

# Aprendendo HTML

- [HTML](#)
  - [Linguagem HTML](#)
  - [Estrutura HTML](#)
- [TAGS - HTML](#)
  - [Tags de Comentários](#)
  - [Tags Estruturais](#)
  - [Tags de Conteúdo](#)
  - [Tags de Texto](#)
  - [Tag de Link](#)
  - [Tags de Multimídia](#)
  - [Tags de Listas](#)
  - [Tags de Formulário](#)
  - [Tags de Estilos e Scripts](#)
- [TAGS - HTML](#)
  - [Tags de Comentários](#)
  - [Tags Estruturais](#)
  - [Tags de Conteúdo](#)
  - [Tags de Texto](#)
  - [Tag de Link](#)
  - [Tags de Multimídia](#)
  - [Tags de Listas](#)
  - [Tags de Formulário](#)
  - [Tags de Estilos e Scripts](#)
- [Crie sua Página em HTML](#)

- Sua Página em HTML
- Utilizando CSS no HTML
  - Vinculando Arquivos CSS
  - Seletores CSS
  - Propriedades e Valores CSS
  - Estilização de Elementos, Classes e IDs

HTML

# Linguagem HTML

## HTML

Neste texto vamos entender como montar uma estrutura de uma página web e entender um pouco sobre o que HTML HyperText Markup Language - linguagem de marcação de hipertexto), está em toda web. Então todos os sites que nós conhecemos são feitos a partir dessa linguagem de marcação chamada HTML. O HTML é interpretado pelos navegadores como o FireFox, Internet Explorer, Google e Chrome entre outros. Você escreve HTML utilizando estruturas chamadas tags.

## O que são tags HTML?

As tags são usadas para informar ao navegador a estrutura do site. Ou seja: quando se escreve um código em HTML, as tags serão interpretadas pelo navegador, produzindo assim a estrutura e o conteúdo visual da página. A principal característica das tags é estarem sempre dentro dos sinais de chevron (sinal de “maior que” e “menor que”), ou seja: `< >`.

As tags HTML são divididas em dois tipos: as que precisam de fechamento e as que não precisam de fechamento. As tags que precisam de fechamento possuem a sintaxe `<tag> </tag>`, já as que não precisam de fechamento possuem como estrutura `<tag/>`. Além disso, uma mesma tag pode receber um ou mais atributos, que possuirá um valor que modifica sua estrutura ou funcionalidade.

## Atributos

Os atributos são usados para personalizar as tags, modificando sua estrutura ou funcionalidade. Igualmente, os atributos são utilizados para atribuir uma classe ou id a um elemento. A maioria das tags tem seus próprios atributos. Contudo, existem alguns atributos genéricos que podem ser utilizados na maioria das tags HTML, vamos estudá-los:

- `class="..."` - Atribui uma classe ao elemento (uma classe pode ser utilizada para um ou mais elementos);
- `id="..."` - Atribui um id ao elemento (um id deve ser único, ou seja atribuído a um único elemento);
- `style="..."` - Permite incluir elementos CSS (estilos) dentro da tag;
- `lang="..."` - Define o idioma principal do elemento;
- `title="..."` - Define o título do elemento;
- `alt="..."` - Define um texto alternativo e, por isso, é muito utilizado em imagens, auxilia nas práticas de SEO;
- `hidden` - Oculta o elemento;
- `align="..."` - Permite definir o padrão de alinhamento desse elemento, como por exemplo: `right`, `center`, `left` e `justify`;
- `width="..."` - Define uma largura para o elemento;
- `height="..."` - Define uma altura para o elemento.

Essas são os principais atributos, porém existem diversos outros que devem ser estudados e podem ser utilizados no seu código.

# Estrutura HTML

## Estrutura básica de um documento HTML

Um documento HTML recebe algumas tags que formam a sua estrutura básica. No HTML5, o documento padrão recebe a seguinte estrutura:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

Em primeiro lugar, vamos agora entender para que serve cada uma dessas tags:

- `<!DOCTYPE html>` - A tag `!DOCTYPE` informa ao navegador a versão do HTML que está sendo utilizada no documento. Por exemplo: no HTML5, basta incluir `!DOCTYPE html`, e assim o navegador já saberá que se trata de um documento na versão HTML5;
- `<html></html>` - Esta tag é o elemento básico da estrutura do HTML. Assim sendo, ela conterá todos os elementos do seu documento. Todo documento HTML deve iniciar e finalizar com essa tag;
- `<head></head>` - Essa tag delimita o cabeçalho do documento. Não possui nenhum valor visível, porém é capaz de transmitir ao navegador diversas informações muito úteis e essenciais a uma boa apresentação do seu documento HTML;
- `<title></title>` - Essa tag define o título da sua página, o nome que aparecerá na sua aba, janela ou guia. Por esta razão, a tag `>title>` é de grande importância para o SEO;
- `<meta/>` - Essa tag permite inserir metadados ao seu documento, no caso acima, a informação `charset="UTF-8"`, que garante a compatibilidade do código com os caracteres de padrão latino americano;
- `<body></body>` - Finalmente, a tag que representa o corpo do documento. Em síntese, é nessa tag que todos os elementos visíveis do seu site devem ser inseridos.

Agora que já conhecemos as tags HTML que formam a estrutura básica de uma página, podemos então estudar as diversas tags que contemplarão o documento.

# TAGS - HTML

# Tags de Comentários

## Tags de comentários em HTML

Dentro de um documento, muitas vezes precisamos fazer comentários, para facilitar no desenvolvimento. Com isso, o código fica mais organizado e podemos deixar anotações importantes para possíveis mudanças, ou apenas para orientar o código. Dessa forma, na tag de comentários (que é aberta com `<!--` e fechada com `-->`), todos elementos incluídos dentro dela serão apenas comentários, ou seja, não serão visíveis no navegador.

Ex:

```
<!-- Meu comentário em HTML -->  
<p>Olá Mundo</p> <!-- aqui é o meu paragrafo-->
```

Como resultado final no navegador, teremos:

Olá Mundo

Repare que todo conteúdo das tags de comentário não aparecerá, sendo restrito apenas a quem estiver lendo o seu código HTML. Pratique utilizar os comentários em seu código, isso é uma boa prática e muito recomendada dentro do mundo do desenvolvimento.

# Tags Estruturais

## Tags HTML estruturais

As tags abaixo são utilizadas nos documentos em HTML5, e têm função estrutural no seu código. Portanto, essas tags têm grande importância na questão semântica da sua página, saber utilizá-las pode adequar o seu código para uma melhor visualização por parte do navegador e do usuário, além de proporcionar uma otimização para os processos de SEO.

- `<header></header>` - Essas tags definem um cabeçalho. Portanto, todo conteúdo que estiver dentro dela faz parte de um cabeçalho, podendo ser utilizado dentro de outras sessões. Não confundir com as tags `<head>`
- `<main></main>` - Essas tags representam o conteúdo principal do seu corpo, ou seja, o conteúdo relacionado diretamente com o tópico central da página ou com a funcionalidade central da aplicação;
- `<footer></footer>` - Essas tags definem um rodapé para a página, geralmente utilizadas no final da página;
- `<section></section>` - Essas tags definem uma sessão para sua página;
- `<article></article>` - Essas tags definem um artigo da sua página. Nesse sentido, são utilizadas para separar o conteúdo da sua página. Muito utilizado principalmente por blogs ou páginas de conteúdo;
- `<aside></aside>` - Essas tags representam uma seção de uma página cujo conteúdo é tangencialmente relacionado ao conteúdo do seu entorno, que poderia ser considerado separado do conteúdo;
- `<nav></nav>` - Essa tag define um conteúdo de navegação. Portanto, é muito utilizado em conjunto com listas e na criação de menus;
- `<div></div>` - Define uma divisão da página. Desta forma, funciona como um container para conteúdo de fluxo. Uma vez que não possui um valor semântico, é muito utilizado para organizar melhor o conteúdo.

Anteriormente ao HTML5, era utilizado no lugar das categorias acima.

# Tags de Conteúdo

## A tags HTML de conteúdo

Agora que você já conhece as principais tags estruturais, pode organizar o seu conteúdo de forma adequada. Assim sendo, vamos listar as principais tags para incluir conteúdo à página, como títulos, parágrafos, imagens, links, etc.

## Tags HTML de título

As tags de título possuem valor semântico, variando entre seis níveis hierárquicos. Para fins de otimização SEO é importante entender como funcionam, e fazer uma utilização adequada. Para definir títulos, utilizamos as tags:

- `<h1></h1>` - Título de maior valor hierárquico
- `<h2></h2>`
- `<h3></h3>`
- `<h4></h4>`
- `<h5></h5>`
- `<h6></h6>` - Título de menor valor hierárquico

Conforme dito anteriormente, a distribuição de título interferem no otimização do SEO de uma página.

Com isso, uma página contendo apenas h1 não seria bem vista pelos mecanismos de busca. Portanto, é importante dividir os títulos deixando o h1 apenas para o título principal, h2 ou h3 para títulos das seções e h4 a h6 para subtítulos ou títulos que possuam baixo valor hierárquico em relação aos demais.

# Tags de Texto

## Tags HTML de texto

As tags de texto definem textos, estilos de fonte, parágrafos, spans, quebras de linhas, etc. Vamos conhecê-las:

- `<p></p>` - Principal tag de texto, compõe um parágrafo;
- `<span></span>` - Apesar de ter uma funcionalidade e características parecidas com os parágrafos, costumam ser utilizadas apenas para pequenas informações, como legendas de um formulário, legendas de uma imagem, etc. também pode ser utilizada para formar um container;
- `<pre></pre>` - Tag utilizada para representar texto pré-formatado. Muito utilizada para inserir códigos;
- `<b></b>` - Transforma o conteúdo em negrito;
- `<i></i>` - Transforma o conteúdo em itálico;
- `<br/>` - Essa tag não necessita de fechamento, ela executa a função de quebra de linha.
- `<hr/>` - Essa tag não necessita de fechamento, ela forma uma linha horizontal.

# Tag de Link

## Tag de link HTML

A tag de link HTML é responsável que faz a ligação entre um documento e outro, sendo ele da mesma página ou de uma página de outro domínio. Esse elemento garantiu que o HTML se destacasse, e moldou a internet da forma que ela é hoje! Portanto, é o principal fundamento que forma a web.

Para realizar um link, podemos chamar as tags `<a></a>` com o atributo `href`. Por exemplo, caso você queira criar um link no seu texto que redirecione à página inicial do google:

```
<p>Para acessar o Google, <a href="https://www.google.com">clique aqui.</a></p>
```

Como resultado, teremos:

Para acessar o Google, [clique aqui](https://www.google.com).

# Tags de Multimídia

## Tags HTML de multimídia

As tags de multimídia servem para incluir imagens, vídeos, áudios, iframes e outros tipos de conteúdo multimídia.

### <img>

Essa tag não necessita de fechamento, serve para incluir uma imagem ao seu texto. A partir dessa tag, utilizamos o atributo `src=""` onde deve ser digitado o local em que a imagem se encontra. Também é muito utilizado em conjunto com o atributo `alt ""` para definir o texto alternativo da imagem.

Por exemplo:

```

```

### <video> e <audio>

Essa tag serve para indicar a inserção de um vídeo. Com isso, devemos colocar alguns atributos dentro da tag para poder ativar o controle, a principal dela é o `controls`. Posteriormente, incluímos a tag `<source>` com os atributos `src` e `type`, onde receberão o local em que está o vídeo e o tipo do vídeo. É recomendado utilizar mais de um tipo de `source` do mesmo vídeo, para garantir maior possibilidade de compatibilidade com o navegador. Finalmente, podemos incluir uma mensagem que será renderizada apenas se o navegador do usuário não suportar a tag `video`.

Vejamos os exemplos a seguir:

```
<video controls>  
<source src="video.mp4" type="video/mp4">  
<source src="video.ogv" type="video/ogg">  
Seu navegador não possui suporte para Vídeos.  
</video>
```

A tag `<audio>` funciona da mesma forma que a tag `<video>`, portanto, utilizamos a tag `<source>` dentro dela.

Vejamos o exemplo a seguir:

```
<audio controls>
<source src="musica.ogg" type="audio/ogg">
<source src="musica.mp3" type="audio/mpeg">Seu navegador não
possui suporte para áudio.
</audio>
```

## <iframe>

Os iframes são muito utilizados na atualidade, servem para incluir recursos de uma outra página nesta página. Vale a pena conferir o exemplo da W3C Schools, página de tutorias pertencente ao grupo W3C, a organização atualmente responsável pelos padrões da web. Portanto, para inserir um iframe, basta utilizar a tag com o atributo src. Além disso, é possível incluir um texto dentro do elemento, caso o navegador do usuário não possua suporte para tal.

Vejamos então o exemplo abaixo:

```
<iframe src="https://www.homehost.com.br">
<p>Seu navegador não possui suporte para iFrames.</p>
</iframe>
```

# Tags de Listas

## Tags HTML de listas

Para poder criar uma lista, podemos utilizar uma lista ordenada, a partir das tags `<ol></ol>`, ou uma lista não ordenada, a partir das tags `<ul></ul>`. Posteriormente, incluimos dentro da lista os elementos da mesma, dentro das tags `<li></li>`.

Vejam os exemplos a seguir:

```
<p>Minha lista ordenada:</p>
  <ol>
    <li>item 1</li>
    <li>item 2</li>
    <li>item 3</li>
  </ol>
<p>Minha lista não ordenada:</p>
  <ul>
    <li>item 1</li>
    <li>item 2</li>
    <li>item 3</li>
  </ul>
```

Como resultado final no navegador, teremos:

Minha lista ordenada:

1. item 1
2. item 2
3. item 3

Minha lista não ordenada:

- item 1
- item 2
- item 3

# Tags de Formulário

## Tags HTML de formulário

As tags de formulário são muito utilizadas para obter informações do usuário, realizar cadastros, receber opiniões, entre outros. São importantíssimas para qualquer ramo do mercado. Para iniciar um formulário, incluímos as tags `<form>` e `</form>`. Posteriormente, devemos incluir o conteúdo do formulário. Digamos que, por exemplo, você queira incluir três campos, sendo dois para coletar informações e um para receber uma mensagem. Desta forma, para criar os campos de preenchimento, deverá utilizar a tag `<input>`, enquanto para o campo referente à mensagem, deverá utilizar a tag `<textarea>`.

## A tag `<input>`

A tag `<input>` possui o atributo `type`, que varia entre diversos tipos (vamos explicar os principais deles abaixo). Também há o atributo `placeholder`, que é um texto que ficará disponível enquanto nada for digitado nesse campo. Também é importante definir um atributo `name` para cada `input`.

- `<input type="text">` - Define um campo que receberá qualquer caractere;
- `<input type="email">` - Define um campo que receberá caracteres e verificará se o mesmo consiste em um e-mail válido;
- `<input type="submit" value="ENVIAR">` - Define um botão que servirá para o envio do formulário. Dentro dele, podemos atribuir o `value`, que será o texto dentro do botão de envio.

Existem outros tipos de `<inputs>` que podem ser estudados na documentação disponível pela W3C.

```
<form>
  <input type="text" placeholder="Nome" name="nome"/>
  <input type="email" placeholder="Email" name="email"/>
  <input type="date" name="aniversario"/>
  <input type="password" placeholder="Senha" name="senha" />
  <button type="submit">Cadastrar</button>
</form>
```

**Como resultado final no navegador, teremos:**



Visualização de um formulário web com dois campos de entrada e um botão "Cadastrar".

As tags `<textarea></textarea>` .

Assim como a tag `<input>`, essa tag define um campo para o formulário. Porém, diferentemente, ela tem como principal característica ser uma área de preenchimento de texto, ou seja, permite que o usuário escreva um texto ou uma mensagem no seu interior. Também traz opções para que o usuário redimensione seu tamanho (resize). Dessa forma, podemos incluir uma área de texto utilizando as tags `<textarea>` e `</textarea>` .

```
<fieldset>
<legend>Sua opinião</legend>
<label for ="opinio"></label>
O que você mais gostou no evento ? <br>
<textarea cols=60 id="opinio" rows="10" name="opinio" maxlength="500" wrap="hard"
placeholder="coragem, você consegue ! "></textarea>
</fieldset>
```

### Como resultado final no navegador, teremos:

Sua opinião O que você mais gostou no evento ?

# Tags de Estilos e Scripts

## Tags de estilos e scripts

Para podermos concluir esse tutorial, não poderíamos deixar de citar as tags `<style>` e `<script>`. A tag `<style>` e `</style>` deve ser incluída no `<head>` do seu código HTML. Dentro dessa tag, é possível incluir todo o seu código CSS, ou seja, seu código de estilos.

Já a tag `<script>` e `</script>` tem como objetivo incluir códigos de scripts ao seu HTML, podendo ser incluída em qualquer parte. Contudo, recomenda-se fortemente que seja inserida após o `<footer>`. Dessa forma, podemos incluir nela um código javascript.

# TAGS - HTML

# Tags de Comentários

## Tags de comentários em HTML

Dentro de um documento, muitas vezes precisamos fazer comentários, para facilitar no desenvolvimento. Com isso, o código fica mais organizado e podemos deixar anotações importantes para possíveis mudanças, ou apenas para orientar o código. Dessa forma, na tag de comentários (que é aberta com `<!--` e fechada com `-->`), todos elementos incluídos dentro dela serão apenas comentários, ou seja, não serão visíveis no navegador.

Ex:

```
<!-- Meu comentário em HTML -->  
<p>Olá Mundo</p> <!-- aqui é o meu paragrafo-->
```

Como resultado final no navegador, teremos:

Olá Mundo

Repare que todo conteúdo das tags de comentário não aparecerá, sendo restrito apenas a quem estiver lendo o seu código HTML. Pratique utilizar os comentários em seu código, isso é uma boa prática e muito recomendada dentro do mundo do desenvolvimento.

# Tags Estruturais

## Tags HTML estruturais

As tags abaixo são utilizadas nos documentos em HTML5, e têm função estrutural no seu código. Portanto, essas tags têm grande importância na questão semântica da sua página, saber utilizá-las pode adequar o seu código para uma melhor visualização por parte do navegador e do usuário, além de proporcionar uma otimização para os processos de SEO.

- `<header></header>` - Essas tags definem um cabeçalho. Portanto, todo conteúdo que estiver dentro dela faz parte de um cabeçalho, podendo ser utilizado dentro de outras sessões. Não confundir com as tags `<head>`
- `<main></main>` - Essas tags representam o conteúdo principal do seu corpo, ou seja, o conteúdo relacionado diretamente com o tópico central da página ou com a funcionalidade central da aplicação;
- `<footer></footer>` - Essas tags definem um rodapé para a página, geralmente utilizadas no final da página;
- `<section></section>` - Essas tags definem uma sessão para sua página;
- `<article></article>` - Essas tags definem um artigo da sua página. Nesse sentido, são utilizadas para separar o conteúdo da sua página. Muito utilizado principalmente por blogs ou páginas de conteúdo;
- `<aside></aside>` - Essas tags representam uma seção de uma página cujo conteúdo é tangencialmente relacionado ao conteúdo do seu entorno, que poderia ser considerado separado do conteúdo;
- `<nav></nav>` - Essa tag define um conteúdo de navegação. Portanto, é muito utilizado em conjunto com listas e na criação de menus;
- `<div></div>` - Define uma divisão da página. Desta forma, funciona como um container para conteúdo de fluxo. Uma vez que não possui um valor semântico, é muito utilizado para organizar melhor o conteúdo.

Anteriormente ao HTML5, era utilizado no lugar das categorias acima.

# Tags de Conteúdo

## A tags HTML de conteúdo

Agora que você já conhece as principais tags estruturais, pode organizar o seu conteúdo de forma adequada. Assim sendo, vamos listar as principais tags para incluir conteúdo à página, como títulos, parágrafos, imagens, links, etc.

## Tags HTML de título

As tags de título possuem valor semântico, variando entre seis níveis hierárquicos. Para fins de otimização SEO é importante entender como funcionam, e fazer uma utilização adequada. Para definir títulos, utilizamos as tags:

- `<h1></h1>` - Título de maior valor hierárquico
- `<h2></h2>`
- `<h3></h3>`
- `<h4></h4>`
- `<h5></h5>`
- `<h6></h6>` - Título de menor valor hierárquico

Conforme dito anteriormente, a distribuição de título interferem no otimização do SEO de uma página.

Com isso, uma página contendo apenas h1 não seria bem vista pelos mecanismos de busca. Portanto, é importante dividir os títulos deixando o h1 apenas para o título principal, h2 ou h3 para títulos das seções e h4 a h6 para subtítulos ou títulos que possuam baixo valor hierárquico em relação aos demais.

# Tags de Texto

## Tags HTML de texto

As tags de texto definem textos, estilos de fonte, parágrafos, spans, quebras de linhas, etc. Vamos conhecê-las:

- `<p></p>` - Principal tag de texto, compõe um parágrafo;
- `<span></span>` - Apesar de ter uma funcionalidade e características parecidas com os parágrafos, costumam ser utilizadas apenas para pequenas informações, como legendas de um formulário, legendas de uma imagem, etc. também pode ser utilizada para formar um container;
- `<pre></pre>` - Tag utilizada para representar texto pré-formatado. Muito utilizada para inserir códigos;
- `<b></b>` - Transforma o conteúdo em negrito;
- `<i></i>` - Transforma o conteúdo em itálico;
- `<br/>` - Essa tag não necessita de fechamento, ela executa a função de quebra de linha.
- `<hr/>` - Essa tag não necessita de fechamento, ela forma uma linha horizontal.

# Tag de Link

## Tag de link HTML

A tag de link HTML é responsável que faz a ligação entre um documento e outro, sendo ele da mesma página ou de uma página de outro domínio. Esse elemento garantiu que o HTML se destacasse, e moldou a internet da forma que ela é hoje! Portanto, é o principal fundamento que forma a web.

Para realizar um link, podemos chamar as tags `<a></a>` com o atributo href. Por exemplo, caso você queira criar um link no seu texto que redirecione à página inicial do google:

```
<p>Para acessar o Google, <a href="https://www.google.com">clique aqui.</a></p>
```

Como resultado, teremos:

Para acessar o Google, [clique aqui](https://www.google.com).

# Tags de Multimídia

## Tags HTML de multimídia

As tags de multimídia servem para incluir imagens, vídeos, áudios, iframes e outros tipos de conteúdo multimídia.

### <img>

Essa tag não necessita de fechamento, serve para incluir uma imagem ao seu texto. A partir dessa tag, utilizamos o atributo `src=""` onde deve ser digitado o local em que a imagem se encontra. Também é muito utilizado em conjunto com o atributo `alt ""` para definir o texto alternativo da imagem.

Por exemplo:

```

```

### <video> e <audio>

Essa tag serve para indicar a inserção de um vídeo. Com isso, devemos colocar alguns atributos dentro da tag para poder ativar o controle, a principal dela é o `controls`. Posteriormente, incluímos a tag `<source>` com os atributos `src` e `type`, onde receberão o local em que está o vídeo e o tipo do vídeo. É recomendado utilizar mais de um tipo de `source` do mesmo vídeo, para garantir maior possibilidade de compatibilidade com o navegador. Finalmente, podemos incluir uma mensagem que será renderizada apenas se o navegador do usuário não suportar a tag `video`.

Vejamos os exemplos a seguir:

```
<video controls>  
<source src="video.mp4" type="video/mp4">  
<source src="video.ogv" type="video/ogg">  
Seu navegador não possui suporte para Vídeos.  
</video>
```

A tag `<audio>` funciona da mesma forma que a tag `<video>`, portanto, utilizamos a tag `<source>` dentro dela.

Vejamos o exemplo a seguir:

```
<audio controls>
<source src="musica.ogg" type="audio/ogg">
<source src="musica.mp3" type="audio/mpeg">Seu navegador não
possui suporte para áudio.
</audio>
```

## <iframe>

Os iframes são muito utilizados na atualidade, servem para incluir recursos de uma outra página nesta página. Vale a pena conferir o exemplo da W3C Schools, página de tutorias pertencente ao grupo W3C, a organização atualmente responsável pelos padrões da web. Portanto, para inserir um iframe, basta utilizar a tag com o atributo src. Além disso, é possível incluir um texto dentro do elemento, caso o navegador do usuário não possua suporte para tal.

Vejamos então o exemplo abaixo:

```
<iframe src="https://www.homehost.com.br">
<p>Seu navegador não possui suporte para iFrames.</p>
</iframe>
```

# Tags de Listas

## Tags HTML de listas

Para poder criar uma lista, podemos utilizar uma lista ordenada, a partir das tags `<ol></ol>`, ou uma lista não ordenada, a partir das tags `<ul></ul>`. Posteriormente, incluimos dentro da lista os elementos da mesma, dentro das tags `<li></li>`.

Vejam os exemplos a seguir:

```
<p>Minha lista ordenada:</p>
  <ol>
    <li>item 1</li>
    <li>item 2</li>
    <li>item 3</li>
  </ol>
<p>Minha lista não ordenada:</p>
  <ul>
    <li>item 1</li>
    <li>item 2</li>
    <li>item 3</li>
  </ul>
```

Como resultado final no navegador, teremos:

Minha lista ordenada:

1. item 1
2. item 2
3. item 3

Minha lista não ordenada:

- item 1
- item 2
- item 3

# Tags de Formulário

## Tags HTML de formulário

As tags de formulário são muito utilizadas para obter informações do usuário, realizar cadastros, receber opiniões, entre outros. São importantíssimas para qualquer ramo do mercado. Para iniciar um formulário, incluímos as tags `<form>` e `</form>`. Posteriormente, devemos incluir o conteúdo do formulário. Digamos que, por exemplo, você queira incluir três campos, sendo dois para coletar informações e um para receber uma mensagem. Desta forma, para criar os campos de preenchimento, deverá utilizar a tag `<input>`, enquanto para o campo referente à mensagem, deverá utilizar a tag `<textarea>`.

## A tag `<input>`

A tag `<input>` possui o atributo `type`, que varia entre diversos tipos (vamos explicar os principais deles abaixo). Também há o atributo `placeholder`, que é um texto que ficará disponível enquanto nada for digitado nesse campo. Também é importante definir um atributo `name` para cada `input`.

- `<input type="text">` - Define um campo que receberá qualquer caractere;
- `<input type="email">` - Define um campo que receberá caracteres e verificará se o mesmo consiste em um e-mail válido;
- `<input type="submit" value="ENVIAR">` - Define um botão que servirá para o envio do formulário. Dentro dele, podemos atribuir o `value`, que será o texto dentro do botão de envio.

Existem outros tipos de `<inputs>` que podem ser estudados na documentação disponível pela W3C.

```
<form>
  <input type="text" placeholder="Nome" name="nome"/>
  <input type="email" placeholder="Email" name="email"/>
  <input type="date" name="aniversario"/>
  <input type="password" placeholder="Senha" name="senha" />
  <button type="submit">Cadastrar</button>
</form>
```

**Como resultado final no navegador, teremos:**

As tags `<textarea></textarea>` .

Assim como a tag `<input>`, essa tag define um campo para o formulário. Porém, diferentemente, ela tem como principal característica ser uma área de preenchimento de texto, ou seja, permite que o usuário escreva um texto ou uma mensagem no seu interior. Também traz opções para que o usuário redimensione seu tamanho (resize). Dessa forma, podemos incluir uma área de texto utilizando as tags `<textarea>` e `</textarea>` .

```
<fieldset>
<legend>Sua opinião</legend>
<label for ="opinio"></label>
O que você mais gostou no evento ? <br>
<textarea cols=60 id="opinio" rows="10" name="opinio" maxlength="500" wrap="hard"
placeholder="coragem, você consegue ! "></textarea>
</fieldset>
```

### Como resultado final no navegador, teremos:

Sua opinião O que você mais gostou no evento ?

# Tags de Estilos e Scripts

## Tags de estilos e scripts

Para podermos concluir esse tutorial, não poderíamos deixar de citar as tags `<style>` e `<script>`. A tag `<style>` e `</style>` deve ser incluída no `<head>` do seu código HTML. Dentro dessa tag, é possível incluir todo o seu código CSS, ou seja, seu código de estilos.

Já a tag `<script>` e `</script>` tem como objetivo incluir códigos de scripts ao seu HTML, podendo ser incluída em qualquer parte. Contudo, recomenda-se fortemente que seja inserida após o `<footer>`. Dessa forma, podemos incluir nela um código javascript.

# Crie sua Página em HTML

Crie sua Página em HTML

# Sua Página em HTML

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Título da página</title>
  <style>
    /*AQUI VAI TODO NOSSO CÓDIGO CSS*/
  </style>
</head>
<body>
  <header><!--criando um cabeçalho para nossa página com um menu-->
    <h2>Minha Página</h2>
    <nav><!--vamos criar um menu utilizando listas-->
      <ul>
        <li>Home</li>
        <li>Meu menu</li>
      </ul>
    </nav><!--aqui finaliza o meu menu-->
  </header>
  <main>
    <section><!--vamos criar a primeira section do meu documento-->
      <article>
        <h3>Titulo do meu artigo</h3>
        <p>Conteúdo do meu artigo</p>
        <p>Mais conteúdo para o meu artigo</p>
        <ol><!--vamos criar uma lista ordenada-->
          <li>item 1</li>
          <li>item 2</li>
          <li>item 3</li>
        </ol>
      </article><!-- aqui finaliza meu primeiro artigo da section-->
      <article>
```

```

    <h3>Titulo do meu segundo artigo</h3>
    <p>Conteúdo do meu segundo artigo</p>
    <p>Mais conteúdo para o meu segundo artigo</p>
  </article><!--aqui finaliza meu segundo artigo da section-->
</section><!--aqui encerra a primeira section do meu documento-->
</main><!--aqui finaliza todo conteúdo principal do corpo da pagina-->
<aside>
  <h4>Conteúdos relacionados</h4>
  <ul>
    <li>Página oficial da <a href="https://www.homehost.com.br/">HomeHost</a></li>
    <li>item da lista</li>
    <li>item da lista</li>
  </ul>
</aside>
<footer>
  <div><!--criando uma divisão para meu rodapé-->
    <p>Inscreva-se para receber noticias</p>
    <form method="post">
      <input type="text" name="nome" placeholder="Digite seu Nome">
      <input type="email" name="email" placeholder="Digite seu Email">
      <input type="submit" name="enviar" value="Enviar">
    </form>
  </div>
  <div><!--criando outra para meu rodapé-->
    <h3>Minha primeira página html</h3>
    <span>Todos os direitos reservados</span>
  </div>
</footer>
<script>
//AQUI VAI NOSSO CÓDIGO DE SCRIPT (JAVASCRIPT)
</script>
<body>
</html>

```

**Como resultado final no navegador, teremos:**

# Minha Página

- Home
- Meu menu

## Titulo do meu artigo

Conteudo do meu artigo

Mais conteúdo para o meu artigo

1. item 1
2. item 2
3. item 3

## Titulo do meu segundo artigo

Conteudo do meu segundo artigo

Mais conteúdo para o meu segundo artigo

## Conteúdos relacionados

- Página oficial da [HomeHost](#)
- item da lista
- item da lista

Inscreva-se para receber noticias

## Minha primeira página html

Todos os direitos reservados



# Utilizando CSS no HTML

Neste capítulo, vamos explorar como aplicar estilos CSS a elementos HTML e como você pode controlar a apresentação de suas páginas web. Aprenderemos como vincular arquivos CSS externos, criar seletores, definir propriedades e valores, e, finalmente, como estilizar eficazmente seu conteúdo HTML.

# Vinculando Arquivos CSS

## Vinculando Arquivos CSS a uma Página HTML

Para aplicar estilos CSS a uma página HTML, é fundamental vincular um arquivo CSS externo. Esse arquivo contém todas as regras de estilo que serão aplicadas à página. Aqui estão os passos básicos para fazer isso:

1. **Crie um arquivo CSS:** Comece criando um arquivo de texto com a extensão `.css`, por exemplo, "estilos.css". Este arquivo conterá todas as regras de estilo.
2. **Vincule o arquivo CSS à sua página HTML:** No elemento `<head>` do seu documento HTML, adicione o seguinte código para vincular o arquivo CSS externo:

```
<head>
  <link rel="stylesheet" type="text/css" href="estilos.css">
</head>
```

Agora, todas as regras de estilo definidas no arquivo "estilos.css" serão aplicadas aos elementos HTML da página.

# Seletores CSS

## Seletores CSS

Os seletores são a parte central do CSS, permitindo que você especifique quais elementos HTML deseja estilizar. Existem vários tipos de seletores, incluindo:

- **Seletores de tipo:** Estilizam todos os elementos de um determinado tipo, como `<h1>`, `<p>`, ou `<a>`.
- **Seletores de classe:** Permitem que você estilize elementos com base em suas classes. Por exemplo, `<p class="destaque">`.
- **Seletores de ID:** Estilizam elementos com base em seus IDs exclusivos. Por exemplo, `<div id="cabecalho">`.
- **Seletores universais:** Estilizam todos os elementos na página, utilizando o asterisco (\*).

# Propriedades e Valores CSS

## Propriedades e Valores CSS

As propriedades definem quais aspectos do elemento serão estilizados, enquanto os valores determinam como essas propriedades serão aplicadas. Aqui estão alguns exemplos:

```
seletor {  
  propriedade: valor;  
}
```

- **Propriedade:** Define o que será estilizado, como `color` para a cor do texto, `font-size` para o tamanho da fonte, ou `margin` para margens.
- **Valor:** Especifica o valor da propriedade, como `blue` para a cor ou `16px` para o tamanho da fonte.

# Estilização de Elementos, Classes e IDs

## Estilização de Elementos, Classes e IDs

Vamos dar uma olhada em exemplos de como estilizar elementos, classes e IDs:

```
/* Estilizando um elemento */  
p {  
  font-size: 18px;  
  color: #333;  
}  
  
/* Estilizando uma classe */  
.destaque {  
  background-color: yellow;  
  font-weight: bold;  
}  
  
/* Estilizando um ID */  
#cabecalho {  
  font-size: 24px;  
  text-align: center;  
}
```

Estes são apenas exemplos iniciais para ilustrar como aplicar estilos usando seletores, propriedades e valores. Você pode criar estilos complexos e personalizados para se adequar ao design desejado de suas páginas web.

No próximo capítulo, você aprenderá a criar uma página web completa, aplicando os conceitos de HTML e CSS que exploramos até agora.