

MANUAL DE PROGRAMAÇÃO

Senac Vila Prudente

HTML

HTML Básico

HTML é uma linguagem de marcação composta de elementos que são representados por tags. As tags geralmente são compostas por uma tag de abertura `<tag>` e uma tag de fechamento `</tag>`, com o conteúdo entre elas. Aqui estão algumas tags HTML básicas:

`<!DOCTYPE html>`: Declara o tipo de documento HTML.

`<html>`: Define o início e o fim do documento HTML.

`<head>`: Contém informações sobre a página, como o título e links para estilos ou scripts.

`<title>`: Define o título da página que aparece na aba do navegador.

`<body>`: Contém o conteúdo visível da página.

`<meta>`: Fornece metadados sobre a página, como a codificação de caracteres.

`<link>`: Liga a folhas de estilo CSS externas.

`<script>`: Inclui scripts JavaScript na página.

`<body>`: Contém o conteúdo principal da página, como texto, imagens e outros elementos.

Exemplo de Documento HTML Básico:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Minha Página HTML</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>Olá, Mundo!</h1>
  <p>Esta é uma página HTML simples.</p>
</body>
</html>
```

Estrutura de Texto

Você pode usar tags HTML para estruturar e formatar o texto em sua página. Alguns exemplos:

<h1>, **<h2>**, **<h3>**, **<h4>**, **<h5>**, **<h6>**: Cabeçalhos de diferentes níveis.

<p>: Parágrafos.

****: Texto enfatizado (geralmente exibido em itálico).

****: Texto importante (geralmente exibido em negrito).

```
<h1>Título Principal</h1>
<p>Isso é um <strong>texto em negrito</strong> e isso é um <em>texto em itálico</em>.</p>
```

Links e Imagens

Texto do Link: Cria um link para outra página ou recurso.

```
<a href="https://www.example.com">Visitar o Exemplo</a>
```

``:

Inclui uma imagem na página.

```

```

Listas

Listas não ordenadas: `` e ``.

Listas ordenadas: `` e ``.

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>

<ol>
  <li>Primeiro</li>
  <li>Segundo</li>
</ol>
```

Tabelas

`<table>`: Cria uma tabela.

`<tr>`: Define uma linha na tabela.

`<td>`: Define uma célula de dados na tabela.

```
<table>
  <tr>
    <th>Nome</th>
    <th>Idade</th>
  </tr>
  <tr>
    <td>João</td>
    <td>30</td>
  </tr>
  <tr>
    <td>Maria</td>
    <td>25</td>
  </tr>
</table>
```

Neste exemplo, temos uma tabela com duas colunas (Nome e Idade) e duas linhas de dados. Os cabeçalhos das colunas são definidos usando `<th>`, enquanto os dados reais são colocados em células `<td>`.

Além disso, você pode usar atributos em elementos de tabela para personalizar a aparência e o comportamento da tabela, como `border` para definir a largura da borda, `cellpadding` e `cellspacing` para controlar o espaçamento interno e externo das células, e assim por diante.

É importante notar que, nos padrões modernos de desenvolvimento web, tabelas são geralmente usadas apenas para dados tabulares reais. Para layouts de página, é preferível usar CSS e elementos de layout como `<div>` e `<section>`.

Div e Section

`<div>` e `<section>` são elementos HTML usados para organizar e estruturar o conteúdo em uma página da web, mas eles têm finalidades diferentes e são usados em contextos distintos:

`<div>` (divisão):

O elemento `<div>` é um bloco genérico de divisão ou contêiner no HTML.

Ele é usado principalmente para agrupar elementos e aplicar estilos CSS a eles ou para manipular o conteúdo por meio de scripts **JavaScript**.

Não possui uma semântica específica, ou seja, não carrega nenhum significado especial por si só.

É frequentemente usado para criar layouts e estruturar a página em seções.

Exemplo de uso:

```
<div id="header">
  <h1>Meu Site</h1>
  <p>Bem-vindo ao meu site.</p>
</div>
```

<section> (seção):

O elemento **<section>** é usado para definir uma seção temática ou significativa de conteúdo em uma página da web.

Ele é uma maneira de estruturar o conteúdo com base em sua semântica, tornando o código mais compreensível e acessível.

Pode conter cabeçalhos, parágrafos, listas e outros elementos relacionados ao tópico da seção.

É uma escolha apropriada para estruturar partes distintas de uma página, como capítulos de um livro, artigos de um blog etc.

Exemplo de uso:

```
<section>
  <h2>Notícias</h2>
  <p>Aqui estão as últimas notícias do mundo.</p>
</section>
```

Em resumo, o elemento **<div>** é usado para criar contêineres genéricos para organização e estilo, enquanto o elemento **<section>** é usado para estruturar o conteúdo com base em sua semântica, dividindo-o em partes significativas. A escolha entre

eles depende da intenção de design e da estrutura de conteúdo da sua página. É importante usar elementos HTML semânticos como `<section>` sempre que possível, pois isso melhora a acessibilidade e a compreensão do seu código por parte dos motores de busca e desenvolvedores.

Cellpadding e Cellspacing

cellpadding e **cellspacing** são atributos de tabelas HTML que afetam o espaçamento entre as células e o espaçamento dentro das células de uma tabela.

cellpadding (espaçamento interno de células):

O atributo **cellpadding** controla o espaço entre o conteúdo de uma célula e sua borda interna (a parte dentro da célula onde o conteúdo é exibido).

Você pode definir um valor em pixels para este atributo para determinar o espaço interno de todas as células da tabela.

Um valor maior de **cellpadding** resultará em mais espaço interno nas células, tornando o conteúdo mais afastado das bordas.

Exemplo:

```
<table cellpadding="10">
  <tr>
    <td>Conteúdo da célula 1</td>
    <td>Conteúdo da célula 2</td>
  </tr>
</table>
```

cellspacing (espaçamento entre células):

O atributo **cellspacing** controla o espaço entre as células adjacentes em uma tabela.

Você pode definir um valor em pixels para este atributo para determinar o espaçamento entre todas as células da tabela.

Um valor maior de **cellspacing** resultará em mais espaço entre as células, criando um espaço visível entre elas.

```
<table cellpadding="5">
  <tr>
    <td>Conteúdo da célula 1</td>
    <td>Conteúdo da célula 2</td>
  </tr>
</table>
```

É importante observar que o uso de `cellpadding` e `cellspacing` é considerado uma prática obsoleta no desenvolvimento web moderno. Em vez disso, é recomendável usar CSS para controlar o espaçamento e a aparência de tabelas, usando propriedades como `padding` e `border-spacing`. O CSS oferece maior flexibilidade e controle sobre o design da tabela e separa a formatação da estrutura do documento HTML.

Formulários

`<form>`: Define um formulário.

`<input type="text">`, `<input type="password">`, `<input type="checkbox">`, `<input type="radio">`, `<input type="submit">`, entre outros, são usados dentro de formulários para coletar informações do usuário.

```
<form>
  <label for="nome">Nome:</label>
  <input type="text" id="nome" name="nome">
  <input type="submit" value="Enviar">
</form>
```

Comentários

Você pode adicionar comentários no código HTML usando `<!--` Seu comentário aqui `-->`. Os comentários não são visíveis na página, mas podem ser úteis para documentar seu código.

```
<!-- Este é um comentário -->
```

Elementos em linha

Elementos em linha, também conhecidos como inline, são elementos que ocupam somente a largura do seu conteúdo, como as tags:

`<a>`: é possível definir links com essa tag;

``: é um container genérico para agrupar textos;

`
`: cria uma quebra de linha;

``: insere imagens no site;

`<audio>`: insere áudios no site.

Elementos em bloco

Elementos em bloco, ou elementos block level, são elementos que ocupam todo o espaço do seu elemento pai. Por exemplo, as tags:

`<header>`: define o cabeçalho da página;

`<main>`: define o conteúdo principal da página;

`<footer>`: define o rodapé da página;

`<section>`: representa uma seção dentro do conteúdo principal;

`<article>`: inclui um artigo na página;

`<aside>`: faz referência a um outro conteúdo da página;

`<nav>`: cria o menu de navegação;

`<div>`: é como uma divisão na página para fins estéticos;

`<hr>`: constrói uma linha horizontal;

`<video>`: insere vídeos no site;

Caracteres especiais e símbolos

Você pode usar entidades de caractere para exibir caracteres especiais e símbolos que não estão disponíveis diretamente no teclado ou que podem causar conflitos com a sintaxe do código HTML. As entidades de caractere começam com & e terminam com ;.

Aqui estão alguns exemplos de entidades de caractere comuns:

< :exibe o símbolo < (menor que);

> :exibe o símbolo > (maior que);

& :exibe o símbolo & (e comercial);

" :exibe o símbolo " (aspas duplas);

' :exibe o símbolo ' (aspas simples);

** ** :exibe um espaço não quebrável;

© :exibe o símbolo de direitos autorais ©;

® :exibe o símbolo de marca registrada ®

™ :exibe o símbolo de marca comercial ™.

Um exemplo de uso é o seguinte, em que aplicamos uma entidade de caractere dentro da tag de texto:

```
<p>&lt; &gt; &amp; &quot; &apos; &nbsp; &copy; &reg; &trade;</p>
```

O resultado desse código é: < > & " ' © ® ™.

Além dessas entidades de caractere, você também pode usar códigos numéricos para caracteres especiais. Os códigos numéricos começam com &# e terminam com ;. Por exemplo:

© :exibe o símbolo de direitos autorais ©;

® :exibe o símbolo de marca registrada ®;

™ :exibe o símbolo de marca comercial ™.

Para usar, você também pode aplicar esses valores dentro de uma tag de parágrafo:

```
<p>© ® ™;</p>
```


E esperar o resultado: © ® ™. Essas entidades de caractere são úteis quando você precisa exibir símbolos especiais ou caracteres que não são facilmente inseridos diretamente no código HTML.

Elemento **
** (Quebra de linha):

Você pode usar o elemento **
** para criar uma quebra de linha manualmente.

Não possui uma tag de fechamento, e é inserido como uma tag única.

Exemplo:



```
<p>Esta é uma linha de texto.<br>Esta é uma nova linha.</p>
```

Elemento **<p>** (Parágrafo):

O elemento **<p>** é usado para criar parágrafos e, por padrão, insere uma quebra de linha antes e depois do conteúdo.

Exemplo:

```
<p>Este é o primeiro parágrafo.</p>
<p>Este é o segundo parágrafo.</p>
```

Ao criar conteúdo para a web, o uso adequado de parágrafos e estilos CSS é geralmente preferível às quebras de linha manuais com **
**. As quebras de linha manuais devem ser usadas com moderação e apenas quando necessário, como em endereços ou poemas, em que a formatação é crítica. O uso excessivo de quebras de linha pode tornar o código HTML menos legível e dificultar a manutenção do site.

Para adicionar emojis ao seu site HTML usando a Emojipedia, siga os passos abaixo:

Visite a Emojipedia:

Acesse o site da Emojipedia em <https://emojipedia.org/>.

Pesquise o Emoji Desejado:

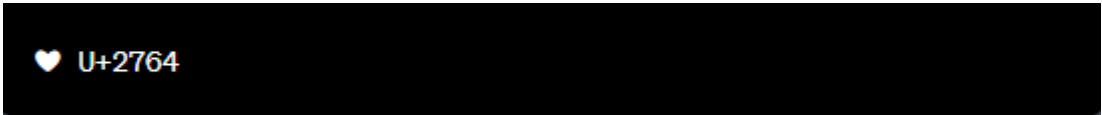
Use a barra de pesquisa ou navegue pelas categorias para encontrar o emoji que você deseja adicionar ao seu site.

Selecione o Emoji:

Clique no emoji que você deseja usar. Isso abrirá a página do emoji específico.

Copie o Código do Emoji:

Na página do emoji, você verá uma caixa de código que inclui uma representação textual do emoji. Por exemplo, o código para um emoji de coração pode ser algo como:




❤️ U+2764

Copie o código, incluindo o caractere U+.

Cole o Código no seu Site:

Agora você pode colar o código do emoji no seu site HTML, onde deseja exibi-lo. Você pode usar esse código em qualquer lugar que aceite texto, como em um parágrafo, um título, uma lista etc.

Exemplo de uso em HTML:



```
<p>Eu amo programar ❤️</p>
```

Certifique-se de que o emoji seja exibido corretamente no seu site. A maioria dos navegadores modernos deve renderizar os emojis corretamente.

Personalize a Aparência (Opcional):

Você também pode personalizar a aparência do emoji usando CSS, caso deseje alterar seu tamanho, cor ou qualquer outra propriedade de estilo.

Verifique a Compatibilidade:

Lembre-se de que a renderização de emojis pode variar entre diferentes sistemas operacionais e navegadores. É uma boa prática verificar como os emojis aparecem em diferentes dispositivos para garantir uma experiência consistente para os visitantes do seu site.

Certificar-se de que seu site use uma fonte que suporte emojis também é importante para garantir a exibição adequada. Muitas fontes modernas incluem suporte a emojis, mas é algo a ser considerado se você estiver criando um site personalizado.

Crie o Ícone Favicon:

Primeiro, você precisa criar um ícone de favicon. O ícone deve ser uma imagem quadrada com dimensões de pelo menos 16x16 pixels, mas é recomendável criar ícones maiores para melhor qualidade em dispositivos de alta resolução. Você pode usar um editor de imagem, como o Adobe Photoshop ou o GIMP, para criar o ícone. Certifique-se de salvar o ícone no formato .ico, .png, ou .svg.

Renomeie o Ícone para "favicon":

Salve o ícone com o nome "favicon" (sem as aspas), com a extensão adequada. Por exemplo, "favicon.ico" para ícones no formato .ico, "favicon.png" para ícones no formato .png, ou "favicon.svg" para ícones no formato .svg.

Coloque o Ícone na Raiz do seu Site:

Coloque o arquivo do ícone (favicon.ico, favicon.png ou favicon.svg) na raiz do diretório do seu site. Isso é importante para que os navegadores possam encontrá-lo facilmente.

Adicione o Código HTML ao seu Site:

Abra a página HTML do seu site (geralmente é o arquivo index.html) no editor de texto ou IDE de sua escolha.

Adicione a seguinte linha de código dentro do elemento <head> da página, geralmente entre as tags <head> e </head>:

```
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

Certifique-se de ajustar o valor do atributo href para corresponder ao nome do arquivo de ícone que você criou e colocou na raiz do site.

Exemplo para um ícone no formato .png:

```
<link rel="icon" href="favicon.png" type="image/png">
```

Salve e Atualize o Site:

Salve as alterações no arquivo HTML e atualize seu site no servidor da web.

Limpe o Cache do Navegador:

Para ver as mudanças imediatamente, é recomendável limpar o cache do seu navegador ou abrir uma nova janela do navegador para verificar o novo favicon.

Depois de seguir esses passos, o novo favicon deve ser exibido na guia do navegador e na barra de favoritos quando os visitantes acessarem o seu site.

Lembre-se de que as dimensões, o formato e a qualidade do favicon são importantes para garantir uma exibição atraente e consistente em vários dispositivos e navegadores. Certifique-se de criar ícones em conformidade com as diretrizes recomendadas.

Lorem

O "comando Lorem" que você está se referindo é geralmente chamado de "Lorem Ipsum". Não é exatamente um comando HTML, mas sim um texto fictício usado como espaço reservado em documentos ou designs. "Lorem Ipsum" é uma sequência de texto em latim que tem sido usada na indústria de impressão e design gráfico por muitos anos. É comumente usado para preencher espaços em branco em layouts de página ou como um espaço reservado para conteúdo que ainda não foi criado.

Aqui está como você pode usar o "Lorem Ipsum" em HTML:

Inserindo o Lorem Ipsum em HTML:

Você pode simplesmente copiar e colar o texto "Lorem Ipsum" em seu código HTML onde deseja usá-lo. Aqui está um exemplo:

```
<p>  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
  Sed ut purus eget lectus varius condimentum a non ligula.  
  Vivamus aliquam, lorem eget faucibus vestibulum, justo nulla.  
</p>
```

Usando Geradores de Lorem Ipsum:

Em vez de escrever o texto manualmente, você pode usar geradores de Lorem Ipsum online que criam parágrafos de texto fictício automaticamente. Basta copiar o texto gerado e colá-lo no seu HTML. Alguns geradores permitem que você especifique o número de parágrafos, palavras ou caracteres.

Personalizando o Texto:

Se você quiser personalizar o texto fictício, você pode optar por traduzir o "Lorem Ipsum" para o idioma que desejar ou simplesmente criar seu próprio texto fictício personalizado.

O "Lorem Ipsum" é útil quando você está projetando um site ou um documento e deseja preencher o espaço com texto que se assemelha ao conteúdo real, mas não quer se distrair com o texto real até que o conteúdo esteja pronto. É uma prática comum para designers e desenvolvedores usarem o "Lorem Ipsum" como espaço reservado até que o conteúdo final seja inserido.

Tag Address

A tag <address> em HTML é usada para marcar informações de contato ou detalhes de endereço, como informações de contato de uma pessoa, organização, local ou localização física. O conteúdo dentro dessa tag normalmente inclui informações de contato, como endereço, telefone, e-mail e outros detalhes de contato.

Aqui está um exemplo de como usar a tag **<address>**:

```
<address>
  <p>John Doe</p>
  <p>1234 Elm Street</p>
  <p>City, State, ZIP Code</p>
  <p>Email: <a href="mailto:john.doe@example.com">john.doe@example.com</a></p>
  <p>Phone: <a href="tel:+123456789">+1 234 567 89</a></p>
</address>
```

Neste exemplo, **<address>** envolve informações de contato de "John Doe", incluindo seu nome, endereço, endereço de e-mail e número de telefone. Os links de e-mail (****) e de telefone (****) são usados para criar links clicáveis que permitem que os visitantes cliquem para enviar um e-mail ou ligar para o número de telefone, respectivamente.

Além disso, é importante notar que o elemento **<address>** é um elemento semântico, o que significa que ele carrega significado em relação ao conteúdo. Os mecanismos de busca e leitores de tela podem interpretar o conteúdo marcado com **<address>** como informações de contato, tornando-o útil para acessibilidade e SEO.

A tag **<address>** é mais comumente usada em páginas da web que fornecem informações de contato, como páginas de perfil de pessoas, páginas de contato ou rodapés de sites que contêm informações de contato da empresa. Lembre-se de que, embora seja uma boa prática usar a tag **<address>** para marcar informações de contato, você pode estilizar e formatar esses dados usando CSS para atender às necessidades de design do seu site.

Texto Grande e Pequeno

Em HTML, você pode definir o tamanho do texto usando várias técnicas e unidades de medida. Aqui estão as principais maneiras de ajustar o tamanho do texto em HTML:

Usando a Tag **** (Não recomendado):

A tag **** era usada no HTML 4.01 para definir o tamanho da fonte, mas ela é obsoleta em HTML5 e não deve ser usada.

Exemplo (não recomendado):

```
<font size="4">Este é um texto com tamanho 4.</font>
```

Usando Estilos CSS em Linha (Inline CSS):

Você pode definir o tamanho da fonte diretamente em um elemento HTML usando o atributo **'style'** com a propriedade **'font-size'**.

A unidade de medida mais comum é o **'px'** (pixels), mas você também pode usar outras unidades, como **'em'**, **'%'**, **'rem'** etc.

Exemplo:

```
<p style="font-size: 16px;">Este é um parágrafo com tamanho de fonte 16 pixels.</p>
```

Usando CSS Incorporado (na seção <style> do cabeçalho HTML):

Você pode definir estilos CSS em um bloco **<style>** dentro da seção **<head>** do seu documento HTML e, em seguida, aplicar esses estilos aos elementos no corpo do documento.

Exemplo:

```
<head>
  <style>
    p {
      font-size: 18px;
    }
  </style>
</head>
<body>
  <p>Este é um parágrafo com tamanho de fonte 18 pixels.</p>
</body>
```

Usando CSS Externo (Arquivo CSS Separado):

Você pode criar um arquivo CSS separado com todas as regras de estilo, incluindo o tamanho da fonte, e vincular esse arquivo ao seu documento HTML usando a tag **<link>**.

Exemplo de arquivo CSS externo (**estilos.css**):

```
p {
  font-size: 20px;
}
```


Exemplo de HTML com ligação ao arquivo CSS:

```
<head>
  <link rel="stylesheet" type="text/css" href="estilos.css">
</head>
<body>
  <p>Este é um parágrafo com tamanho de fonte 20 pixels.</p>
</body>
```

Usando Unidades Relativas (em, rem, %):

Além de `px`, você pode usar unidades relativas para definir o tamanho da fonte. As unidades `em` e `rem` se baseiam no tamanho da fonte do elemento pai, enquanto `%` é uma porcentagem do tamanho padrão da fonte do navegador.

Exemplo com '`em`':

```
<p style="font-size: 1.2em;">Este é um parágrafo com 1,2 vezes o tamanho da fonte do elemento pai.</p>
```

Usando Media Queries (Estilos Responsivos):

Você pode usar media queries em CSS para definir tamanhos de fonte diferentes com base na largura da tela ou em outras características do dispositivo.

Exemplo:

```
@media (max-width: 600px) {
  p {
    font-size: 16px;
  }
}
@media (min-width: 601px) {
  p {
    font-size: 20px;
  }
}
```

Lembre-se de que é importante escolher tamanhos de fonte legíveis e manter a acessibilidade em mente ao estilizar o texto em seu site. Além disso, a unidade de medida '`em`' é frequentemente usada para criar layouts escaláveis e acessíveis,

já que se ajusta automaticamente ao tamanho da fonte do elemento pai, tornando o site mais flexível para diferentes tamanhos de tela e definições de acessibilidade.

Tag Small

A tag `<small>` em HTML é usada para definir o tamanho do texto em um nível menor em relação ao texto circundante. Isso é geralmente usado para indicar que o texto entre as tags `<small>` é de importância menor, como notas de rodapé, legenda de fotos, observações ou informações adicionais. A tag `<small>` não é usada para aplicar estilos visuais, mas sim para indicar que o texto deve ser renderizado em um tamanho menor, sem a necessidade de especificar um tamanho de fonte exato.

Aqui está um exemplo de como usar a tag `<small>`:

```
<p>Este é um parágrafo de texto normal.</p>  
<small>Este é um texto em tamanho menor, geralmente usado para notas de rodapé.</small>
```

O texto envolvido pela tag `<small>` será exibido em um tamanho de fonte menor, mas a especificação do tamanho exato depende das configurações de estilo padrão do navegador. A tag `<small>` é semântica, o que significa que não se destina apenas a diminuir o tamanho da fonte, mas também a comunicar a intenção de que o texto é de importância secundária em relação ao texto circundante.

É importante observar que o tamanho do texto especificado pela tag `<small>` pode ser ajustado pelo usuário final, dependendo das configurações de acessibilidade do navegador. Isso permite que os leitores de tela ou pessoas com deficiências visuais ajustem o tamanho do texto para torná-lo legível para eles.

No entanto, se você deseja aplicar estilos de tamanho de fonte específicos ao texto, é melhor usar CSS em vez da tag `<small>`. O CSS oferece maior controle sobre a formatação do texto, permitindo que você defina tamanhos de fonte exatos e aplique outros estilos, se necessário. Exemplo de CSS para definir o tamanho de fonte:

```
<p style="font-size: 12px;">Este é um parágrafo com tamanho de fonte de 12 pixels.</p>
```

Em resumo, a tag **<small>** é usada para indicar que o texto deve ser exibido em um tamanho menor, mas a aparência exata é determinada pelas configurações de estilo padrão do navegador. Se você deseja um controle mais preciso sobre o tamanho da fonte, use CSS para aplicar estilos de formatação específicos.

Tag Del e Ins

Em HTML, os elementos **** (deletado) e **<ins>** (inserido) são usados para marcar mudanças em um documento, como texto que foi removido ou texto que foi adicionado. Eles são úteis em contextos de controle de versão, revisão de documentos ou sempre que você deseja indicar alterações específicas no texto.

Aqui está uma explicação mais detalhada sobre cada um deles:

** (Deletado):**

A tag **** é usada para marcar o texto que foi removido ou excluído de uma versão anterior do documento. Normalmente, o texto marcado com **** é exibido com uma linha riscada ou uma linha tracejada para indicar que foi removido.

Exemplo:

```
<p>Este é um <del>texto deletado</del> no documento.</p>
```

<ins> (Inserido):

A tag **<ins>** é usada para marcar o texto que foi adicionado ou inserido em uma versão posterior do documento. Normalmente, o texto marcado com **<ins>** é exibido com sublinhado ou outra formatação para indicar que foi inserido.

Exemplo:

```
<p>Este é um <ins>texto inserido</ins> no documento.</p>
```

É importante notar que o comportamento visual dessas tags (como o uso de linha riscada, sublinhado, ou outra formatação) depende da folha de estilos CSS aplicada ao documento. As tags **** e **<ins>** fornecem a marcação semântica para indicar alterações, enquanto a formatação específica é definida pelo CSS.

As tags **** e **<ins>** também podem incluir atributos **datetime** que indicam quando a alteração foi feita. Isso fornece informações adicionais sobre a alteração.

Por exemplo:

```
<p>Este é um <del datetime="2023-09-15T15:30:00-07:00">texto deletado</del> no documento.</p>
<p>Este é um <ins datetime="2023-09-15T15:35:00-07:00">texto inserido</ins> no documento.</p>
```

Em um contexto de controle de versão ou revisão de documentos, o uso de **** e **<ins>** pode ajudar a comunicar claramente as alterações feitas em um documento e fornecer informações adicionais sobre quando essas alterações foram feitas.

Citações e Códigos

Em HTML, você pode usar as tags **<blockquote>** para criar citações e **<code>** para incorporar código-fonte ou texto que deve ser exibido em uma fonte monoespaçada para distinguir o código do texto normal. Aqui estão as explicações detalhadas para cada uma dessas tags:

<blockquote> (Citações):

A tag **<blockquote>** é usada para marcar citações em HTML. Isso é comumente usado para citar trechos de texto de outras fontes, como livros, artigos ou sites.

Pode ser usado tanto para citações curtas quanto para blocos longos de texto.

Um atributo **cite** pode ser adicionado para fornecer a URL da fonte da citação.

Exemplo:

```
<blockquote>
  <p>Isso é uma citação de um autor famoso.</p>
</blockquote>
```

<code> (Código):

A tag **<code>** é usada para marcar pedaços de código-fonte ou texto que deve ser exibido em uma fonte monoespaçada, geralmente para destacar o código do texto normal.

É útil para inserir código em documentos ou páginas da web, e o navegador normalmente formata o texto entre as tags `<code>` com uma fonte monoespaçada.

Se você desejar destacar ainda mais o código com sintaxe colorida e formatação avançada, convém considerar o uso de bibliotecas ou plugins de realce de sintaxe, como o Prism.js.

Exemplo:

```
<p>Aqui está um exemplo de código em HTML: <code>&lt;p&gt;Texto de exemplo&lt;/p&gt;</code></p>
```

Embora essas tags sejam úteis para fins específicos, lembre-se de que a apresentação visual de elementos como `<blockquote>` e `<code>` depende da folha de estilos CSS do seu site. Você pode personalizar a aparência desses elementos usando CSS para corresponder ao estilo geral do seu site. Além disso, o uso adequado dessas tags é importante para manter a acessibilidade e a estrutura semântica da página da web.

Tag Abbr

A tag `<abbr>` em HTML é usada para marcar uma abreviatura ou sigla em um documento e, opcionalmente, fornecer uma expansão ou descrição completa da abreviatura usando o atributo `title`. Isso é útil para melhorar a acessibilidade e a compreensão do conteúdo, pois permite que os leitores saibam o significado completo da abreviatura ao passar o mouse sobre ela.

A estrutura básica da tag `<abbr>` é a seguinte:

```
<abbr title="Descrição completa da abreviatura">Abreviatura</abbr>
```

title: Este é o atributo obrigatório que fornece a descrição completa da abreviatura. Quando o cursor do mouse é colocado sobre a abreviatura, um tooltip (dica de ferramenta) exibe o conteúdo do atributo `title`.

Abreviatura: Isso é onde você coloca a abreviatura ou sigla que deseja marcar no seu documento.

Aqui está um exemplo de como usar a tag **<abbr>**:

```
<p>A <abbr title="HyperText Markup Language">HTML</abbr> é a linguagem de marcação usada para criar páginas da web.</p>
```

Neste exemplo, quando o usuário move o cursor do mouse sobre a palavra "HTML", verá uma dica de ferramenta com a descrição completa "HyperText Markup Language".

O uso da tag **<abbr>** é particularmente útil para tornar o conteúdo mais acessível a pessoas que podem não estar familiarizadas com certas abreviaturas ou siglas, bem como para melhorar a compreensão do conteúdo em geral. Além disso, pode ser útil para mecanismos de busca entenderem o significado das abreviaturas em um documento, o que pode ter um impacto positivo no SEO (Otimização para Mecanismos de Busca).

Links Internos

Links internos, também conhecidos como âncoras internas ou links de navegação, são elementos HTML que permitem que você crie links dentro de um site, página da web ou documento para navegar para outras partes do mesmo documento ou para outras páginas dentro do mesmo site. Os links internos são uma parte fundamental da navegação e da estrutura de um site, tornando a experiência do usuário mais eficiente.

Aqui está uma explicação abrangente sobre links internos em HTML:

Sintaxe do Link Interno:

Para criar um link interno em HTML, você usa a tag **<a>** (âncora) com o atributo **href**, que aponta para o local de destino, geralmente usando um identificador chamado de âncora.

A âncora é definida no destino usando o atributo **id**.

Exemplo de link interno:

```
<a href="#secao2">Ir para a Seção 2</a>
```

Destino do Link Interno:

O destino do link interno é especificado usando o atributo href da tag **<a>**.

Pode apontar para um identificador em uma página, para outra página dentro do mesmo site ou para um caminho de arquivo relativo ou absoluto.

Exemplos:

Navegar para uma seção na mesma página:

```
<a href="#secao2">Ir para a Seção 2</a>
```

Navegar para outra página no mesmo site:

```
<a href="/outrapagina.html">Ir para outra página</a>
```

Identificadores (Âncoras):

Para criar um link interno que aponte para uma seção específica da mesma página, você deve atribuir um identificador à seção de destino usando o atributo id.

Exemplo de seção com identificador:

```
<h2 id="secao2">Seção 2</h2>
```

Estilo dos Links Internos:

Os links internos podem ser estilizados usando CSS para corresponder à estética do site.

O uso de pseudo-classes, como **:hover** e **:visited**, permite que você personalize a aparência dos links internos.

Compatibilidade e Acessibilidade:

Ao criar links internos, certifique-se de que funcionem corretamente em diferentes navegadores e dispositivos.

Os links internos também devem ser projetados com acessibilidade em mente, para que sejam utilizáveis por todas as pessoas, incluindo aquelas com deficiências visuais que usam leitores de tela.

SEO (Otimização para Mecanismos de Busca):

O uso apropriado de links internos é importante para a otimização de mecanismos de busca, pois ajuda os motores de busca a entenderem a estrutura e o conteúdo do seu site.

Os links internos desempenham um papel crucial na navegação de um site, ajudando os visitantes a se deslocarem facilmente entre as diferentes partes do site. Eles também melhoram a usabilidade, a acessibilidade e a otimização para mecanismos de busca do seu site. Portanto, é importante usá-los de maneira eficaz na criação de páginas da web.

Hospedagem do Site:

Escolha um Provedor de Hospedagem: Você precisará de um serviço de hospedagem web para armazenar os arquivos do seu site e disponibilizá-los online. Alguns provedores populares incluem Bluehost, HostGator, SiteGround e muitos outros. Considere suas necessidades de hospedagem, como o tipo de site que está criando e o tráfego esperado, ao escolher um provedor.

Registre uma Conta de Hospedagem: Inscreva-se em um plano de hospedagem oferecido pelo provedor escolhido. Isso envolve a escolha do plano, a configuração de sua conta e o pagamento.

Conecte-se ao Servidor FTP: Depois de configurar sua conta de hospedagem, você precisará carregar os arquivos do seu site no servidor. A maioria dos provedores de hospedagem oferece acesso FTP (File Transfer Protocol) para fazer isso. Você pode usar um programa de FTP, como o FileZilla, para se conectar ao servidor e transferir seus arquivos.

Publicação do Site: Transfira os arquivos do seu site para a pasta raiz do servidor ou para o diretório apropriado, dependendo das instruções do seu provedor. Isso pode incluir arquivos HTML, CSS, JavaScript, imagens e qualquer outro conteúdo necessário para o seu site.

Teste o Site: Depois de publicar os arquivos, teste o site para garantir que tudo funcione corretamente. Verifique se todas as páginas são acessíveis e se os links e recursos estão funcionando.

Registro de Domínio:

Escolha um Registrador de Domínio: Você pode registrar um domínio por meio de registradores de domínio, como GoDaddy, Namecheap, Google Domains, entre outros.

Verifique a Disponibilidade: Use a ferramenta de pesquisa do registrador para verificar se o domínio desejado está disponível. Se estiver disponível, você poderá prosseguir com o registro.

Registre o Domínio: Após escolher um domínio disponível, siga as instruções do registrador para registrar o domínio. Isso envolverá a configuração de informações de contato e pagamento.

Configurações de DNS: Quando você registra um domínio, o próximo passo é configurar as informações de DNS (Domain Name System) para apontar o domínio para o servidor web onde seu site está hospedado. Isso permite que as solicitações de acesso ao seu domínio sejam direcionadas para o servidor correto.

Configurações de DNS:

Acesse a Área de Configurações de DNS: Acesse a conta de domínio no registrador e localize a seção de configurações de DNS.

Configure os Registros DNS: Geralmente, você precisará configurar os registros DNS para o seu domínio. Os registros comuns incluem registros A (para apontar para endereços IP), registros CNAME (para apontar para outros domínios), registros MX (para configuração de email) e registros TXT (para informações adicionais).

Preencha as Informações de Hospedagem: Dependendo do seu provedor de hospedagem, você receberá as informações de DNS específicas a serem usadas. Isso pode incluir o endereço IP do servidor ou nomes de host.

Salve as Configurações: Depois de configurar os registros DNS, salve as alterações. Isso pode levar algum tempo para propagar pela internet, normalmente algumas horas a 48 horas, durante as quais seu domínio começará a apontar para o servidor de hospedagem.

Lembre-se de que os detalhes exatos para hospedar um site e configurar um domínio podem variar de acordo com o provedor de hospedagem e o registrador de domínio que você escolher. Certifique-se de seguir as instruções específicas fornecidas por esses serviços.

CSS

O CSS é uma linguagem de estilo que descreve como os elementos HTML devem ser exibidos em uma página da web. Ele separa a estrutura (HTML) da apresentação (CSS) do conteúdo.

Inclusão de CSS em uma Página

Você pode adicionar CSS a uma página da web de várias maneiras:

CSS Inline

Use o atributo **style** ou inclua um arquivo de folha de estilo externo para estilizar elementos HTML.

Exemplo (atributo **style**):

```
<p style="color: blue; font-size: 16px;">Este é um parágrafo azul.</p>
```

CSS Interno

Você pode incluir CSS internamente na seção <head> de uma página usando a tag <style>:

```
<head>
  <style>
    p {
      color: blue;
      font-size: 16px;
    }
  </style>
</head>
```

CSS Externo

A melhor prática é usar um arquivo CSS externo e vinculá-lo ao seu documento HTML. Crie um arquivo **.css** separado e vincule-o assim:

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

Seletores CSS

Os seletores CSS são usados para selecionar elementos HTML aos quais você deseja aplicar estilos. Aqui estão alguns exemplos:

Selecionar todos os parágrafos:

```
p {
  /* Estilos aqui */
}
```

Selecionar elementos com uma classe específica:

```
.classe {
  /* Estilos aqui */
}
```

Selecionar um elemento com um ID específico:

```
#id {
  /* Estilos aqui */
}
```

Propriedades CSS

A propriedade `color` em CSS é usada para definir a cor do texto em elementos HTML, como parágrafos (`<p>`), títulos (`<h1>`, `<h2>`, etc.), links (`<a>`), entre outros. Essa propriedade permite que você escolha a cor que deseja que o texto seja exibido no navegador.

Existem várias maneiras de especificar a cor usando a propriedade `color`:

Nomes de Cores Pré-definidos: CSS oferece uma série de nomes de cores pré-definidos que você pode usar diretamente. Por exemplo:

```
p {  
  color: red; /* Define a cor do texto para vermelho */  
}
```

Notação Hexadecimal: Você pode especificar cores usando a notação hexadecimal, que usa uma combinação de seis caracteres alfanuméricos que representam os valores de vermelho, verde e azul (RGB). Por exemplo:

```
p {  
  color: #FF5733; /* Define a cor do texto para um laranja avermelhado */  
}
```

Notação RGB: A notação RGB (Red, Green, Blue) permite que você defina a cor do texto especificando os valores de intensidade de vermelho, verde e azul. Por exemplo:

```
p {  
  color: rgb(255, 87, 51); /* Define a cor do texto para um laranja avermelhado */  
}
```

Notação RGBA: A notação RGBA é semelhante à RGB, mas inclui um quarto valor para a transparência (alfa). Isso permite criar cores semi-transparentes. Por exemplo:

```
p {  
  color: rgba(255, 87, 51, 0.5); /* Define um texto laranja avermelhado semi-transparente */  
}
```

Notação HSL: A notação HSL (Hue, Saturation, Lightness) permite definir cores com base no matiz, saturação e luminosidade. Por exemplo:

```
p {  
  color: hsl(10, 100%, 50%); /* Define a cor do texto para uma to  
}
```

Notação HSLA: Semelhante à HSL, a notação HSLA também inclui um valor alfa para transparência. Por exemplo:

```
p {  
  color: hsla(10, 100%, 50%, 0.7); /* Define um texto laranja sem  
}
```

A escolha da notação de cores depende da sua preferência e das necessidades de design do seu projeto. A propriedade color é uma das muitas propriedades CSS que permitem que você personalize a aparência dos elementos em suas páginas da web.

Tamanho da fonte:

```
font-size: 16px;
```

Margem:

```
margin: 10px;
```

Preenchimento:

```
padding: 5px;
```