência visual. Em outras palavras, um estilo é uma coleção de propriedades CSS que trabalham juntas para definir como um elemento ou conjunto de elementos deve ser exibido

```
22 p {  
23     color: #333;  
24     font-size: 16px;  
25     font-family: Arial, sans-serif;  
26 }
```

# Exercícios - CSS

Segunda parte do exercício é a criação de um arquivo CSS que formate o html gerado anteriormente. A formatação deverá ter o fundo em tom de cinza, título centralizado na parte superior com o tamanho da fonte em 50 px, o formulário centralizado com o fundo branco e as bordas circulares pretas com 1 pixel. a fonte dos labels deverá ser Calibri 15 na cor preta e o botão deverá ser preto com a escrita em branco. A mensagem de sucesso deverá ser centralizada e com o tamanho de fonte em 50 px abaixo do formulário.

**Dúvidas???**