

<b>CSS INLINE.....</b>	<b>4</b>
VANTAGENS DO CSS INLINE: .....	4
DESVANTAGENS DO INLINE CSS: .....	4
<b>CSS INTERNO .....</b>	<b>5</b>
VANTAGENS DE CSS INTERNO:.....	5
DESVANTAGENS DE CSS INTERNO:.....	5
<b>CSS EXTERNO .....</b>	<b>6</b>
VANTAGENS DE CSS EXTERNO: .....	6
DESVANTAGENS DE CSS EXTERNO:.....	6
<b>SELETORES SIMPLES DE ELEMENTOS CSS.....</b>	<b>7</b>
O SELETOR DE ELEMENTOS CSS .....	7
O SELETOR DE ID CSS .....	7
O SELETOR DE CLASSE CSS.....	7
O SELETOR UNIVERSAL CSS .....	8
O SELETOR DE AGRUPAMENTO CSS.....	8
<b>CORES EM CSS .....</b>	<b>9</b>
NOMES DE CORES CSS .....	9
CORES RGB .....	9
VALOR RGBA .....	9
CORES HEX (HEXADECIMAL) .....	10
CORES HSL.....	10
VALOR HSLA.....	10
<b>BACKGROUND .....</b>	<b>11</b>
BACKGROUND-COLOR.....	11
BACKGROUND-IMAGE.....	11
BACKGROUND-REPEAT .....	11
BACKGROUND-POSITION .....	12

BACKGROUND-ATTACHMENT .....	12
PROPRIEDADE BACKGROUND ABREVIADA .....	13
<b>UNIDADE DE MEDIDAS - ABSOLUTAS.....</b>	<b>14</b>
PÍXELS (PX).....	14
POINTS (PT) .....	15
POLEGADAS IN (INCHES).....	15
CENTÍMETRO E MILÍMETRO (CM / MM) .....	15
PAICA (PC) .....	16
<b>UNIDADES DE MEDIDAS - RELATIVAS .....</b>	<b>17</b>
EMS (EM) .....	17
REMS (REM, "ROOT EM") .....	18
PORCENTAGEM (%) .....	19
EX E CH.....	19
VW (VIEWPORT WIDTH).....	19
VH (VIEWPORT HEIGHT) .....	20
VMIN (VIEWPORT MINIMUN) .....	20
VMAX (VIEWPORT MAXIMUM).....	21
<b>MARGIN E PADDING .....</b>	<b>22</b>
MARGINS .....	22
CSS PADDING.....	22
MARGIN E PADDING COM PROPRIEDADES ABREVIADAS .....	23
<b>BORDER .....</b>	<b>24</b>
BORDER-STYLE .....	24
BORDER-WIDTH.....	25
BORDER-COLOR .....	25
BORDER-RADIUS .....	25
<b>BOX MODEL .....</b>	<b>26</b>
<b>OUTLINE .....</b>	<b>27</b>
OUTLINE-STYLE.....	27
OUTLINE-WIDTH .....	27

OUTLINE-COLOR.....	28
OUTLINE-OFFSET .....	28
<b>AJUSTES DE TEXTO .....</b>	<b>29</b>
COLOR .....	29
BACKGROUND-COLOR.....	29
ALINHAMENTO E DIREÇÃO DE TEXTO .....	29
TEXT-DECORATION-LINE .....	30
TEXT-TRANSFORM .....	30
ESPAÇAMENTO DE TEXTO .....	31
TEXT-SHADOW .....	31
<b>FONTES .....</b>	<b>32</b>
FONT-FAMILY .....	32
FONT-SIZE .....	32
FONT-STYLE .....	33
FONT-VARIANT .....	33
FONT-WEIGHT .....	34
FONT .....	34
<b>FONTES CUSTOMIZADAS .....</b>	<b>35</b>
FONTES DO GOOGLE.....	35
@FACE-FONT .....	35
<b>LINK .....</b>	<b>36</b>
<b>LISTAS .....</b>	<b>37</b>
LIST-STYLE-TYPE .....	37
LIST-STYLE-IMAGE.....	38
LIST-STYLE-POSITION .....	38
LIST-STYLE .....	39

## CSS INLINE

O CSS inline é usado para dar estilo a um elemento HTML específico. Para este estilo de CSS, você somente vai precisar adicionar o atributo style para cada tag HTML, sem usar os seletores. Este tipo de CSS não é realmente recomendado, já que cada tag HTML precisa ser estilizada de maneira individual. Gerenciar o seu site pode se tornar uma tarefa bem difícil se você só usa o CSS inline. Contudo, o CSS inline no HTML pode ser útil para algumas situações. Por exemplo, em casos onde você não tem acesso aos arquivos CSS ou precisa aplicar estilos para um elemento único. Vamos dar uma olhada num exemplo. Aqui, nós adicionamos um CSS inline para as tags <p> e <h1>:

```
<h1 style="color:white;padding:30px;">Hostinger Tutorials</h1>
```

```
<p style="color:white;">Something usefull here.</p>
```

### **Vantagens do CSS Inline:**

Você pode inserir elementos CSS de maneira fácil e rápida numa página HTML. É por isso que esse método é útil para testar e pré-visualizar mudanças, assim como executar correções rápidas no seu site.

Você não precisa criar e fazer upload de um documento separado como no estilo externo.

### **Desvantagens do Inline CSS:**

Adicionar regras CSS para cada elemento HTML consome muito tempo e faz a sua estrutura HTML ficar bagunçada.

Estilizar múltiplos elementos pode afetar o tamanho da sua página o tempo para download.

## CSS INTERNO

O CSS interno ou incorporado requer que você adicione a tags `<style>` na seção `<head>` do seu documento HTML. Este estilo de CSS é um método efetivo de estilizar uma única página. Contudo, usar esse estilo em múltiplas páginas pode consumir muito tempo, já que você precisa definir as regras CSS para cada página do seu site. `<style> body { background-color: blue; } h1 { color: red; padding : 60px; } </style> </head>`.

### **Vantagens de CSS Interno:**

Classes e seletores de IDs podem ser usados por stylesheet interno. Confira um exemplo abaixo:

Não há necessidade de carregar vários arquivos. HTML e CSS podem estar no mesmo arquivo.

### **Desvantagens de CSS Interno:**

Adicionar o código para o documento HTML pode aumentar o tamanho da página e o tempo de carregamento.

## CSS EXTERNO

Com o CSS externo, você vai linkar as páginas da internet com um arquivo .css externo, que você pode criar usando qualquer editor de texto no seu dispositivo (ex.: Notepad++).

Este tipo de CSS é um método mais eficiente, especialmente se você está estilizando um site grande. Ao editar um arquivo .css, você pode modificar um site inteiro de uma só vez. O CSS externo acontece quando é criado um novo arquivo .css com um editor de texto e então adicione regras de estilo, logo após isso na seção <head> da planilha HTML, é adicionada uma referência para o arquivo .css logo depois da tag <title>: <link rel="stylesheet" type="text/css" href="style.css" /> (style.css é o nome do arquivo de estilo css, caso seja outro, o mesmo deve ser alterado).

### **Vantagens de CSS Externo:**

Como o código CSS está num documento separado, os seus arquivos HTML terão uma estrutura mais limpa e serão menores

O mesmo arquivo .css pode ser usado em várias páginas.

### **Desvantagens de CSS Externo:**

Até que o CSS externo seja carregado, a página pode não ser processada corretamente.

Fazer o upload ou links para múltiplos arquivos CSS pode aumentar o tempo de download do seu site.

## SELETORES SIMPLES DE ELEMENTOS CSS

Seletores CSS são usados para "encontrar" (ou selecionar) os elementos HTML que você deseja estilizar. Podemos dividir os seletores CSS em cinco categorias: Seletores simples (selecione elementos com base no nome, id, classe); Seletores de combinação (selecione elementos com base em um relacionamento específico entre eles); Seletores de pseudo-classe (selecionar elementos com base em um determinado estado); Seletores de pseudo-elementos (selecione e estilize uma parte de um elemento); Seletores de atributo (selecione elementos com base em um atributo ou valor de atributo). Os seletores simples estudados nesse tópico são os mais básicos dos seletores.

### O seletor de elementos CSS

O seletor de elemento seleciona elementos HTML com base no tipo do elemento. Aqui, todos os elementos `<p>` na página serão alinhados ao centro, com uma cor de texto vermelha: `p { text-align: center; color: red; }`

### O seletor de id CSS

O seletor id usa o atributo id de um elemento HTML para selecionar um elemento específico. O id de um elemento é único dentro de uma página, então o seletor de id é usado para selecionar um elemento único! Para selecionar um elemento com um id específico, escreva um caractere hashtag (#), seguido do id do elemento. A regra CSS abaixo será aplicada ao elemento HTML com `id="para1"`: `#para1 { text-align: center; color: red }.`

### O seletor de classe CSS

O seletor de classe seleciona elementos HTML com um atributo de classe específico. Para selecionar elementos com uma classe específica, escreva um caractere ponto (.), seguido do nome da classe. Neste exemplo, todos os elementos HTML com `class="center"` serão vermelhos e alinhados ao centro: `center {text-align: center; color: red;}`

### **O seletor universal CSS**

O seletor universal (\*) seleciona todos os elementos HTML na página. A regra CSS abaixo afetará todos os elementos HTML na página: `* { text-align: center; color: blue;}`

### **O seletor de agrupamento CSS**

O seletor de agrupamento seleciona todos os elementos HTML com as mesmas definições de estilo. Veja o seguinte código CSS (os elementos h1, h2 e p têm as mesmas definições de estilo): Será melhor agrupar os seletores, para minimizar o código. Para agrupar seletores, separe cada seletor com uma vírgula: `h1, h2, p { text-align: center; color: red;}`



## CORES EM CSS

As cores em CSS são especificadas usando nomes de cores predefinidos ou valores RGB, HEX, HSL, RGBA, HSLA.

### Nomes de cores CSS

Em CSS, uma cor pode ser especificada usando um nome de cor predefinido. É possível encontrar esses nomes no site [https://www.w3schools.com/colors/colors\\_names.asp](https://www.w3schools.com/colors/colors_names.asp)

Ex: `<h1 style="background-color:DodgerBlue;">Hello World</h1>` **HELLO WORLD**

### Cores RGB

Um valor de cor RGB representa fontes de luz VERMELHA (red), VERDE (green) e AZUL (blue). Em CSS, uma cor pode ser especificada como um valor RGB, usando esta fórmula: **rgb (vermelho, verde, azul)**. Cada parâmetro (vermelho, verde e azul) define a intensidade da cor entre 0 e 255. Por exemplo, `rgb(255, 0, 0)` é exibido como vermelho, porque vermelho é definido como seu valor mais alto (255) e os outros são definidos como 0. Para exibir preto, defina todos os parâmetros de cor para 0, assim: `rgb(0, 0, 0)`. Para exibir branco, defina todos os parâmetros de cor para 255, assim: `rgb(255, 255, 255)`.

Ex: `<h1 style="background-color: rgb(0, 0, 255);">Hello World</h1>` **HELLO WORLD**

### Valor RGBA

Os valores de cor RGBA são uma extensão dos valores de cor RGB com um canal alfa - que especifica a opacidade de uma cor. Um valor de cor RGBA é especificado com: **rgba(vermelho, verde, azul, alfa)**. O parâmetro alfa é um número entre 0,0 (totalmente transparente) e 1,0 (nada transparente):

## Cores HEX (Hexadecimal)

Uma cor hexadecimal é especificada com: #RRGGBB, onde os inteiros hexadecimais RR (vermelho), GG (verde) e BB (azul) especificam os componentes da cor. Em CSS, uma cor pode ser especificada usando um valor hexadecimal na forma: **#rrggbb** \_ Onde rr (vermelho), gg (verde) e bb (azul) são valores hexadecimais entre 00 e ff (o mesmo que decimal 0-255). Por exemplo, #ff0000 é exibido como vermelho, porque o vermelho é definido com o valor mais alto (ff) e os outros são definidos com o valor mais baixo (00). Para exibir preto, defina todos os valores para 00, assim: #000000. Para exibir branco, defina todos os valores para ff, assim: #ffffff. Ex <h1 style="background-color:#0000ff;">HELLO WORLD</h1> **HELLO WORLD**.

## CORES HSL

HSL significa tonalidade(hue), saturação(saturation), luminosidade(lightness). Em CSS, uma cor pode ser especificada usando (HSL) na forma: **hsl (tonalidade, saturação, luminosidade)**. Tonalidade é um grau na roda de cores de 0 a 360. 0 é vermelho, 120 é verde e 240 é azul. A saturação é um valor percentual, 0% significa um tom de cinza e 100% é a cor completa. A luminosidade também é uma porcentagem, 0% é preto, 50% não é claro nem escuro, 100% é branco. EX <h1 style="background-color:hsl(240, 100%, 50%);">HELLO WORLD</h1> **HELLO WORLD**.

## Valor HSLA

Os valores de cor HSLA são uma extensão dos valores de cor HSL com um canal alfa - que especifica a opacidade de uma cor. Um valor de cor HSLA é especificado com: **hsla (tonalidade, saturação, luminosidade, alfa)**. O parâmetro alfa é um número entre 0,0 (totalmente transparente) e 1,0 (nada transparente):

## BACKGROUND

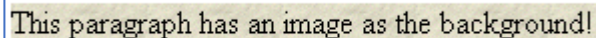
As propriedades de background CSS são usadas para adicionar efeitos de plano de fundo para elementos.

### background-color

A propriedade background-color especifica a cor de fundo de um elemento. Com CSS, uma cor é mais frequentemente especificada por: um nome de cor válido - como "red"; um valor HEX - como "#ff0000"; um valor RGB - como "rgb(255,0,0)". Ex: `body {background-color: lightblue;}`.

### background-image

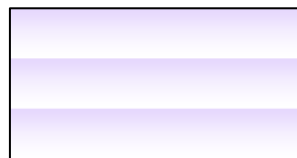
A propriedade background-image especifica uma imagem a ser usada como plano de fundo de um elemento. Por padrão, a imagem é repetida para cobrir todo o elemento. É importante observar ao usar uma imagem de fundo, use uma imagem que não perturbe o texto. Ex: `p { background-image: url("paper.gif" ) ; }.`



This paragraph has an image as the background!

### background-repeat

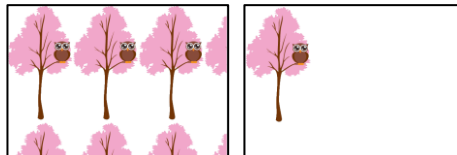
Por padrão, a propriedade background-image repete uma imagem na horizontal e na vertical, em alguns casos essa repetição fica desproporcional como no exemplo: `body { background-image: url("gradient_bg.png"); }`



Se a imagem acima for repetida apenas horizontalmente ( `background-repeat: repeat-x;`), o plano de fundo ficará melhor: `body { background-image: url("gradient_bg.png"); background-repeat: repeat-x;}`



Para mostrar a imagem de fundo apenas uma vez também é especificado pela propriedade `background-repeat (no-repeat)`: `body { background-image: url("img_tree.png"); background-repeat: no-repeat; }`



### **background-position**

A propriedade `background-position` é usada para especificar a posição da imagem de fundo. Exemplo: Posicione a imagem de fundo no canto superior direito: `body { background-image: url("img_tree.png"); background-repeat: no-repeat; background-position: right top; }`



### **background-attachment**

A propriedade `background-attachment` especifica se a imagem de fundo deve rolar ou ser fixa (não rolará com o resto da página). Exemplo: Especifique que a imagem de fundo deve ser corrigida: `body { background-image: url("img_tree.png"); background-repeat: no-repeat; background-attachment: fixed; }`. Exemplo: Especifique que a imagem de fundo deve rolar com o resto da página: `body { background-image: url("img_tree.png"); background-repeat: no-repeat; background-attachment: scroll; }`

### **propriedade background abreviada**

Para encurtar o código, também é possível especificar todas as propriedades do plano de fundo em uma única propriedade. Isso é chamado de propriedade abreviada. Para fazer isso deve-se usar a propriedade abreviada `background:`. Exemplo: Use a propriedade abreviada para definir as propriedades do plano de fundo em uma declaração: `body { background: #ffffff url("img_tree.png") no-repeat right top;}`

Ao usar a propriedade abreviada, a ordem dos valores da propriedade é:

1. `background-color`
2. `background-image`
3. `background-repeat`
4. `background-attachment`
5. `background-position`

Não importa se um dos valores da propriedade está faltando, desde que os outros estejam nesta ordem.

## UNIDADE DE MEDIDAS - ABSOLUTAS

São medidas que não estão referenciadas a qualquer outra unidade, ou seja, não dependem de um valor de referência. São unidades de medidas definidas pela física, como o píxel, centímetro, metro, etc. Essas medidas são fixas e não mudam de acordo com as especificações do dispositivo. Esse tipo de medida é indicada para quando conhecemos perfeitamente as características físicas e as configurações das mídias onde serão exibidos nossos projetos.

unit	name	equivalence
<b>'cm'</b>	centimeters	1cm = 96px/2.54
<b>'mm'</b>	millimeters	1mm = 1/10th of 1cm
<b>'Q'</b>	quarter-millimeters	1Q = 1/40th of 1cm
<b>'in'</b>	inches	1in = 2.54cm = 96px
<b>'pc'</b>	picas	1pc = 1/6th of 1in
<b>'pt'</b>	points	1pt = 1/72th of 1in
<b>'px'</b>	pixels	1px = 1/96th of 1in

```
h1 { margin: 0.5in }      /* inches */
h2 { line-height: 3cm }  /* centimeters */
h3 { word-spacing: 4mm } /* millimeters */
h3 { letter-spacing: 1Q } /* quarter-millimeters */
h4 { font-size: 12pt }   /* points */
h4 { font-size: 1pc }    /* picas */
p { font-size: 12px }    /* px */
```

### Píxels (px)

Essa é uma medida bastante famosa para os web designers, grande parte dos desenvolvedores web usam o píxel como unidade principal de seus projetos. A definição de píxel de referência no CSS é o ângulo visual(0.0213deg) de um píxel em um dispositivo com a densidade de 96dpi a uma distancia de um braço do leitor (28 polegadas). O benefício desse píxel de referência é que ele leva a proximidade da tela em consideração, ao usarmos um celular que seguramos próximos de nós, o píxel de referência terá o tamanho semelhante ao de um

monitor mais distante de nós, por exemplo. Um dos aspectos mais importantes para esses Web Designers é a escalabilidade e adaptabilidade de um layout, ou seja, a medida em que as unidades aumentam de uma maneira previsível e razoável, seu layout deve ser capaz de se adequar a essas mudanças. De um lado, a ideia de se manter o mesmo aspecto (tamanho) em diversos dispositivos pode parecer atrativa (cuidado com o píxel de referência!), mas do outro lado, temos consequências negativas quando estamos lidando com dispositivos de baixa resolução (blurry rendering).

### **Points (pt)**

A próxima unidade é Point. Definitivamente essa unidade é mais conhecida pelos designers, principalmente os que estudam tipografia. Essa medida é geralmente utilizada em propriedades relacionadas a fonte do seu projeto. Sua abreviação se dá com a marcação de pt e seu uso não é tão comum, você provavelmente verá essa unidade muito raramente. Geralmente espera-se que essa medida seja utilizada em folhas de estilo para impressões, quando se precisa ter certeza do tamanho da fonte utilizada. Não é recomendada para a estilização em tela!

### **Polegadas in (inches)**

Polegada ou inch em inglês é mais uma unidade de medida que conhecemos do mundo das medidas absolutas - geralmente vemos elas quando queremos comprar uma nova TV ou monitor , mas essa unidade também existe no mundo Web. Apesar de existirem, elas não costumam ser utilizadas em projetos, uma vez que não existem um uso prático para elas (podemos atingir os mesmos resultados utilizando outras unidades)

### **Centímetro e Milímetro (cm / mm)**

Nós brasileiros, que adotamos o sistema métrico, conhecemos bem essas duas medidas, que são bastante utilizadas no dia a dia. Apesar de bastante comuns, tanto centímetro e milímetro são pouco usadas no CSS. Assim como o

pt, o uso dessas duas é esperado para folhas de estilo para impressões (medidas mais precisas), evitando que elas sejam aplicadas para exibições em tela.

### **Paica (pc)**

Também uma unidade pouco usada no mundo web, a Paica também vem para o CSS sendo herdada da tipografia. Por não ser uma unidade amplamente conhecida, ela acaba sendo fadada ao esquecimento, mas é sempre importante conhecermos todas as ferramentas que estão à nossa disposição. A relação entre as unidades absolutas é:



## UNIDADES DE MEDIDAS - RELATIVAS

Essas medidas são calculadas tendo como base uma outra unidade de medida definida, como por exemplo “em” e o “rem”. O uso delas é mais apropriado para que possamos fazer ajustes em diferentes dispositivos garantindo um layout consistente e fluido em diversas mídias. Devido ao fato de que essas medidas são calculadas pelo browser baseando-se em outra unidade, elas tendem a ser bastante flexíveis. Ou seja, podemos ter resultados diferentes de acordo com o ambiente.

### Ems (em)

Uma técnica bastante utilizada consiste justamente em fazer uso desse poder do em componentizando nossos elementos. A ideia é que a alteração do tamanho da fonte do elemento pai faça com que todo o componente se modifique e redimensione baseando-se nesse novo valor. Quando expressamos tamanhos como margin, padding utilizando “em”, isso significa que eles serão relativos ao tamanho da fonte do elemento pai. Portanto, de acordo com o tamanho da fonte utilizada em determinado elemento, os elementos filhos serão redimensionados de forma a obedecer a referência a esse tamanho de fonte!

```
<style>
  #div{
    font-size: 16px;
  }

  #filho{
    font-size: 2em;
  }
</style>

<div id="pai">
  div pai
  <div id="filho">
    div filho
  </div>
</div>
```

Nesse exemplo o elemento filho terá um tamanho de 32px que é 2em do tamanho do elemento pai, ou seja, 16px \*2.

Essa é uma unidade de medida utilizada para facilitar a manutenção do componente como um todo, sem ter que sofrer alterando valores de todas as partes do componente, como por exemplo pra ícones e elementos que precisam escalar junto com o texto, como botões e inputs.

Porém um ponto que se deve atentar ao usar o “em”, é que quando usamos essa medida, nós temos que considerar o font-size de todos os elementos pai. Isso tende a se complicar quando estamos falando de 5, 6, 7 divs aninhadas, provavelmente não será muito divertido calcular isso. O lado bom é que a unidade de medida rem resolve esse problema.

### **Rems (rem, "root em")**

O REM vem como sucessor do EM e ambos compartilham a mesma lógica de funcionamento (font-size), porém a forma de implementação é diferente. Enquanto o em está diretamente relacionado ao tamanho da fonte do elemento pai, o rem está relacionado com o tamanho da fonte do elemento root (raiz), no caso, a tag. O fato de que o rem se relaciona com o elemento raiz resolve aquele problema que tínhamos com diversas divs (elementos) aninhados, uma vez que não haverá essa "herança" de tamanhos, lembra?! Ou seja, não precisaremos ter dor de cabeça tendo que realizar cálculos, uma vez que nos baseamos na tag raiz(root), ou seja, o arquivo html.

```
html{
  font-size: 62,5%;
}

h1{
  font-size: 1.2rem; /*equivalente a 12px*/
}

p{
  font-size: 2.4rem; /*equivalente a 24px*/
}
```

Normalmente os browsers especificam o tamanho default da fonte do elemento root (raiz) sendo 16px, nesse exemplo foi definido o tamanho da fonte para 62,5% desse valor, ou seja, 10px. Logo após foi definido o tamanho da fonte do elemento h1 para 1.2rem ou seja 120% do valor do elemento raiz, 12px. Outro exemplo também foi definido o elemento p para equivaler a 2.4rem, ou seja, 24px.

## **Porcentagem (%)**

Apesar de não ser uma unidade de medida, a porcentagem costuma ser bastante utilizada quando falamos de layout responsivo e fluido, por isso, não poderia deixá-la passar. A porcentagem permite que criemos módulos que sempre vão se readaptar para ocupar a quantidade especificada. Por exemplo, se definirmos um elemento tendo um tamanho de 50%, independente do dispositivo em questão, esse módulo sempre ocupará metade do espaço que lhe cabe (caso esteja dentro de algum outro elemento).

A porcentagem tem um comportamento parecido com o “em”, já que ele se relaciona diretamente com o tamanho da propriedade do elemento pai. Portanto, ao trabalharmos com a porcentagem, temos o mesmo problema que tínhamos com o em, quanto mais elementos aninhados, mais complicado será de definirmos exatamente o tamanho, por isso, deve-se ter cuidado ao utilizá-la.

## **Ex e Ch**

Diferentemente da forma como a EM e a REM funcionavam, essa unidade não se relaciona com o tamanho da fonte (font-size), mas com qual fonte está sendo utilizada naquele momento (font-family), mais especificamente ao tamanho do caractere x dessa fonte em questão (x-height).

Como o browser sabe esse valor? Esse valor pode vir diretamente com a fonte, o browser pode medir o caractere em caixa baixa (lower case) e se esses dois não funcionarem, o browser estipula um valor de 0.5em para 1ex.

Diferente do “ex” mas também relacionado ao estilo da fonte, o ch (character unit) é definida na documentação como sendo a "medida avançada" da largura do caractere zero ("0").

## **Vw (viewport width)**

Essa medida faz parte das medidas mais atuais e do futuro do CSS. Viewport units. Viewport nada mais é que a área visível de uma página web para

o seu usuário, essa viewport pode variar de acordo com o dispositivo, sendo menor em celulares e maior em desktops. Antigamente, quando não existiam tablets e celulares capazes de acessar sites, todas as web pages eram pensadas para a tela de um computador, com tamanho fixo e design estático. Com a chegada desses dispositivos móveis, essas páginas eram grandes demais para serem exibidas nesses aparelhos, o que tornava muito difícil a navegação. A primeira solução partiu dos browsers desses dispositivos, eles adotavam um comportamento de retirar o zoom de forma que o site inteiro coubesse na tela do aparelho, definitivamente não era o ideal, mas uma solução rápida. No HTML5, foi introduzida uma maneira para que os desenvolvedores conseguissem alterar a viewport através da tag, corrigindo esse problema de usabilidade relacionado aos dispositivos móveis.

A unidade vw essa unidade se relaciona diretamente com a largura da viewport, onde 1vw representa 1% do tamanho da largura dessa área visível. A diferença entre vw e a % é bem semelhante a diferença entre em e rem, onde a % é relativa ao contexto local do elemento e o vw é relativo ao tamanho total da largura da viewport do usuário.

### **Vh (viewport height)**

Essa unidade funciona da mesma forma que o vw, porém dessa vez, a referência será a altura e não a largura

### **Vmin (viewport minimum)**

Essa unidade também se relaciona com as dimensões da viewport, mas com um porém. Anteriormente quando vimos vh e vw precisávamos escolher se gostaríamos de nos basear na altura (vh) ou na largura (vw) da viewport. Diferentemente das anteriores, o vmin utilizará como base a menor dimensão da viewport (altura x largura). Imagine que estamos trabalhando com uma viewport de 1600px de altura e 900px de largura. Nesse caso, 1vmin terá o valor de 9px (1% da menor dimensão!), caso tenhamos 100vmin, esse será igual a 900px. No caso acima, a menor dimensão foi a da largura, porém se tivéssemos 300px para

altura e 1400px para largura, nosso valor de referência seria o 300px! Sempre a menor dimensão!

### **Vmax (viewport maximum)**

Seguindo a mesma base lógica da unidade anterior, o vmax terá como valor de referência a maior dimensão da viewport. Ou seja, utilizando o mesmo exemplo, se tivermos 1600px de altura e 900px de largura, 1vmax será equivalente a 16px. No segundo exemplo ocorrerá a mesma inversão, tendo 300px para altura e 1400px para largura, 1vmax será equivalente a 14px. Dessa vez sempre será a maior dimensão.

## MARGIN E PADDING

### Margins

As propriedades CSS marginsão usadas para criar espaço ao redor dos elementos, fora de quaisquer bordas definidas. Com CSS, você tem controle total sobre as margens. Existem propriedades para definir a margem para cada lado de um elemento (superior, direito, inferior e esquerdo).

- margin-top
- margin-right
- margin-bottom
- margin-left

Todas as propriedades de margem podem ter os seguintes valores:

- auto - o navegador calcula a margem
- length - especifica uma margem em px, pt, cm, etc.
- % - especifica uma margem em % da largura do elemento que o contém
- inherit - especifica que a margem deve ser herdada do elemento pai
- 

### CSS Padding

O padding (preenchimento) é usado para criar espaço ao redor do conteúdo de um elemento, dentro de qualquer borda definida. Assim como margin o padding possui propriedades para definir cada lado do elemento (superior, direito, inferior e esquerdo).

- padding-top
- padding-right
- padding-bottom
- padding-left

Todas as propriedades de preenchimento podem ter os seguintes valores:

- length - especifica um preenchimento em px, pt, cm, etc.
- % - especifica um preenchimento em % da largura do elemento que o contém
- inherit - especifica que o preenchimento deve ser herdado do elemento pai

### **Margin e Padding com propriedades abreviadas**

Tanto o Margin quanto o Padding aceitam o formato chamado ShortHand, onde você pode indicar mais de um valor na mesma linha.

Exemplos:

- Com 4 valores:
  - margin: top right bottom left;
  - margin: 25px 30px 35px 40px;
- Com apenas 3 valores:
  - padding: top right|left bottom
  - padding: 25px 50px 20px ( 25 px de margem para cima, 50 tanto para direita quanto para esquerda e 20 para baixo)
- Com dois valores:
  - margin: top|bottom right|left
  - margin: 25px 50px (25 px de margem para cima e para baixo, 50 tanto para direita quanto para esquerda)
- Com apenas 1 valor:
  - padding: top|bottom|right|left
  - padding: 25px; (25px de margem para todos os lados);

## BORDER

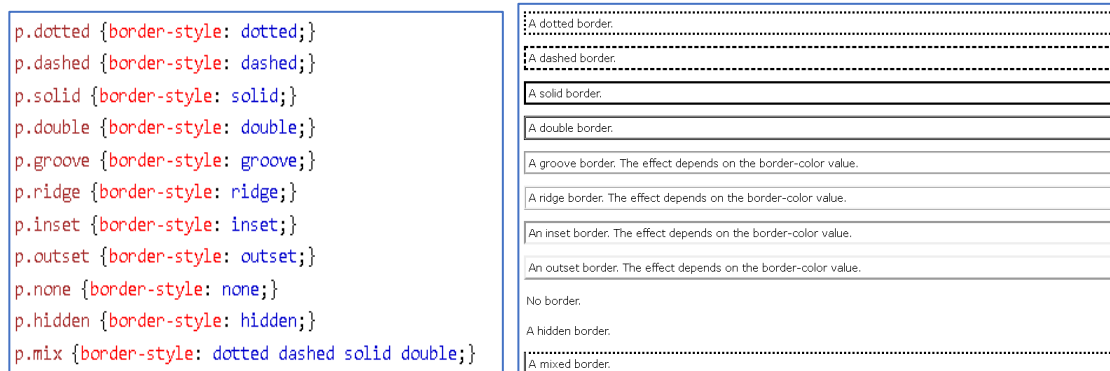
As propriedades da borda CSS permitem que você especifique o estilo, a largura e a cor da borda de um elemento.

### border-style

A propriedade border-style especifica que tipo de borda exibir. Os seguintes valores são permitidos:

- dotted- Define uma borda pontilhada
- dashed- Define uma borda tracejada
- solid- Define uma borda sólida
- double- Define uma borda dupla
- groove- Define uma borda ranhurada 3D. O efeito depende do valor da cor da borda
- ridge- Define uma borda 3D estriada. O efeito depende do valor da cor da borda
- inset- Define uma borda de inserção 3D. O efeito depende do valor da cor da borda
- outset- Define uma borda inicial 3D. O efeito depende do valor da cor da borda
- none- Não define borda
- hidden- Define uma borda oculta

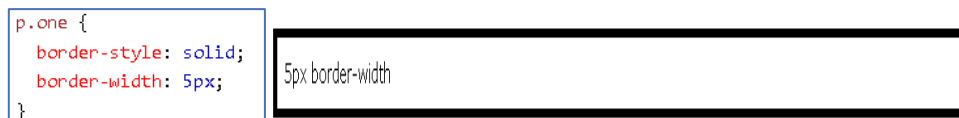
A propriedade border-style pode ter de um a quatro valores (para a borda superior, borda direita, borda inferior e borda esquerda).





## border-width

A propriedade `border-width` especifica a largura das quatro bordas. A largura pode ser definida como um tamanho específico (em px, pt, cm, em, etc) ou usando um dos três valores predefinidos: `thin`, `medium`, or `thick` (fino, médio ou grosso).



## border-color

A propriedade `border-color` é usada para definir a cor das quatro bordas. A cor pode ser definida por:

name - especifique um nome de cor, como "red"

HEX - especifique um valor HEX, como "#ff0000"

RGB - especifique um valor RGB, como "rgb(255,0,0)"

HSL - especifique um valor HSL, como "hsl(0, 100%, 50%)"

transparente

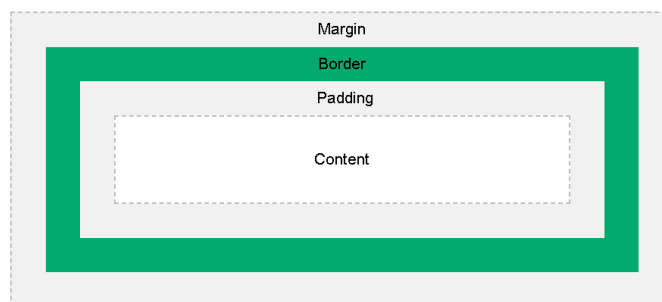
Nota: Se `border-color` não estiver definido, herda a cor do elemento.

## border-radius

A propriedade `border-radius` é usada para adicionar bordas arredondadas a um elemento:

## BOX MODEL

Todos os elementos HTML podem ser considerados como caixas. Em CSS, o termo “box model” (modelo de caixa) é usado quando se fala de design e layout. O modelo de caixa CSS é essencialmente uma caixa que envolve cada elemento HTML. Ele consiste em: margens, bordas, preenchimento e o conteúdo real. A imagem abaixo ilustra o modelo da caixa:



Explicação das diferentes partes:

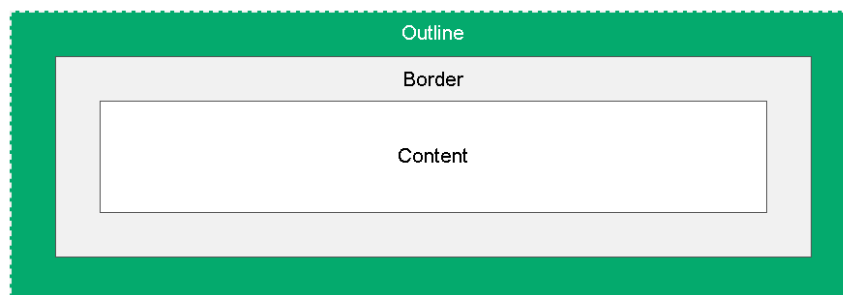
- Content - O conteúdo da caixa, onde o texto e as imagens aparecem
- Padding - Limpa uma área ao redor do conteúdo. O forro é transparente
- Border - Uma borda que contorna o preenchimento e o conteúdo
- Margin - Limpa uma área fora da fronteira. A margem é transparente

O modelo de caixa nos permite adicionar uma borda ao redor dos elementos e definir o espaço entre os elementos.

Para definir a largura e a altura de um elemento corretamente em todos os navegadores, você precisa saber como funciona o modelo de caixa. Ao definir as propriedades de largura e altura de um elemento com CSS, basta definir a largura e a altura da área de conteúdo. Para calcular o tamanho total de um elemento, você também deve adicionar preenchimento, bordas e margens. Ex: 320px (largura) + 20px (preenchimento esquerdo + direito) + 10px (borda esquerda + direita) + 0px (margem esquerda + direita) = largura total (width) 350px.

## OUTLINE

A propriedade outline é um contorno, uma linha desenhada fora da borda do elemento. O contorno difere das bordas. Ao contrário da borda, o contorno é desenhado fora da borda do elemento e pode sobrepor outro conteúdo. Além disso, o contorno NÃO faz parte das dimensões do elemento; a largura e a altura totais do elemento não são afetadas pela largura do contorno.



### outline-style

A propriedade outline-style especifica o estilo do contorno e pode ter um dos seguintes valores:

- dotted- Define um contorno pontilhado
- dashed- Define um contorno tracejado
- solid- Define um contorno sólido
- double- Define um contorno duplo
- groove- Define um contorno ranhurado 3D
- ridge- Define um contorno 3D estriado
- inset- Define um contorno de inserção 3D
- outset- Define um esboço inicial 3D
- none- Não define contorno
- hidden- Define um contorno oculto

Nenhuma das outras propriedades de contorno terá qualquer efeito a menos que a propriedade outline-style seja definida.

### outline-width

A propriedade outline-width especifica a largura do contorno e pode ter um dos seguintes valores:

- thin (normalmente 1px)
- medium (normalmente 3px)
- thick (normalmente 5px)
- Um tamanho específico (em px, pt, cm, em, etc)

### **outline-color**

A propriedade outline-color é usada para definir a cor do contorno. A cor pode ser definida por:

- name - especifique um nome de cor, como "vermelho"
- HEX - especifique um valor hexadecimal, como "#ff0000"
- RGB - especifique um valor RGB, como "rgb(255,0,0)"
- HSL - especifique um valor HSL, como "hsl(0, 100%, 50%)"
- invert - executa uma inversão de cores (o que garante que o contorno fique visível, independentemente da cor do fundo)

### **outline-offset**

A propriedade outline-offset adiciona espaço entre um contorno e a borda/borda de um elemento. O espaço entre um elemento e seu contorno é transparente. O exemplo a seguir especifica um contorno de 15px fora da borda da borda:

Este parágrafo tem um contorno de 15px fora da borda da borda.

## AJUSTES DE TEXTO

CSS tem muitas propriedades para formatar texto.

### color

A propriedade color é usada para definir a cor do texto. A cor é especificada por:

- um nome de cor - como "red"
- um valor HEX - como "#ff0000"
- um valor RGB - como "rgb(255,0,0)"

```
h1 {  
  color: green;  
}
```

This is heading 1

### background-color

A propriedade background-color define o fundo do texto. O alto contraste é muito importante para pessoas com problemas de visão. Portanto, certifique-se sempre de que o contraste entre a cor do texto e a cor de fundo (ou imagem de fundo) seja bom.

```
h1 {  
  background-color: black;  
  color: white;  
}
```

This is a Heading

### Alinhamento e direção de texto

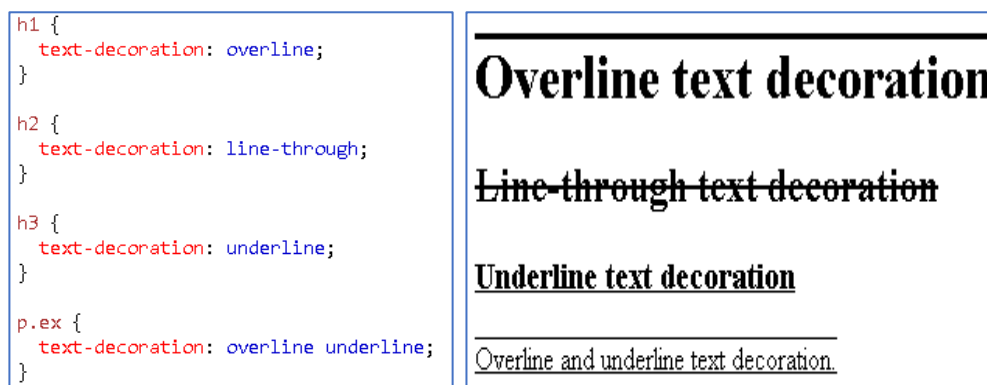
As propriedades que alteram o direcionamento e o alinhamento de texto são:

- **text-align** - Especifica o alinhamento horizontal do texto (center, left, right, justify)
- **text-align-last** - Especifica como alinhar apenas a última linha de um texto
- **vertical-align** - Define o alinhamento vertical de um elemento

- **direction** - Especifica a direção do texto/direção de escrita (da esquerda para direita ou direita para esquerda)
- **unicode-bidi** - Usado junto com a propriedade direction para definir ou retornar se o texto deve ser substituído para suportar vários idiomas no mesmo documento.

### text-decoration-line

A propriedade text-decoration-line é usada para adicionar uma linha de decoração ao texto.



As propriedades de decoração de texto são:

- **text-decoration** - Define todas as propriedades de decoração de texto em uma declaração
- **text-decoration-color** - Especifica a cor da decoração de texto
- **text-decoration-line** - Especifica o tipo de decoração de texto a ser usado (sublinhado, overline, etc.)
- **text-decoration-style** - Especifica o estilo da decoração do texto (solid, dotted, wave, etc.)
- **text-decoration-thickness** - Especifica a espessura da linha de decoração de texto

### text-transform

A propriedade text-transform é usada para especificar letras maiúsculas e minúsculas em um texto. Ele pode ser usado para transformar tudo em letras

maiúsculas ou minúsculas, ou colocar em maiúscula a primeira letra de cada palavra: através dos valores uppercase, lowercase e capitalize.

## Espaçamento de texto

Existem propriedades em CSS que tornam possível alterar o recuo de texto, espaçamento entre letras, altura da linha, espaçamento entre palavras e espaço em branco. Essas propriedades são:

- **text-indent** - Especifica o recuo da primeira linha em um bloco de texto
- **letter-spacing** - Especifica o espaço entre os caracteres em um texto
- **line-height** - Especifica a altura da linha
- **word-spacing** - Especifica o espaço entre as palavras em um texto
- **white-space** - Especifica como lidar com o espaço em branco dentro de um elemento

## text-shadow

A text-shadow propriedade adiciona sombra ao texto. Ela aceita uma lista de sombras separadas por vírgula que serão aplicados ao texto e ao text-decoration do elemento. Cada sombra é especificada como um deslocamento do texto, juntamente com valores opcionais de cor e raio de desfoque. Múltiplas sombras são aplicadas de frente-para-trás, com a primeira sombra especificada no topo.

```
h1 {
  text-shadow: 2px 2px 5px red;
}
```

**Text-shadow effect!**

```
h1 {
  color: white;
  text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px darkblue;
}
```

**Text-shadow effect!**

## FONTES

Escolher a fonte certa tem um enorme impacto em como os leitores experimentam um site. A fonte certa pode criar uma identidade forte para sua marca. Usar uma fonte que seja fácil de ler é importante. A fonte agrega valor ao seu texto. Também é importante escolher a cor e o tamanho do texto corretos para a fonte.

### **font-family**

Especifica a família da fonte para o texto. A propriedade `font-family` deve conter vários nomes de fontes como um sistema "fallback", para garantir a máxima compatibilidade entre navegadores/sistemas operacionais. Comece com a fonte desejada e termine com uma família genérica (para permitir que o navegador escolha uma fonte semelhante na família genérica, se não houver outras fontes disponíveis). Os nomes das fontes devem ser separados por vírgula. Se o nome da fonte for mais de uma palavra, deve estar entre aspas, como: "Times New Roman". Em CSS existem cinco famílias de fontes genéricas:

- **Serif** - As fontes com serifa têm um pequeno traço nas bordas de cada letra.
- **Sans-serif** - As fontes sem serifa têm linhas limpas (sem pequenos traços anexados).
- **Monospace** - Fontes monoespaçadas todas as letras têm a mesma largura fixa. Eles criam uma aparência mecânica.
- **Cursive** - As fontes cursivas imitam a caligrafia humana.
- **Fantasy** - Fontes de fantasia são fontes decorativas/lúdicas.

### **font-size**

Especifica o tamanho da fonte do texto. Ser capaz de gerenciar o tamanho do texto é importante no design da web. No entanto, você não deve usar ajustes de tamanho de fonte para fazer os parágrafos parecerem títulos ou os títulos parecerem parágrafos. O valor do tamanho da fonte pode ser um tamanho absoluto ou relativo.



- **Tamanho absoluto:** Define o texto para um tamanho especificado. Não permite que um usuário altere o tamanho do texto em todos os navegadores (ruim por motivos de acessibilidade). O tamanho absoluto é útil quando o tamanho físico da saída é conhecido.
- **Tamanho relativo:** Define o tamanho em relação aos elementos circundantes. Permite que um usuário altere o tamanho do texto nos navegadores.

```
h1 {
  font-size: 40px; /* 40px/16=2.5em */
}

h2 {
  font-size: 1.875em; /* 30px/16=1.875em */
}

p {
  font-size: 0.875em; /* 14px/16=0.875em */
}
```

# This is heading 1

## This is heading 2

This is a paragraph.

### font-style

Especifica o estilo da fonte para o texto. É usada principalmente para especificar texto em itálico. Esta propriedade tem três valores:

- **normal** - O texto é mostrado normalmente
- **italic** - O texto é mostrado em itálico
- **oblique** - O texto está "inclinado" (o oblíquo é muito semelhante ao itálico, mas menos suportado)

```
p.normal {
  font-style: normal;
}

p.italic {
  font-style: italic;
}

p.oblique {
  font-style: oblique;
}
```

This is a paragraph in normal style.

*This is a paragraph in italic style.*

*This is a paragraph in oblique style.*

### font-variant

Especifica se um texto deve ou não ser exibido em um fontes small-caps. Em uma fonte de letras minúsculas, todas as letras minúsculas são convertidas

em letras maiúsculas. No entanto, as letras maiúsculas convertidas aparecem em um tamanho de fonte menor do que as letras maiúsculas originais no texto.

```
p.normal {
  font-variant: normal;
}

p.small {
  font-variant: small-caps;
}
```

My name is Hege Refsnes.

MY NAME IS HEGE REFSNES.

## font-weight

Especifica o peso de uma fonte

```
p.normal {
  font-weight: normal;
}

p.light {
  font-weight: lighter;
}

p.thick {
  font-weight: bold;
}

p.thicker {
  font-weight: 900;
}
```

This is a paragraph.

This is a paragraph.

**This is a paragraph.**

**This is a paragraph.**

## font

Uma propriedade abreviada para definir todas as propriedades da fonte em uma declaração. Os valores font-size e font-family são obrigatórios. Se um dos outros valores estiver ausente, seu valor padrão será usado.

```
p.a {
  font: 20px Arial, sans-serif;
}

p.b {
  font: italic bold 12px/30px Georgia, serif;
}
```

Este é um parágrafo. O tamanho da fonte é definido como 20 pixels e a família de fontes é Arial.

*Este é um parágrafo. A fonte é definida como itálico e negrito, o tamanho da fonte é definido como 12 pixels, a altura da linha é definida como 30 pixels e a família de fontes é Georgia*

## FONTES CUSTOMIZADAS

Se você não quiser usar nenhuma das fontes padrão em HTML, poderá usar o Google Fonts, ou baixar um arquivo de fonte separado (formato TTF/OTF recomendado).

### Fontes do Google

As fontes do Google são gratuitas e têm mais de 1.000 fontes para escolher. Para usar as fontes do Google basta adicionar um link de folha de estilo especial na seção <head> e, em seguida, consultar a fonte no CSS.

```
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">
<style>
body {
  font-family: "Sofia", sans-serif;
}
</style>
</head>
```

**Sofia Font**

Lorem ipsum dolor sit amet.

123456790

### @face-font

As fontes da Web permitem que os designers da Web usem fontes que não estão instaladas no computador do usuário. Quando você encontrar/comprar a fonte que deseja usar, basta incluir o arquivo de fonte em seu servidor web, e ele será baixado automaticamente para o usuário quando necessário. Suas fontes "próprias" são definidas dentro da regra CSS @font-face. Para usar a fonte em um elemento HTML, consulte o nome da fonte por meio da propriedade font-family.

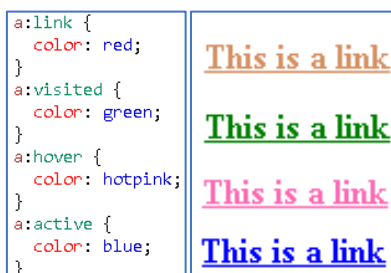
```
@font-face {
  font-family: myFirstFont;
  src: url(sansation_light.woff);
}

div {
  font-family: myFirstFont;
}
```

## LINK

Com CSS, os links podem ser estilizados de muitas maneiras diferentes. Além disso, os links podem ter estilos diferentes dependendo do estado em que estão. Os quatro estados de links são:

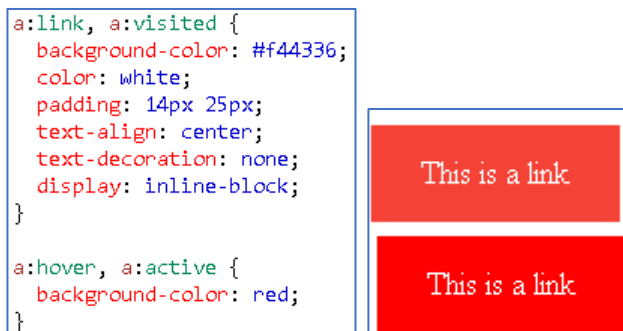
- **a:link**- um link normal e não visitado
- **a:visited**- um link que o usuário visitou
- **a:hover**- um link quando o usuário passa o mouse sobre ele
- **a:active**- um link no momento em que é clicado



Ao definir o estilo para vários estados de link, existem algumas regras de ordem:

- a:hover DEVE vir depois de a:link e a:visited
- a:active DEVE vir depois de a:hover

A propriedade text-decoration é usada principalmente para remover sublinhados de links. A propriedade background-color pode ser usada para especificar uma cor de fundo para links.



Nesse exemplo um link foi estilizado para se parecer com um botão, foram definidos dois estilos para o botão, sendo um para os estados link e visited, e outro estilo para os estados hover e active.

## LISTAS

Em HTML, existem dois tipos principais de listas:

- listas não ordenadas (<ul>) - os itens da lista são marcados com marcadores
- listas ordenadas (<ol>) - os itens da lista são marcados com números ou letras

As propriedades da lista CSS permitem:

- Definir marcadores de itens de lista diferentes para listas ordenadas
- Definir marcadores de itens de lista diferentes para listas não ordenadas
- Definir uma imagem como marcador de item da lista
- Adicionar cores de fundo a listas e itens de lista

### list-style-type

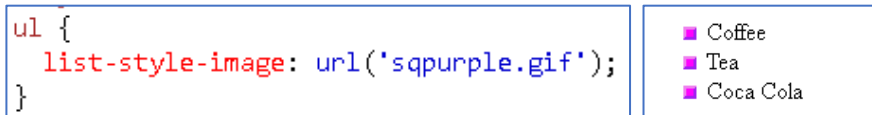
A propriedade list-style-type especifica o tipo de marcador de item de lista.

O exemplo a seguir mostra alguns dos marcadores de itens de lista disponíveis:

<pre>ul.a {   list-style-type: circle; }  ul.b {   list-style-type: square; }  ol.c {   list-style-type: upper-roman; }  ol.d {   list-style-type: lower-alpha; }</pre>	<p>Example of unordered lists:</p> <ul style="list-style-type: none"> <li>o Coffee</li> <li>o Tea</li> <li>o Coca Cola</li> </ul> <ul style="list-style-type: none"> <li>▪ Coffee</li> <li>▪ Tea</li> <li>▪ Coca Cola</li> </ul> <p>Example of ordered lists:</p> <ol style="list-style-type: none"> <li>I Coffee</li> <li>II Tea</li> <li>III Coca Cola</li> </ol> <ol style="list-style-type: none"> <li>a. Coffee</li> <li>b. Tea</li> <li>c. Coca Cola</li> </ol>
<pre>ul.demo {   list-style-type: none;   margin: 0;   padding: 0; }</pre>	<p>Coffee Tea Coca Cola</p>

## list-style-image

A propriedade list-style-image especifica uma imagem como o marcador de item da lista:



## list-style-position

A propriedade list-style-position especifica a posição dos marcadores de item de lista (pontos de bala).

"list-style-position: outside;" significa que os marcadores estarão fora do item da lista. O início de cada linha de um item de lista será alinhado verticalmente. Este é o padrão:

- |   |
|---|
| <ul style="list-style-type: none"> <li>• Café - Uma bebida preparada a partir de grãos de café torrados...</li> </ul> |
| <ul style="list-style-type: none"> <li>• Chá</li> </ul>   |
| <ul style="list-style-type: none"> <li>• Coca Cola</li> </ul>   |

"list-style-position: inside;" significa que os marcadores estarão dentro do item da lista. Como faz parte do item da lista, fará parte do texto e empurrará o texto no início:

- |   |
|---|
| <ul style="list-style-type: none"> <li>• Café - Uma bebida preparada a partir de grãos de café torrados...</li> </ul> |
| <ul style="list-style-type: none"> <li>• Chá</li> </ul>   |
| <ul style="list-style-type: none"> <li>• Coca Cola</li> </ul>   |

## list-style


A propriedade `list-style` é uma propriedade abreviada. Ele é usado para definir todas as propriedades da lista em uma declaração. Ao usar a propriedade abreviada, a ordem dos valores da propriedade é:

1. `list-style-type`(se uma imagem de estilo de lista for especificada, o valor desta propriedade será exibido se a imagem por algum motivo não puder ser exibida)
2. `list-style-position`(especifica se os marcadores de item de lista devem aparecer dentro ou fora do fluxo de conteúdo)
3. `list-style-image`(especifica uma imagem como marcador de item da lista)

Se um dos valores de propriedade acima estiver ausente, o valor padrão para a propriedade ausente será inserido, se houver.

```
ul {  
  background: #3399ff;  
  padding: 20px;  
}
```

```
ul li {  
  background: #cce5ff;  
  color: darkblue;  
  margin: 5px;  
}
```

- 
- Coffee
  - Tea
  - Coca Cola

**REFERENCIAS:**

[https://www.hostinger.com.br/tutoriais/diferenca-entre-estilos-css?ppc\\_campaign=google\\_performance\\_max&qclid=CjwKCAjw6dmSBhBkEiwA\\_W-EoHOeEhDWeSvYEHb7XyCi0W4nZvEHfjwvt6iwEwrDEs6saraWC0eqUxoCgLQQA\\_vD\\_BwE](https://www.hostinger.com.br/tutoriais/diferenca-entre-estilos-css?ppc_campaign=google_performance_max&qclid=CjwKCAjw6dmSBhBkEiwA_W-EoHOeEhDWeSvYEHb7XyCi0W4nZvEHfjwvt6iwEwrDEs6saraWC0eqUxoCgLQQA_vD_BwE)

<https://www.maujor.com/tutorial/insetut.php#:~:text=Uma%20folha%20de%20estilo%20%C3%A9%20dita%20inline%2C%20quando%20as%20regras,a%20apresenta%C3%A7%C3%A3o%20com%20a%20marca%C3%A7%C3%A3o.>

<https://www.w3schools.com/cssref/default.asp> (referencias para alterações em css)

<https://www.w3.org/TR/css-values-3/> (unidade de medidas)

<https://www.alura.com.br/artigos/guia-de-unidades-no-css> (UNIDADE DE MEDIDA)