**Estilos de formatação (b, strong, I,**

**em, sub, sup)**

Tag B

<p>Texto normal</p>

<p><b>Texto destacado bold.</b></p>

Tag Strong

<p>Texto normal</p>

<p><strong>Texto importante</strong></p>

Qual a diferença entre os dois? Utilizando a tag B ele transforma em negrito o texto selecionado e só, já utilizando strong, semanticamente indica um texto de grande importância, isso pode afetar a forma como os mecanismos de busca interpretam.

Tag i

<p>Texto normal</p>

<p><i>Texto em itálico</i></p>

Tag em

<p>Texto normal</p>

<p><em>Texto destacado</em></p>

No desenvolvimento web moderno, semanticamente para se dar destaque em uma frase, o ideal é utilizar as tags <strong> e <em>. As tags <i> e <b> podem ser utilizadas

porém de uma forma que não seja como uma forma de destaque.

Tag sub

<p> H2O <p>

<p>H<sub>2</sub>0</p>

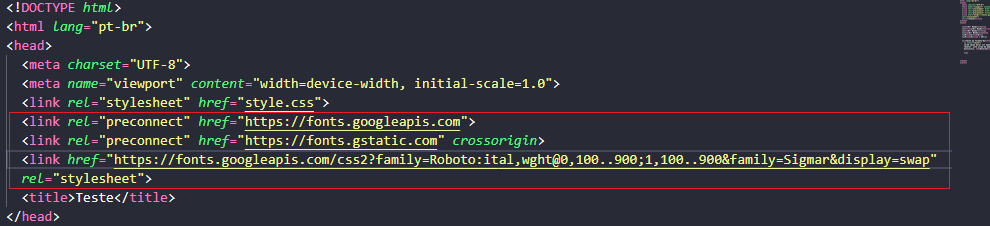
Tag sup

<p>4<sup>2</sup> = 16</p>

**Fonte**

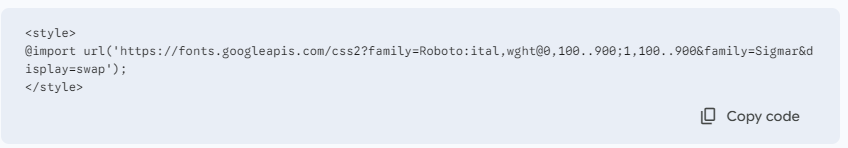
Você pode importar fontes da internet para adicioná-las ao seu site, um exemplo de uso seria o <https://fonts.google.com/> , nele você pode selecionar a fonte que deseja e colocar em seu site de duas formas diferentes seja por link em seu html  
  


**google fonts**

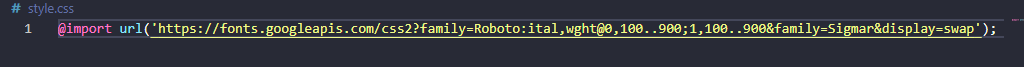


**VS Code**

Ou importando via css



**google fonts**

  
**VS Code**

Dessa forma podemos manipular a fonte que selecionados em nosso CSS com alguns elementos como:

**font-family:** Usado para trocar a fonte de nosso texto ou trecho de algum texto, seja ela uma fonte serif, sans serif, monospace…

**font-size :** Para aumentar ou diminuir o tamanho de nossa fonte.  
**font-style:** Para colocar nosso texto ou um trecho dele em negrito, itálico..

**Atributos de Formatação e Alinhamento**

**text-align:** Define o alinhamento horizontal do conteúdo de texto dentro de um elemento. Valores comuns incluem left, right, center e justify.

**text-indent:** Especifica a indentação (recuo) da primeira linha de um bloco de texto. Por exemplo, text-indent: 20px; aplica um recuo de 20 pixels na primeira linha do texto.

**line-height:** Controla a altura da linha do texto, influenciando o espaçamento vertical entre as linhas. Um valor comum é line-height: 1.5;, que define a altura da linha como 1,5 vezes o tamanho da fonte.

**letter-spacing:** Ajusta o espaçamento entre os caracteres do texto. Por exemplo, letter-spacing: 0.05em; aumenta o espaçamento entre as letras em 0,05 vezes o tamanho da fonte.

**text-transform:** Modifica a capitalização do texto. Valores possíveis são uppercase (caixa alta), lowercase (caixa baixa) e capitalize (primeira letra maiúscula).

**text-decoration:** Define decorações no texto, como sublinhar, sobrealinhar ou nenhum. Por exemplo, text-decoration: underline; aplica um sublinhado ao texto.

**Elementos de âncoras (links)**

A tag <a> em HTML, conhecida como elemento âncora (ou link), é fundamental para criar hyperlinks que permitem a navegação entre diferentes páginas, seções dentro de uma mesma página, sites externos ou para download de arquivos.

**Dentro de arquivos**

Para criar links que direcionam o usuário para diferentes seções dentro do mesmo documento HTML, utilize o atributo id nos elementos de destino e o atributo href nos links apontando para esses identificadores.

<nav>

<ul>

<li><a href = "#secao1" >Seção 1</a></li>

</ul>

</nav>

<section id=”secao1”>

<p>Conteúdo deste parágrafo</p>

</section>

Neste exemplo, os links no menu de navegação direcionam o usuário para as seções correspondentes dentro da mesma página. Ao clicar em "Seção 1", por exemplo, a página rolará até o conteúdo associado a essa seção.

**Entre arquivos**

Para criar links que direcionam o usuário para outras páginas dentro do mesmo site ou para páginas externas, utilize o atributo href com o caminho relativo ou absoluto do destino.

**Link para outra página no mesmo site**

<a href="pagina2.html">Ir para a Página 2</a>

**Link para uma página externa**

<a href="https://www.exemplo.com" target="\_blank">Visitar Exemplo</a>

O atributo target="\_blank" faz com que o link seja aberto em uma nova aba ou janela do navegador.

**Para download de arquivos**

Para criar links que permitam o download de arquivos, utilize o atributo href apontando para o arquivo desejado e o atributo download para sugerir o nome do arquivo a ser salvo.

<a href="documento.pdf" download="NovoDocumento.pdf">Baixar Documento</a>

Ao clicar neste link, o navegador fará o download do arquivo "documento.pdf" e sugerirá o nome "NovoDocumento.pdf" para o arquivo salvo.

**Formatação de links**

**Pseudo classes**

Mudar a cor de um link ao passar o mouse por cima:

a:hover {

color: blue;

}

Faz com que ao passar o mouse por cima do link, ele fique com a cor azul.

a:active {

color: red;}

Sempre que você clicar em um link e segurar, ele mudará a cor.

a:visited {

color: black;

}

Quando você acessar um link, ele mudará de cor para mostrar que aquele link já foi clicado.

Para criar alguns efeitos diferentes tanto em fundo quanto em imagens, podemos usar algumas ferramentas junto ao hover. Vamos ver como ficaria o código abaixo

No html:  
<section class=”Teste”>

<h1> Botão</h1>

</section>

No css:

/\*.Teste {

padding: 50px;

text-align: center;

}

.Teste h1{

background-color: blue;

color: white;

padding: 20px;

width: 100px;

text-align: center;

margin: 0 auto;

}

Criaremos uma simples caixa com texto ao centro, que se colocarmos um link pode se tornar um botão. Agora vamos ver algumas propriedades que podemos colocar junto ao hover.

.Teste h1:hover{

transform: translate(-10px, -20px);

.Teste h1:hover{

transform: rotate(30deg);

.Teste h1:hover{

transform: skew(-10deg, -10deg);

.Teste h1:hover{

transform: scale(0.8);

.Teste h1:hover{

transform: perspective(500px) rotateY(45deg);

.Teste h1:hover{

transform: perspective(500px) translateY(8px) rotateY(45deg);

.Teste h1:hover{

transform: perspective(500px) scaleY(0.7) rotateY(45deg);

.Teste h1:hover{

transform: perspective(200px) rotateX(40deg) scale(1.2);

.Teste h1:hover{

transform: matrix3d(0.6, 0.1, 0.7, 0, -0.5, 0.8, 0.1, 0, -0.6, -0.5, 0.5, 0, 0, 0, 0, 1);

**Pseudo Elementos**

Se você colocar da seguinte forma :

***::selection* {**

**color: red;**

**background-color: aqua;**

**}**

Como você não definiu nenhuma area específica, ao selecionar qualquer texto do seu site, ele transforma o texto na cor vermelha e o fundo do texto na cor aqua.

Mude somente a primeira letra do seu texto:

p*::first-letter* {

color: #ff0000;

font-size: xx-large;

}

Criando desta forma, você mudará apenas a primeira lera do seu texto para um tamanho em específico e na cor vermelha.

Utilizando o ::before

<!DOCTYPE *html*>

<html *lang*="pt-br">

<head>

<meta *charset*="UTF-8">

<meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

<link *rel*="stylesheet" *href*="style.css">

<title>Teste</title>

</head>

<body>

<button *class*="botao">Pesquisar</button>

</body>

</html>

**HTML**

***.botao* {**

**padding: 10px 40px;**

**font-size: 16px;**

**cursor: pointer;**

**position: relative;**

**border: 2px solid #007bff;**

**background-color: #ffffff;**

**color: #007bff;**

**border-radius: 5px;**

**transition: all 0.3s ease;**

**}**

***.botao::before* {**

**content: '🔍'; /\* Ícone de lupa \*/**

**position: absolute;**

**left: 10px;**

**top: 50%;**

**transform: translateY(-50%);**

**font-size: 18px;**

**}**

***.botao:hover* {**

**background-color: #007bff;**

**color: #ffffff;**

**}**

***.botao:hover::before* {**

**content: '✔️'; /\* Ícone de check \*/**

**color: #ffffff;**

**}**

**CSS**

Esta é uma forma interessante de se utilizar um ::before em um botão, neste código ao passar o mouse sobre um botão ele irá mudar a cor e o ícone adicionado com o ::before, ou seja, inicialmente foi adicionado um ícone de lupa, ao passar o mouse por cima do botão ele coloca o ícone de “check”. Caso queira colocar o ícone ao final do escrito dentro do botão, ou ao final de algum texto por exemplo, utilize o pseudo elemento ::after.

**Medidas Relativas**

**REM**

A unidade rem é relativa ao tamanho da fonte raiz (<html>). Ou seja, 1 rem é igual ao tamanho da fonte do elemento raiz. Se a fonte raiz for 16px (o valor padrão na maioria dos navegadores), 1rem será igual a 16px.

Por exemplo, se a fonte raiz for 16px, 1rem = 16px, 2rem = 32px, 0.5rem = 8px.

**Escalabilidade e Responsividade:** O uso de rem permite que o layout e a tipografia se ajustem automaticamente quando o usuário altera o tamanho da fonte no navegador ou quando a tela tem diferentes resoluções. Isso torna a experiência do usuário mais acessível.

**Acessibilidade:** Como o valor é relativo ao tamanho da fonte raiz, os usuários podem ajustar o tamanho da fonte no navegador sem perder a legibilidade do site.

**Consistência:** Se você alterar o tamanho da fonte no elemento raiz (por exemplo, em um tema de modo claro/escuro ou em dispositivos móveis), o layout e os elementos em rem se ajustarão proporcionalmente, mantendo a consistência.

**Facilidade de manutenção:** Ao utilizar rem, você define uma vez o tamanho da fonte base (em geral, no html ou body) e o resto da tipografia e espaçamentos podem ser baseados em rem, tornando a manutenção mais fácil e evitando números fixos e mágicos espalhados pelo código.

**Medidas Absolutas**

**PX**

Criando um parágrafo comum:

**<!DOCTYPE *html*>**

**<html *lang*="pt-br">**

**<head>**

**<meta *charset*="UTF-8">**

**<meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">**

**<link *rel*="stylesheet" *href*="style.css">**

**<title>Teste</title>**

**</head>**

**<body>**

**<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Hic fugit impedit corrupti incidunt soluta ut molestiae consectetur nam laudantium ullam, architecto unde laborum eius pariatur temporibus optio, facere quas totam.</p>**

**</body>**

**</html>**

**HTML**

Ao estilizá-lo desta forma abaixo utilizando **PX** ela é absoluta, ou seja, ele não está se baseando em nada. Se você for nas configurações do Chrome e alterar o tamanho das letras, o seu texto ainda continuará com o mesmo tamanho. Por padrão, o navegador já vem formatado com medidas relativas.

**p {**

**font-size: 20px;**

**}**

**CSS**

O px é uma unidade fixa e estática. Cada valor especificado em pixels representa um número exato de pixels na tela.  
  
**Precisão visual:** Como o valor é fixo, os elementos serão exibidos com a mesma largura, altura, margem, etc., independentemente das configurações do navegador.

**Facilidade de controle:** Em alguns casos, pode ser útil quando você precisa de uma medição exata e controlada.

**Escalabilidade limitada:** Como os pixels são unidades fixas, eles não se ajustam bem a diferentes tamanhos de tela ou preferências de acessibilidade do usuário (como aumentar o tamanho da fonte).

**Acessibilidade reduzida:** Pessoas com deficiência visual que aumentam o zoom no navegador podem ter dificuldade em ler o conteúdo quando ele é definido com px.