

CSS
Cascading Style Sheets

Aula 01

<Módulo 04/>

Introdução ao CSS

As folhas de estilo em cascata (Cascade Style Sheets - CSS) consistem em uma linguagem de estilo utilizada para descrever a apresentação de um documento escrito em HTML ou em XML. Seu principal objetivo é descrever como os elementos devem ser mostrados na tela, no papel, na fala ou em outras mídias.



Atualmente na sua terceira versão, o CSS trouxe o que de mais avançado há em estilização, tal como:

- **Bordas arredondadas:** proporcionando a criação de designs suaves e modernos
- **Sombras:** oferecendo uma dimensão tridimensional e controle de profundidade visual
- **Gradientes:** permitindo transições suaves entre cores
- **Transições e animações:** criando páginas dinâmicas e atraentes
- **Transformações:** possibilitando a aplicação de rotações, escalas, transições e inclinações em elementos
- **Flexbox e Grid:** layouts avançados que conferem a criação de estruturas complexas e responsivas de forma eficiente
- **Fontes personalizadas:** permitindo maior diversidade tipográfica
- **Media Queries:** aprimoramentos que possibilitaram aplicação de estilos com base nas características dos dispositivos, tais como tamanho de tela, orientação e resolução

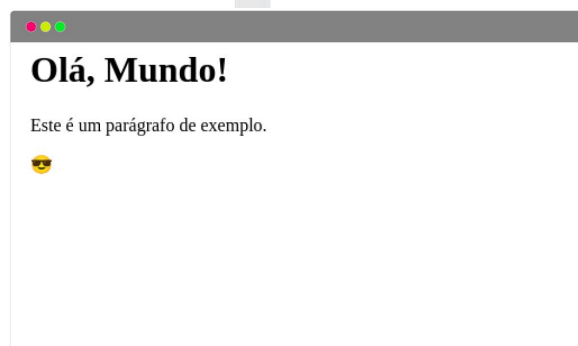
O CSS é uma das principais linguagens da **open web**, sendo padronizado em navegadores web.



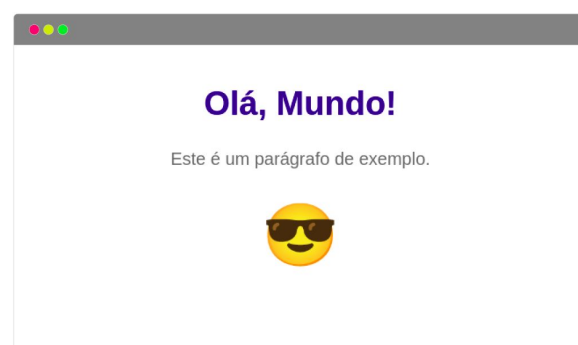
CSS e HTML

CSS e HTML são linguagens essenciais no desenvolvimento de páginas web.

Enquanto o HTML define a estrutura, hierarquia e organização do conteúdo da página, o CSS define como os elementos HTML devem ser apresentados visualmente.



Exemplo de página sem a utilização de CSS.



Aplicação de CSS à página.

Tal separação de responsabilidades facilita a criação e a manutenção das páginas.

Incorporação de CSS ao HTML

Existem várias formas de incorporar estilos ao HTML utilizando CSS.

Inline Styles

A incorporação em linha de estilos é aplicada diretamente no elemento HTML. Para isso, basta utilizar o atributo **style**:

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

Essa abordagem traz simplicidade, mas não é indicada para páginas complexas. Entre as desvantagens, estão a necessidade de repetir estilização para elementos similares e difícil manutenção.

Internal Styles

A incorporação interna de estilos é aplicada com a utilização da tag **<style>**, posicionada na tag **<head>**:

```
<!DOCTYPE html>

<html>
  <head>
    <style>
      body {
        background-color: #f0f0f0;
        font-family: Arial, sans-serif;
      }

      h1 {
        color: #333;
        text-align: center;
      }
    </style>
  </head>

  <body>
    <h1>Minha Página</h1>
    <p>Este é um parágrafo exemplo.</p>
  </body>
</html>
```

Esta abordagem é útil para páginas únicas, em que não se faz necessário aproveitar os estilos aplicados em outras páginas.

External Styles

Os estilos externos são definidos em um arquivo CSS e vinculados ao HTML através de uma tag `<link>`:

index.css

```
body {  
  background-color: #f0f0f0;  
  font-family: Arial, sans-serif;  
}  
  
h1 {  
  color: #333;  
  text-align: center;  
}
```

index.html

```
<!DOCTYPE html>  
  
<html>  
  <head>  
    <link rel="stylesheet" type="text/css" href="index.css" />  
  </head>  
  
  <body>  
    <h1>Minha Página</h1>  
    <p>Este é um parágrafo exemplo.</p>  
  </body>  
</html>
```

Esta abordagem facilita a reutilização de estilos em várias páginas.



Por dentro do assunto!

A primeira versão do CSS, conhecida como CSS1, foi proposta pelo World Wide Web Consortium (W3C) em 1996. A ideia principal do CSS1 era separar a estrutura do conteúdo (definida em HTML) da sua apresentação visual. Antes do CSS, a formatação visual era muitas vezes incorporada diretamente no HTML, tornando o código mais complexo e difícil de ser mantido.

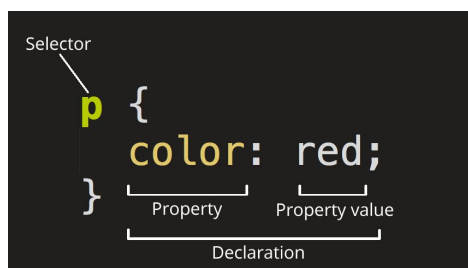
A evolução subsequente do CSS, como o CSS2 e o CSS3, trouxe melhorias significativas, introduzindo conceitos como seletores mais avançados, folhas de estilo externas, e propriedades mais sofisticadas para controle de layout e animações. Essas melhorias contribuíram para uma separação mais eficaz entre conteúdo e apresentação, facilitando o desenvolvimento e a manutenção de páginas web.

Seletores no CSS

Para aplicar estilo aos elementos HTML é necessário determinar quais elementos receberão determinado estilo. Isso é feito com **seletores**.

Os seletores no CSS são padrões que definem quais elementos HTML em uma página web serão afetados pelas regras de estilo especificadas. Em outras palavras, os seletores são usados para segmentar elementos específicos do HTML e aplicar estilos a esses elementos. São uma parte essencial das folhas de estilo CSS.

Cada seletor deve conter um conjunto de regras, ou declarações, que especificam os estilos a serem aplicados nos elementos selecionados. Cada declaração, por sua vez, consiste num par **propriedade:valor**, como mostra a figura abaixo:

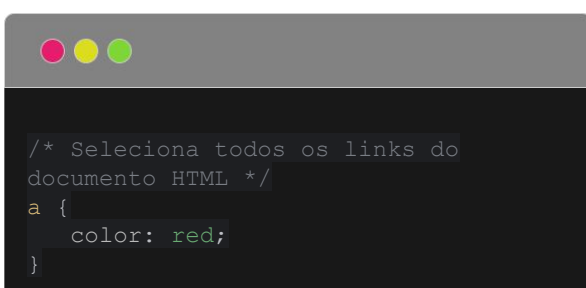


Fonte: <https://developer.mozilla.org>

Existem diversos tipos de seletores que podem ser aplicados para estilização de uma página HTML:

Seletor de tipo

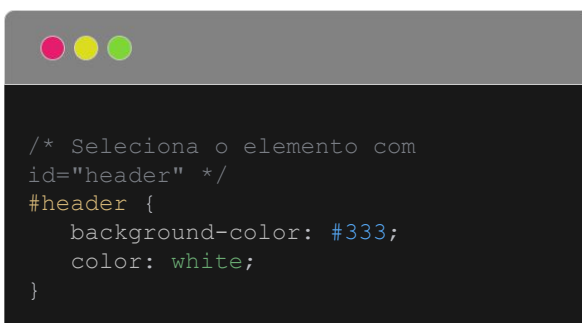
Seleciona os elementos pelo nome do nó.



```
/* Seleciona todos os links do
documento HTML */
a {
  color: red;
}
```

Seletor de ID

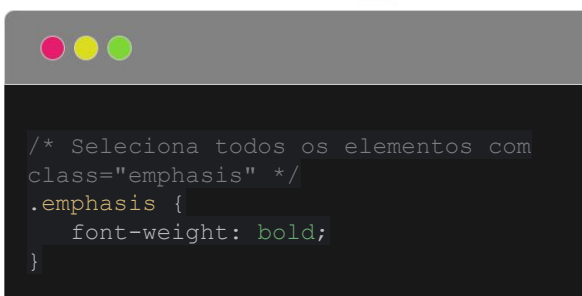
Seleciona o elemento baseado no valor de seu atributo **id**.



```
/* Seleciona o elemento com
id="header" */
#header {
  background-color: #333;
  color: white;
}
```

Seletor de classe

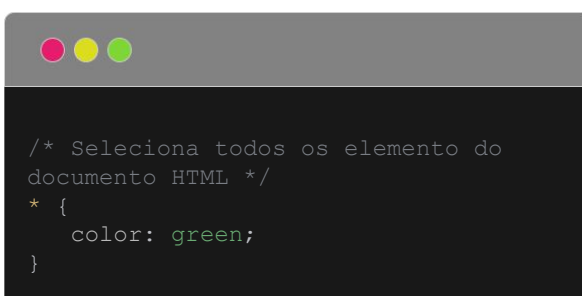
Seleciona elementos baseando-se no valor do atributo **class**.



```
/* Seleciona todos os elementos com
class="emphasis" */
.emphasis {
  font-weight: bold;
}
```

Seletor universal

Seleciona todos os elementos do documento.



```
/* Seleciona todos os elemento do
documento HTML */
* {
  color: green;
}
```

Seletor de atributos

Seleciona elementos que possuem um atributo específico. É possível definir se o elemento deve ter o valor específico no atributo ou apenas uma parte do valor.

```
/* Seleciona todos os links que
possuem o atributo title */
a[title] {
  color: purple;
}

/* Seleciona todos os links que
possuem atributo href com valor igual
a "https://example.org" */
a[href="https://example.org"]
{
  color: green;
}

/* Seleciona todos os links que
possuem atributo href que contenha
"example" em seu valor */
a[href*="example"] {
  font-size: 2em;
}

/* Seleciona todos os links que
possuem atributo href com valor
terminado em ".org" */
a[href$=".org" i] {
  font-style: italic;
}

/* Seleciona todos os links cujo
atributo class contenha o termo
"logo" */
a[class~="logo"] {
  padding: 2px;
}
```

Combinadores no CSS

Os combinadores CSS são elementos que conectam diferentes seletores, permitindo que sejam especificadas regras de estilo com base nas relações entre elementos HTML.

Combinador de adjacência (próximo irmão)

Seleciona o elemento que é adjacente e imediatamente precede o primeiro elemento especificado.

```
/* Seleciona o texto que precede e  
que seja adjacente a um título h2 */  
h2 + p {  
  color: red;  
}
```

Minha Página

Este é primeiro parágrafo exemplo.

Este é segundo parágrafo exemplo.

Este é terceiro parágrafo exemplo.

Combinador de geral de irmãos

Seleciona o elemento que precede o primeiro elemento especificado, se ambos compartilham o mesmo pai.

```
/* Seleciona o texto que precede um  
título h2 */  
h2 ~ p {  
  color: red;  
}
```

Minha Página

Este é primeiro parágrafo exemplo.

🤔 ... 🤔

Este é segundo parágrafo exemplo.

Este é terceiro parágrafo exemplo.

Combinador de de filho direto

Seleciona os elementos correspondentes ao segundo seletor, que sejam filhos direto do primeiro.

```
/* Seleciona o span que seja filho  
direto de uma div*/  
div > span {  
  color: red;  
}
```

Minha Página

Este span é filho direto da div.

Este span é filho direto de um parágrafo que está dentro da div.

Este span é filho direto da div.

Combinador de descendência

Seleciona os elementos que correspondem ao segundo seletor e são descendentes do primeiro elemento especificado.

```
/* Seleciona o span que seja desce de uma div*/  
div span {  
  color: red;  
}
```

Minha Página

Este span é filho direto da div.

Este span é filho direto de um parágrafo que está dentro da div.

Este span é filho direto da div.

Combinador de grupo

Possibilita a aplicação do mesmo conjunto de estilos a um grupo de seletores.

```
/* Seleciona os títulos h2 e os spans  
*/  
h2, span {  
  color: red;  
}
```

Minha Página

Este span é filho direto da div.

Este span é filho direto de um parágrafo que está dentro da div.

Este span é filho direto da div.

Cascata, especificidade e herança

Alguns conceitos são fundamentais para entender como o CSS é aplicado ao HTML e como possíveis **conflitos** entre as declarações de estilo são resolvidos.

Em CSS, **conflitos** referem-se aos casos em que duas ou mais regras de estilo aplicam estilos diferentes ao mesmo elemento HTML, e há a necessidade de determinar qual regra deve prevalecer.

Cascata

CSS significa **Folhas de Estilo em Cascata**. A maneira como a cascata se comporta é a chave para entender o CSS.

Cascata em CSS significa que a ordem em que os estilos são aplicados importam. Quando duas regras da mesma camada em cascata se aplicam e ambas têm igual especificidade, aquela que for definida por último na folha de estilo é a que será utilizada.

```
p {
  color: red;
}

p {
  color: blue;
}

/* O último estilo aplicado é mantido
*/
```

Minha Página

Este é o primeiro parágrafo da página.

Este é o segundo parágrafo da página.

Especificidade

É o algoritmo que o navegador usa para decidir qual valor de propriedade é aplicado a um elemento.

Se vários blocos de estilos visam a mesma propriedade com valores diferentes a um mesmo elemento, a especificidade decide o valor da propriedade que será aplicado ao elemento.

- Um seletor de elemento é o menos específico, pois seleciona todos os elementos desse tipo que aparecem em uma página
- Os seletores de pseudo-elementos têm a mesma especificidade que os seletores de elementos regulares
- Um seletor de classe é o mais específico, pois seleciona apenas os elementos que tenham um valor de atributo **class** específico
- Seletores de atributo e pseudoclasses têm o mesmo peso que uma classe

```
.text {
  color: red;
}

p {
  color: blue;
}

/* O primeiro estilo aplicado é
mantido, pois o seletor de classe é
mais específico que o seletor de tipo
*/
```

Minha Página

Este é o primeiro parágrafo da página.

Este é o segundo parágrafo da página.

Herança

Propriedade que faz com que alguns valores de propriedades definidos em elementos pai são herdados pelos elementos filhos.

```
body {  
  color: red;  
}
```

```
/* Os elementos filhos do elemento  
body herdam a propriedade color  
definida no elemento body */
```

Minha Página

Este é o primeiro parágrafo da página.

Este é o segundo parágrafo da página.

Nem todas as propriedades são herdadas:

- **color** é uma propriedade herdada
- **width**, **padding**, **margin** e **border** não são propriedades herdadas

Propriedades universais especiais podem ser utilizadas para controlar a herança em elementos:

- **inherit**: define a propriedade como sendo igual a do seu elemento pai
- **initial**: define o valor inicial de uma propriedade para o seu valor inicial
- **revert**: redefine o valor da propriedade para o valor padrão do navegador
- **revert-layer**: redefine o valor da propriedade para um valor estabelecido em uma camada em cascata anterior
- **unset**: redefine a propriedade para seu valor natural

Modelo de caixa (Box model)

O modelo em caixa consiste em uma caixa de conteúdo onde os elementos HTML são exibidos.



Camadas opcionais podem ser adicionadas à caixa de conteúdo. O modelo em caixa no CSS descreve como esses elementos funcionam juntos:

- **content**: área de conteúdo da caixa
- **padding**: borda de preenchimento da caixa
- **border**: borda da caixa
- **margin**: margem da caixa

Propriedades para controlar a área de conteúdo

A área de conteúdo da caixa pode ser controlada pelas seguintes propriedades:

- **width**: controla a largura do conteúdo
- **height**: controla a altura do conteúdo
- **max-width**: especifica o valor máximo que pode ser adicionado à largura do conteúdo
- **max-height**: especifica o valor máximo que pode ser adicionado à altura do conteúdo
- **min-width**: especifica o valor mínimo que pode ser adicionado à largura do conteúdo
- **min-height**: especifica o valor mínimo que pode ser adicionado à altura do conteúdo

```
body {  
  width: 100px;  
  height: 100px;  
  max-width: 150px;  
  max-height: 150px;  
  min-width: 50px;  
  min-height: 50px;  
}
```

Propriedades para controlar o preenchimento da caixa

A área de preenchimento da caixa pode ser controlada pelas seguintes propriedades:

- **padding**: configura a área de preenchimento da caixa
- **padding-top**: configura a área de preenchimento da caixa acima do elemento
- **padding-bottom**: configura a área de preenchimento da caixa abaixo do elemento
- **padding-left**: configura a área de preenchimento da caixa no lado esquerdo do elemento
- **padding-right**: configura a área de preenchimento da caixa no lado direito do elemento

```
body {  
  padding: 10px 5px 10px 5px; /* top right bottom left */  
}  
  
body {  
  padding-top: 10px;  
  padding-right: 5px;  
  padding-bottom: 10px;  
  padding-left: 5px;  
}
```

Propriedades para controlar a margem da caixa

A área de margem da caixa pode ser controlada pelas seguintes propriedades:

- **margin**: configura a área de margem da caixa
- **margin-top**: configura a área de margem da caixa acima do elemento
- **margin-bottom**: configura a área de margem da caixa abaixo do elemento
- **margin-left**: configura a área de margem da caixa no lado esquerdo do elemento
- **margin-right**: configura a área de margem da caixa no lado direito do elemento

```
body {  
  margin: 10px 5px 10px 5px; /* top right bottom left */  
}  
  
body {  
  margin-top: 10px;  
  margin-right: 5px;  
  margin-bottom: 10px;  
  margin-left: 5px;  
}
```

Propriedades para controlar a borda da caixa

A borda da caixa pode ser controlada pelas seguintes propriedades:

- **border**: configura a borda da caixa
- **border-top**: configura a borda da caixa acima do elemento
- **border-bottom**: configura a borda da caixa abaixo do elemento
- **border-left**: configura a borda da caixa no lado esquerdo do elemento
- **border-right**: configura a borda da caixa no lado direito do elemento
- **border-color**: configura a cor da borda da caixa
- **border-style**: configura o estilo da borda da caixa
- **border-width**: configura a espessura da borda da caixa

```
body {  
  border: solid 1px red; /* style width color */  
}  
  
body {  
  border-top: dashed 1px red;  
  border-right: dashed 1px red;  
  border-bottom: dashed 1px red;  
  border-left: dashed 1px red;  
}  
  
body {  
  border-color: green;  
  border-style: solid;  
  border-width: 1px;  
}
```


Fluxo normal do documento

O fluxo normal de um documento HTML consiste na disposição padrão dos elementos sem a aplicação de estilos CSS.

Começar com um documento sólido e bem estruturado que possa ser lido no fluxo normal é a melhor maneira de iniciar qualquer página da web. Isso garante que o conteúdo seja legível, mesmo que o usuário esteja usando um navegador muito limitado ou um dispositivo como um leitor de tela que leia o conteúdo da página.

Como o fluxo normal é projetado para tornar o documento legível, ao começar dessa forma você estará trabalhando com o documento em vez de lutar contra ele enquanto faz alterações no layout.

Exibição de elementos

Com CSS, é possível configurar como um elemento HTML é exibido na página. Há dois tipos de controle de exibição de elementos com CSS:

- **externo**: define a participação de um elemento no layout de fluxo
- **interno**: define o layout dos filhos do elemento

A propriedade **display** é utilizada para controlar a exibição dos elementos.



Exibição externa

A exibição externa de um elemento HTML controla a participação do elemento no layout de fluxo normal do documento:

- **block**: o elemento gera uma caixa de bloco, gerando quebras de linha antes e depois do elemento no fluxo normal
- **inline**: uma ou mais caixas de bloco são geradas sem quebras de linha, sendo os elementos posicionados em linha

```
#text1 {
  display: block;
  padding: 3px 20px;
  color: white;
  background-color: darkred;
  font-size: 78px;
}

#text2 {
  display: block;
  padding: 3px 20px;
  color: white;
  background-color: darkblue;
  font-size: 78px;
}

#text3 {
  display: block;
  padding: 3px 20px;
  color: white;
  background-color: darkgreen;
  font-size: 78px;
}
```

Texto 1

(quebra de linha)

Texto 2

(quebra de linha)

Texto 3

(quebra de linha)

```
#text1, #text4 {
  display: inline;
  padding: 3px 20px;
  color: white;
  background-color: darkred;
  font-size: 78px;
}

#text2, #text5 {
  display: inline;
  padding: 3px 20px;
  color: white;
  background-color: darkblue;
  font-size: 78px;
}

#text3 {
  display: inline;
  padding: 3px 20px;
  color: white;
  background-color: darkgreen;
  font-size: 78px;
}
```

Texto 1 Texto 2 Texto 3 Texto 4
Texto 5

Exibição interna

A exibição externa de um elemento HTML controla a apresentação de seu conteúdo:

- **flex**: o elemento se comporta como um elemento de bloco e apresenta seu conteúdo de acordo com o **modelo flexbox**
- **grid**: o elemento se comporta como um elemento de bloco e apresenta seu conteúdo de acordo com o **modelo de grade**
- **table**: o elemento se comporta como um elemento HTML **<table>**

Propriedade float

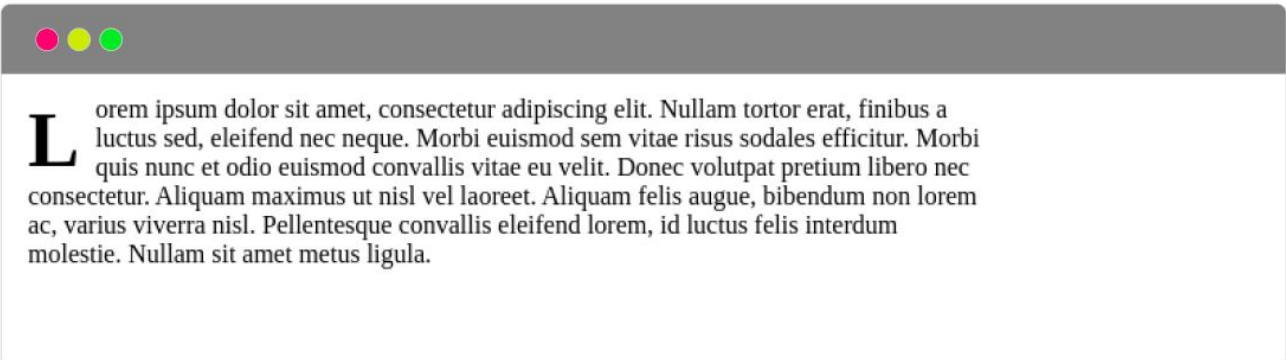
A propriedade **float** do CSS determina que um elemento deve ser retirado do seu fluxo normal e colocado ao longo do lado direito ou esquerdo do seu container, onde textos e elementos em linha irão se posicionar ao seu redor.

index.html

```
<!DOCTYPE html>

<html>
  <head>
    <style>
      p {
        width: 600px;
      }
      span {
        float: left;
        height: 50px;
        margin: 0 10px 0 0;
        font-size: 48px;
        font-weight: 600;
        font-family: 'Times New Roman', serif;
      }
    </style>
  </head>

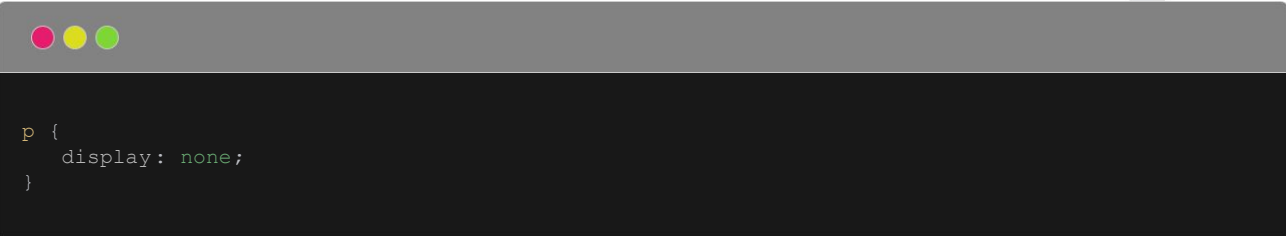
  <body>
    <p>
      <span>L</span>orem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tortor
      erat, finibus a luctus sed, eleifend necneque. Morbi euismod sem vitae risus
      sodales efficitur. Morbi quis nunc et odio euismod convallis vitae eu velit. Donec
      volutpat pretium libero nec consectetur. Aliquam maximus ut nisl vel laoreet.
      Aliquam felis augue, bibendum non lorem ac, varius viverra nisl. Pellentesque
      convallis eleifend lorem, id luctus felis interdum molestie. Nullam sit amet metus
      ligula.
    </p>
  </body>
</html>
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tortor erat, finibus a luctus sed, eleifend nec neque. Morbi euismod sem vitae risus sodales efficitur. Morbi quis nunc et odio euismod convallis vitae eu velit. Donec volutpat pretium libero nec consectetur. Aliquam maximus ut nisl vel laoreet. Aliquam felis augue, bibendum non lorem ac, varius viverra nisl. Pellentesque convallis eleifend lorem, id luctus felis interdum molestie. Nullam sit amet metus ligula.

Ocultando elementos

A propriedade **display** do CSS pode ocultar um elemento na exibição. Para isso, basta configurá-la com o valor **none**.

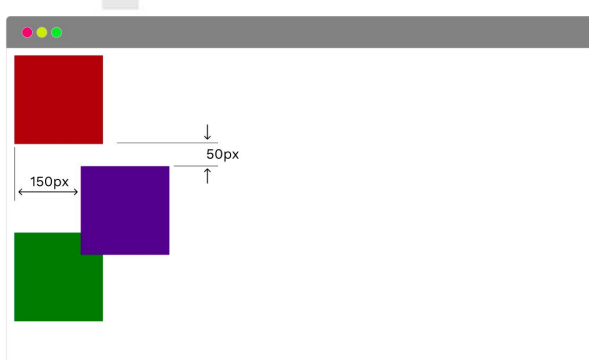


```
p {  
  display: none;  
}
```

Posicionamento de elementos

A propriedade **position** determina como um elemento é posicionado (renderizado) no documento. A propriedade possui os seguintes valores:

- **static**: o elemento é posicionado de acordo com o fluxo normal do documento
- **relative**: o elemento é posicionado de acordo com o fluxo normal do documento e, em seguida, deslocado em relação a si mesmo com base nos valores de **top**, **right**, **bottom**, e **left**

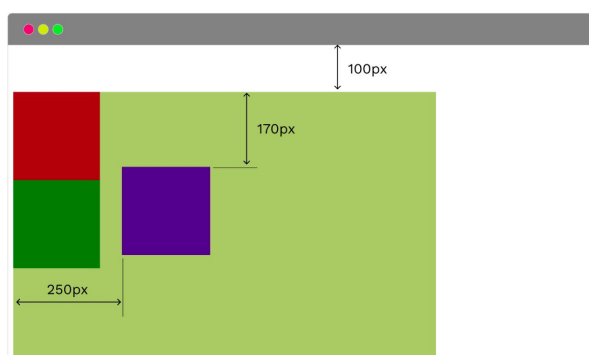


```
#box1 {
  width: 200px;
  height: 200px;
  background-color: darkred;
}

#box2 {
  width: 200px;
  height: 200px;
  background-color: darkblue;
  position: relative;
  top: 50px;
  bottom: 20px;
  left: 150px;
}

#box3 {
  width: 200px;
  height: 200px;
  background-color: darkgreen;
}
```

- **absolute**: o elemento é removido do fluxo normal do documento e nenhum espaço é criado para o elemento no layout da página, sendo posicionado em relação ao seu ancestral posicionado mais próximo, se houver, com base nos valores de **top**, **right**, **bottom**, e **left**



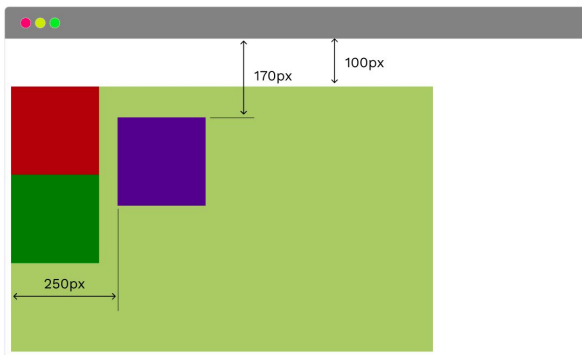
```
#container {
  width: 1000px;
  height: 700px;
  background-color: darkkhaki;
  position: absolute;
  top: 100px;
}

#box1 {
  width: 200px;
  height: 200px;
  background-color: darkred;
}

#box2 {
  width: 200px;
  height: 200px;
  background-color: darkblue;
  position: absolute;
  top: 170px;
  bottom: 20px;
  left: 250px;
}

#box3 {
  width: 200px;
  height: 200px;
  background-color: darkgreen;
}
```


- **fixed**: o elemento é removido do fluxo normal de documentos e nenhum espaço é criado para o elemento no layout da página, deslocado em relação ao container do bloco inicial definido pelo **viewport** com base nos valores de **top**, **right**, **bottom**, e **left**



```
#container {
  width: 1000px;
  height: 700px;
  background-color: darkkhaki;
  position: absolute;
  top: 100px;
}

#box1 {
  width: 200px;
  height: 200px;
  background-color: darkred;
}

#box2 {
  width: 200px;
  height: 200px;
  background-color: darkblue;
  position: fixed;
  top: 170px;
  bottom: 20px;
  left: 250px;
}

#box3 {
  width: 200px;
  height: 200px;
  background-color: darkgreen;
}
```

- **sticky**: o elemento é posicionado de acordo com o fluxo normal do documento e, em seguida, deslocado em relação ao ancestral de rolagem mais próximo com base nos valores de **top**, **right**, **bottom**, e **left**

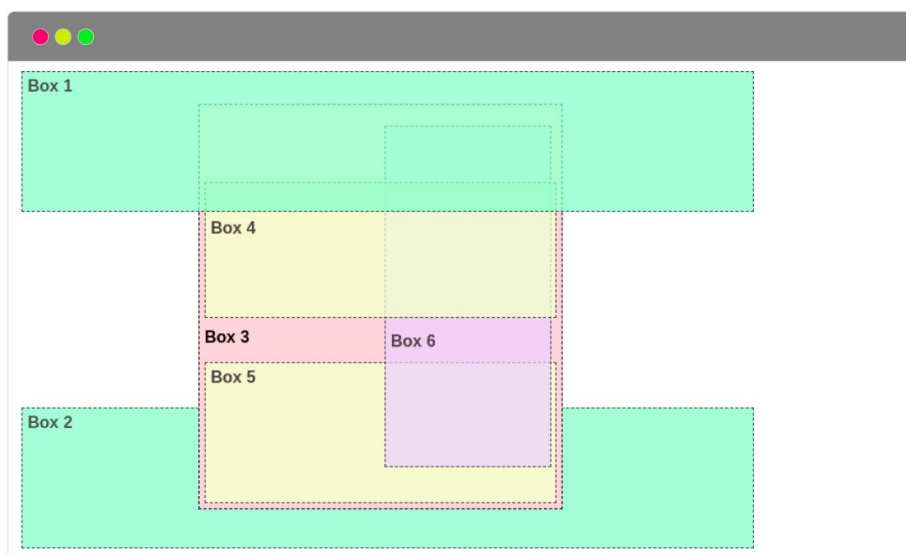
Contexto de empilhamento

O **contexto de empilhamento** é uma conceituação tridimensional de elementos HTML ao longo de um eixo z imaginário em relação ao usuário, que supostamente está de frente para a janela de visualização ou a página da web. Os elementos HTML ocupam este espaço em ordem de prioridade com base nos atributos do elemento.

A ordem de renderização de determinados elementos é influenciada pela propriedade **z-index**.

```
div {
  position: relative;
  border: 1px dashed black;
  font-family: "Worker sans",
  sans-serif;
  font-weight: 600;
  opacity: 0.7;
}
#box1 {
  z-index: 5;
  width: 700px;
  height: 125px;
  padding: 5px;
  margin-bottom: 190px;
  background-color: #C5FFDA;
}
#box2 {
  z-index: 2;
  width: 700px;
  height: 125px;
  padding: 5px;
  background-color: #C5FFDA;
}
#box3 {
  z-index: 4;
  padding: 5px;
  position: absolute;
  top: 40px;
  left: 180px;
  background-color: #FFD9DD;
  opacity: 1;
}
```

```
#box4 {
  z-index: 6;
  width: 330px;
  height: 90px;
  padding: 35px 5px 5px 5px;
  margin-top: 70px;
  margin-bottom: 10px;
  background-color: #FAF4D0;
}
#box5 {
  z-index: 1;
  width: 330px;
  height: 125px;
  margin-top: 15px;
  padding: 5px;
  background-color: #FAF4D0;
}
#box6 {
  z-index: 3;
  width: 150px;
  height: 125px;
  padding: 200px 5px 5px 5px;
  position: absolute;
  top: 20px;
  left: 180px;
  background-color: #EBD8F5;
}
```



display: flex;

Overflow

A propriedade **overflow** é usada para controlar o comportamento do conteúdo que excede as dimensões do container ao qual está contido.

Os seguintes valores podem ser usados na propriedade **overflow**:

- **visible**: o conteúdo não é cortado e pode ser renderizado para fora da caixa de conteúdo
- **hidden**: o conteúdo é cortado e nenhuma barra de rolagem é exibida

```
div {  
  width: 800px;  
  height: 200px;  
  padding: 10px;  
  color: black;  
  background-color: #FF000040;  
  overflow: visible;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tortor erat, finibus a luctus sed, eleifend nec neque. Morbi euismod sem vitae risus sodales efficitur. Morbi quis nunc et odio euismod convallis vitae eu velit. Donec volutpat pretium libero nec consectetur. Aliquam maximus ut nisl vel laoreet. Aliquam felis augue, bibendum non lorem ac, varius viverra nisl. Pellentesque convallis eleifend lorem, id luctus felis interdum molestie. Nullam sit amet metus ligula. Curabitur venenatis, elit vel pharetra fringilla, ligula arcu accumsan ipsum, posuere eleifend ligula lacus sit amet leo. Nulla in pharetra velit. Praesent tincidunt dictum tortor a molestie. Integer in erat urna. Praesent lacinia fermentum tellus ultrices malesuada. Vestibulum enim sem, posuere at lobortis at, rhoncus hendrerit mauris. Nulla ullamcorper, enim eget vehicula hendrerit, lorem lectus dignissim ex, ac tempor urna justo eget urna. Donec non lorem in nibh aliquet vehicula. Curabitur posuere eu quam sit amet fermentum. Sed vel nulla efficitur, finibus arcu in, ullamcorper ligula. Sed vel convallis sapien. Morbi vestibulum purus nec tortor vehicula, et posuere felis mollis. Vivamus nec varius massa. Curabitur dignissim diam convallis felis tempor aliquet. Maecenas eget odio maximus nisl euismod pretium. Integer semper diam sit amet eleifend accumsan. In placerat nec diam nec molestie. Sed ullamcorper turpis quis metus tempor congue. Mauris eget nisl vel nunc finibus finibus. Suspendisse semper felis ac massa euismod, sit amet eleifend dolor hendrerit. Maecenas ac auctor metus. Ut posuere placerat feugiat. Suspendisse quis faucibus nisl. Cras et lacus in lacus cursus porttitor.

```
div {  
  width: 800px;  
  height: 200px;  
  padding: 10px;  
  color: black;  
  background-color: #FF000040;  
  overflow: hidden;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tortor erat, finibus a luctus sed, eleifend nec neque. Morbi euismod sem vitae risus sodales efficitur. Morbi quis nunc et odio euismod convallis vitae eu velit. Donec volutpat pretium libero nec consectetur. Aliquam maximus ut nisl vel laoreet. Aliquam felis augue, bibendum non lorem ac, varius viverra nisl. Pellentesque convallis eleifend lorem, id luctus felis interdum molestie. Nullam sit amet metus ligula. Curabitur venenatis, elit vel pharetra fringilla, ligula arcu accumsan ipsum, posuere eleifend ligula lacus sit amet leo. Nulla in pharetra velit. Praesent tincidunt dictum tortor a molestie. Integer in erat urna. Praesent lacinia fermentum tellus ultrices malesuada. Vestibulum enim sem, posuere at lobortis at, rhoncus hendrerit mauris. Nulla ullamcorper, enim eget vehicula hendrerit, lorem lectus dignissim ex, ac tempor urna justo eget urna. Donec non lorem in nibh aliquet vehicula. Curabitur posuere eu quam sit amet fermentum. Sed vel nulla efficitur, finibus arcu in, ullamcorper ligula. Sed vel convallis sapien. Morbi vestibulum purus nec tortor vehicula, et posuere felis mollis. Vivamus nec varius massa. Curabitur dignissim diam convallis felis tempor aliquet. Maecenas eget odio maximus nisl euismod pretium. Integer semper diam sit amet eleifend accumsan. In

- **scroll**: o conteúdo é acessível através de barras de rolagem que são exibidas mesmo que o conteúdo não precise

```
div {  
  width: 800px;  
  height: 200px;  
  padding: 10px;  
  color: black;  
  background-color: #FF000040;  
  overflow: scroll;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tortor erat, finibus a luctus sed, eleifend nec neque. Morbi euismod sem vitae risus sodales efficitur. Morbi quis nunc et odio euismod convallis vitae eu velit. Donec volutpat pretium libero nec consectetur. Aliquam maximus ut nisl vel laoreet. Aliquam felis augue, bibendum non lorem ac, varius viverra nisl. Pellentesque convallis eleifend lorem, id luctus felis interdum molestie. Nullam sit amet metus ligula. Curabitur venenatis, elit vel pharetra fringilla, ligula arcu accumsan ipsum, posuere eleifend ligula lacus sit amet leo. Nulla in pharetra velit. Praesent tincidunt dictum tortor a molestie. Integer in erat urna. Praesent lacinia fermentum tellus ultrices malesuada. Vestibulum enim sem, posuere at lobortis at, rhoncus hendrerit mauris. Nulla ullamcorper, enim eget vehicula hendrerit, lorem lectus dignissim ex, ac tempor urna justo eget urna. Donec non lorem in nibh aliquet vehicula. Curabitur posuere eu quam sit amet fermentum. Sed vel nulla efficitur, finibus arcu in, ullamcorper ligula. Sed vel convallis sapien. Morbi vestibulum purus nec tortor vehicula, et posuere felis mollis. Vivamus nec varius massa. Curabitur dignissim diam convallis felis tempor aliquet. Maecenas eget odio maximus nisl euismod pretium. Integer semper diam sit

barras de rolagem

- **auto**: depende do navegador utilizado - Alguns navegadores exibem barras de rolagem apenas quando necessário

```
div {  
  width: 800px;  
  height: 200px;  
  padding: 10px;  
  color: black;  
  background-color: #FF000040;  
  overflow: auto;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tortor erat, finibus a luctus sed, eleifend nec neque. Morbi euismod sem vitae risus sodales efficitur. Morbi quis nunc et odio euismod convallis vitae eu velit. Donec volutpat pretium libero nec consectetur. Aliquam maximus ut nisl vel laoreet. Aliquam felis augue, bibendum non lorem ac, varius viverra nisl. Pellentesque convallis eleifend lorem, id luctus felis interdum molestie. Nullam sit amet metus ligula. Curabitur venenatis, elit vel pharetra fringilla, ligula arcu accumsan ipsum, posuere eleifend ligula lacus sit amet leo. Nulla in pharetra velit. Praesent tincidunt dictum tortor a molestie. Integer in erat urna. Praesent lacinia fermentum tellus ultrices malesuada. Vestibulum enim sem, posuere at lobortis at, rhoncus hendrerit mauris. Nulla ullamcorper, enim eget vehicula hendrerit, lorem lectus dignissim ex, ac tempor urna justo eget urna. Donec non lorem in nibh aliquet vehicula. Curabitur posuere eu quam sit amet fermentum. Sed vel nulla efficitur, finibus arcu in, ullamcorper ligula. Sed vel convallis sapien. Morbi vestibulum purus nec tortor vehicula, et posuere felis mollis. Vivamus nec varius massa. Curabitur dignissim diam convallis felis tempor aliquet. Maecenas eget odio maximus nisl euismod pretium. Integer semper diam sit

barra de rolagem

Estilização de textos

O texto é apresentado dentro da caixa de conteúdo do elemento. O conteúdo de texto se comporta efetivamente como uma série de elementos embutidos, sendo dispostos em linhas adjacentes umas às outras e não criando quebras de linha até que o final da linha seja alcançado.

As propriedades CSS usadas para estilizar texto geralmente se enquadram em duas categorias:

- **fonte:** propriedades que afetam a fonte de um texto (família, tamanho ou se está em negrito, itálico, etc.)
- **layout:** propriedades que afetam o espaçamento e outros recursos de layout do texto, permitindo a manipulação (espaço entre linhas e letras e como o texto é alinhado dentro da caixa de conteúdo)

Fonte

Cor

A propriedade **color** define a cor do conteúdo de primeiro plano dos elementos selecionados (textos, sublinhados ou sobrelinhas). Aceita quando valor de **unidade de cores CSS**.

Família

A propriedade **font-family** define a fonte do texto do elemento.

O navegador só aplicará uma fonte se ela estiver disponível na máquina em que o site está sendo acessado; caso contrário, usará apenas uma **fonte padrão** do navegador.

Existe uma lista de fontes seguras (geralmente disponíveis em todos os sistemas operacionais):

- Arial
- Courier New
- Georgia
- Times New Roman
- Trebuchet MS
- Verdana

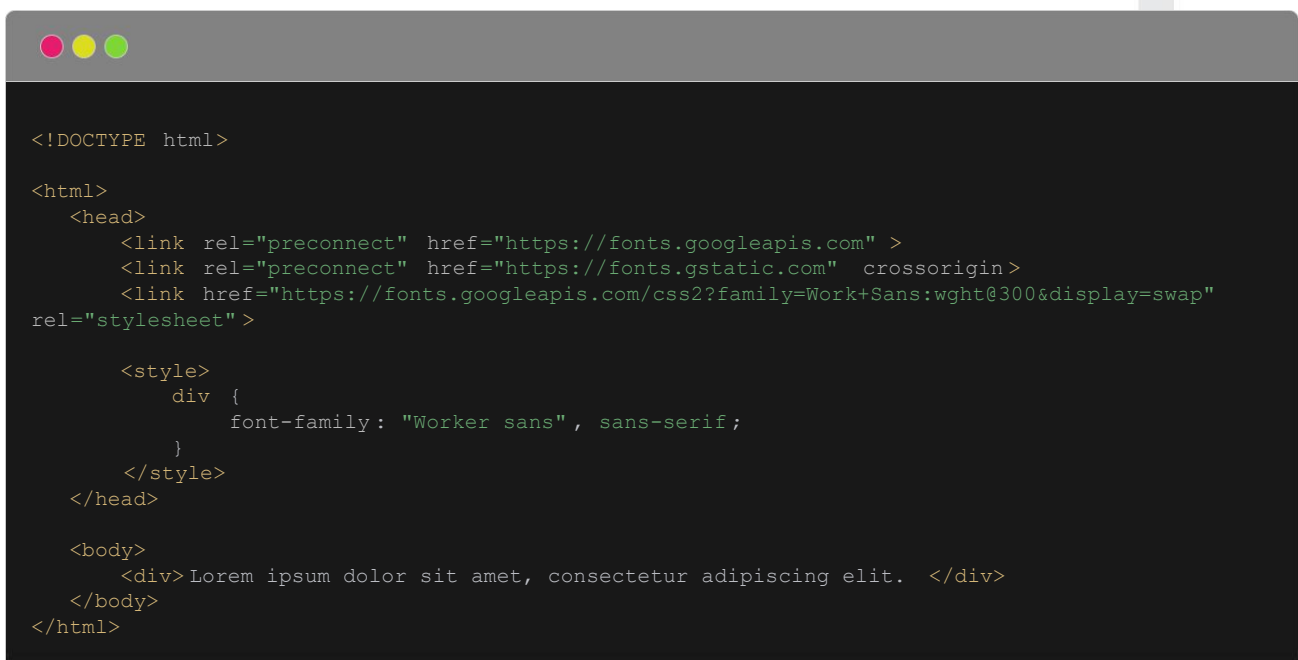
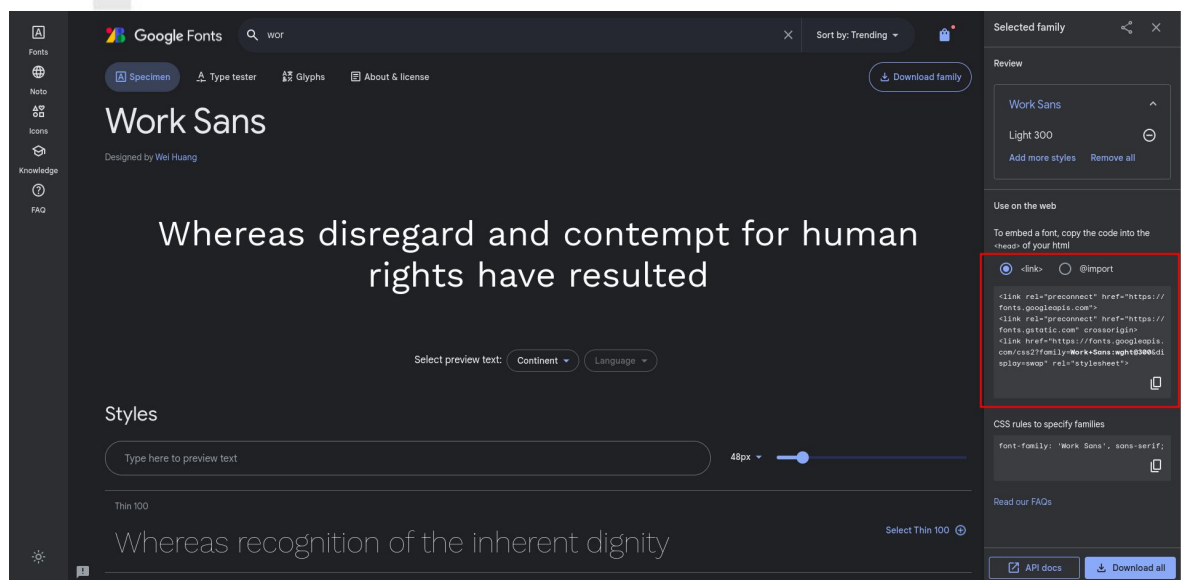
Além disso, há uma lista de fontes padrão que podem ser utilizadas:

- serif
- sans-serif
- monospace
- cursive
- fantasy

Google Fonts

O **Google Fonts** é um serviço que disponibiliza uma ampla variedade de fontes gratuitas para uso em projetos web. Ele permite que desenvolvedores e designers escolham e incorporem fontes em seus sites, melhorando a estética tipográfica e a experiência do usuário.

Para utilizar uma fonte disponível no **Google Fonts**, basta selecioná-la e referenciá-la no *header* do documento HTML.



Tamanho da fonte

A propriedade **font-size** define o tamanho da fonte do texto do elemento.

As unidades mais comuns, utilizadas não só para tamanho de fontes, são:

- **px**: número de pixels de altura
- **em**: é igual ao tamanho da fonte definido no elemento pai do elemento, em proporção (**1em** = 1 vez o tamanho da fonte, **2em** = duas vezes o tamanho da fonte, etc.)
- **rem**: é igual ao tamanho da fonte definido no elemento raiz do documento, ou seja **<html>**, também em proporção

O tamanho da fonte de um elemento é herdado do elemento pai desse elemento.

Peso ou ênfase do texto

As propriedades utilizadas para configuração do peso ou ênfase do texto são:

- **font-style**: ativa ou desativa o itálico do texto:
 - **normal** ou **italic**: desativa ou ativa o itálico
 - **oblique**: define uma versão simulada de uma fonte em itálico
- **font-weight**: define o nível de negrito do texto:
 - **normal** ou **bold**: define o peso como normal ou negrito
 - **lighter** ou **bolder**: define o negrito um peso mais leve ou mais pesado do que o do elemento pai
 - **100 - 900**: controle mais refinado com números que definem a escala de negrito aplicada
- **text-transform**: define transformações no texto:
 - **none**: não aplica transformação
 - **uppercase**: transforma o texto em maiúsculas
 - **lowercase**: transforma o texto em minúsculas
 - **capitalize**: transforma todas as palavras para que a primeira letra seja maiúscula
 - **full-width**: transforma todos os glifos para serem escritos dentro de um quadrado de largura fixa, semelhante a uma fonte monoespaçada
- **text-decoration**: ativa ou desativa decorações de textos em fontes, aceitando vários valores de uma vez:
 - **none**: não aplica decoração
 - **underline**: sublinha o texto
 - **overline**: adiciona uma linha acima do texto
 - **line-through**: adiciona um tachado no texto

Sombras

A propriedade **text-shadow** pode ser utilizada para aplicar sombras ao texto:

```
h1 {  
  font-size: 72px;  
  text-shadow:  
    3px 3px 3px red,  
    6px 6px 3px black;  
}
```

Tommy the cat

Layout de texto

Alinhamento

A propriedade **text-align** é usada para controlar como o texto é alinhado dentro da caixa de conteúdo que o contém:

- **left**: justifica o texto à esquerda
- **right**: justifica o texto à direita
- **center**: centraliza o texto
- **justify**: espalha o texto, variando os espaços entre as palavras para que todas as linhas do texto tenham a mesma largura

Altura da linha

A propriedade **text-align** define a altura de cada linha de texto.

Pode receber a maioria das unidades de comprimento e tamanho, ou um número sem unidade, que se comporta com um multiplicador do valor da propriedade **font-size** do texto.

Espaçamento entre letras e palavras

As propriedades **letter-spacing** e **word-spacing** permitem definir o espaçamento entre letras e palavras do texto.

Aplicação de background

A propriedade **background** define as propriedades de estilo de fundo dos elementos, desde cor até imagens, origem, tamanho e método de repetição.

As seguintes propriedades podem ser configuradas para o background dos elementos:

- **background-attachment**: define se a posição de uma imagem de fundo é fixa ou rola com o bloco que a contém
- **background-clip**: define se o plano de fundo de um elemento se estende abaixo de sua caixa de borda, caixa de preenchimento ou caixa de conteúdo
- **background-color**: define a cor de fundo de um elemento
- **background-image**: define uma ou mais imagens de fundo em um elemento
- **background-origin**: define a origem do plano de fundo: desde o início da borda, dentro da borda ou dentro do preenchimento
- **background-position**: define a posição inicial de cada imagem de fundo
- **background-repeat**: define como as imagens de fundo são repetidas
- **background-size**: define o tamanho da imagem de fundo do elemento

Uso de cores no CSS









Uma cor no CSS pode ser descrita em uma das seguintes formas:

- usando uma **palavra-chave**
- usando o sistema de **coordenada-cúbica RGB**
- usando o sistema de **coordenada-cilíndrica HSL**

Palavras-chave de cores

Palavras-chave de cores são identificadores de representam uma cor específica. Os principais valores são:

black	
silver	
gray	
white	
maroon	
red	
purple	
fuchsia	
green	

lime	
olive	
yellow	
navy	
blue	
teal	
aqua	
orange	

RGB

RGB é uma abreviação para "Red, Green, Blue". Trata-se de um modelo de cores aditivo amplamente utilizado em dispositivos eletrônicos.

No modelo RGB, as cores são criadas combinando diferentes intensidades dessas três cores primárias.

No CSS, cores RGB podem ser definidas em duas nomenclaturas:

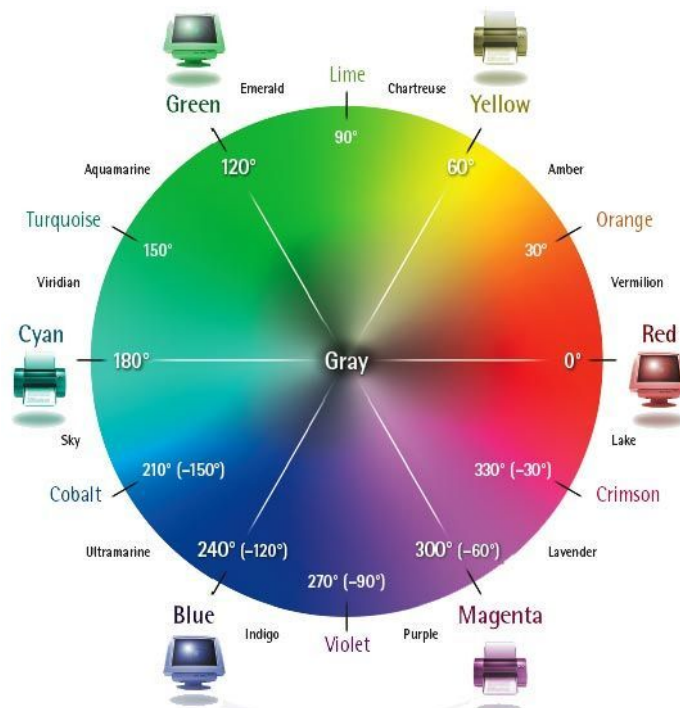
- **notação hexadecimal:** são combinações de três caracteres hexadecimais (00 a ff), mais um fator alfa, que representa a opacidade da cor:
 - **#ffffff:** representa a cor branco
 - **#ff000030:** representa a cor vermelho com um fator de 30% de opacidade
 - **#fff:** representa a cor branco (**notação de três dígitos**)
 - **#f003:** representa a cor vermelho com um fator de 30% de opacidade (**notação de quatro dígitos**)

- **notação funcional:** a cor é definida por uma das funções, `rgb()` ou `rgba()`, com valores numéricos para cada participação de cor primária (0 a 255) e canal alfa (0 a 1):
 - `rgb(255,255,255)`: representa a cor branco
 - `rgba(255,0,0,3)`: representa a cor vermelho com um fator de 30% de opacidade

HSL

HSL define as cores a partir de três componentes principais:

- **Matriz:** tonalidade da cor, em graus, no círculo de cores:
 - **0° ou 360°:** vermelho
 - **120°:** verde
 - **240°:** azul



- **Saturação:** intensidade ou pureza da cor, representada em porcentagem:
 - valores altos indicam cores mais vibrantes e saturadas
 - valores baixos resultam em tons de cinza ou tons mais desbotados
- **Luminosidade:** quantidade de luz na cor, representada em porcentagem:
 - valores altos resultam em cores mais claras
 - valores baixos levam a cores mais escuras

A função **hsl()** pode ser usada para definição de uma cor no formato HSL. A função **hsla()** adiciona parâmetro para o canal alfa.

Opacidade e transparência

A propriedade **opacity** especifica a transparência de um elemento, isto é, o grau no qual o background atrás do elemento é sobreposto.

Os valores são dados de 0 a 1 (0 a 100%).

O valor aplica-se ao elemento como um todo, incluindo seu conteúdo.

Círculo cromático

O círculo cromático possui as doze diferentes cores, que ajudam a visualizar as cores primárias, secundárias e terciárias que formam o espectro visível.



- **cores primárias:** vermelho, amarelo e azul
- **cores secundárias:** laranja, verde e roxo
- **cores terciárias:** vermelho-arroxado, vermelho-alaranjado, amarelo-alaranjado, amarelo-esverdeado, azul-arroxado e azul-esverdeado

Branco e preto são resultado da presença ou ausência de luz, respectivamente.

Harmonia das cores

A harmonia das cores estuda tipos de composições entre cores para compreender quais as combinações podem ser mais adequadas e atrativas para um determinado uso.

Harmonia monocromática

A harmonia monocromática é um esquema de cores que se baseia em uma única cor e suas variações em termos de tom, saturação e luminosidade.



Harmonia análoga

A harmonia análoga é um esquema de cores que envolve a seleção de cores que estão próximas umas das outras no círculo cromático, tendo como base uma das cores primárias.



Harmonia complementar

A harmonia complementar é um esquema de cores que envolve a escolha de cores que estão localizadas diretamente opostas umas às outras no círculo cromático.



Harmonia triádica

A harmonia monocromática é um esquema de cores que envolve a escolha de três cores igualmente espaçadas ao redor do círculo cromático.



Contraste e legibilidade

Contraste e legibilidade são aspectos críticos do design de páginas web, pois desempenham um papel fundamental na experiência do usuário. Garantir um bom contraste e legibilidade contribui para uma navegação mais eficaz e uma compreensão mais fácil do conteúdo.

- **Texto vs fundo:** assegure-se de que o texto tenha um contraste suficiente em relação ao fundo, texto de cor escura sobre fundo claro ou vice-versa
- **Elementos interativos:** botões, links e outros elementos interativos devem ser distinguíveis do restante do conteúdo, com a utilização de cores contrastantes
- **Cores iguais:** evite combinações de cores muito próximas ou praticamente idênticas, pois podem causar confusão e dificultar a leitura
- **Tamanho adequado:** certifique-se de que o tamanho do texto seja suficientemente legível, considerando diferentes dispositivos e tamanhos de tela
- **Destaque de títulos:** utilize tamanhos de fonte diferentes e estilos de texto (negrito, itálico) para destacar hierarquias de informações, como títulos e subtítulos
- **Cor e estilo de fonte:** varie a cor e o estilo de fonte para indicar diferentes tipos de conteúdo ou para destacar informações importantes
- **Acessibilidade nas cores:** certifique-se de que o design atenda a padrões de acessibilidade, garantindo que pessoas com deficiência visual possam ler o conteúdo com facilidade