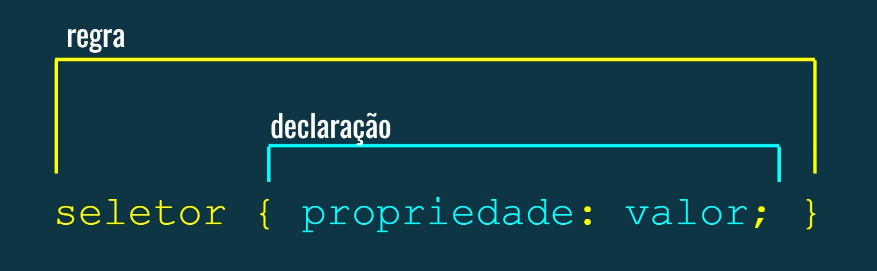
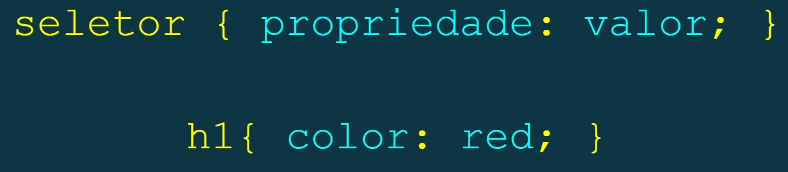
**CSS**

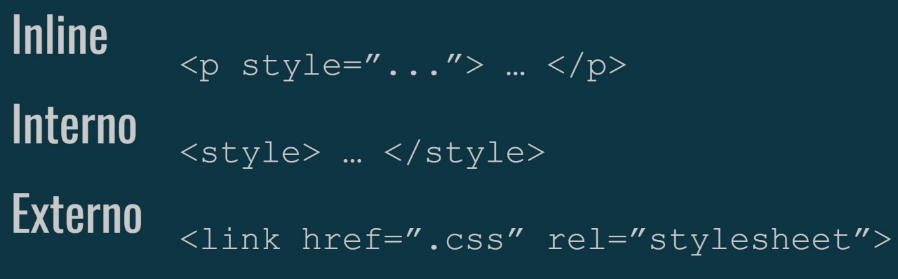
1. **Sintaxe**

Uma regra no CSS é composta por seletor, propriedade e valor.

**Seletor**: qual elemento do HTML você quer formatar.



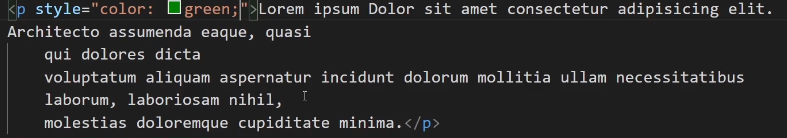




É possível usar Inline, Interno e Externo ao mesmo tempo.

1. **Inline**

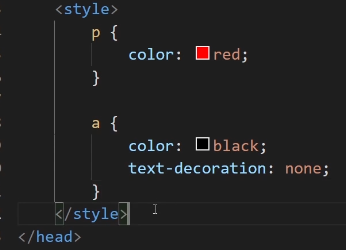
Colocar o atributo dentro da própria tag. Tomar cuidado com a manutenção desses atributos. Atributos inline tem prioridade sobre atributos internos.



1. **Interno**

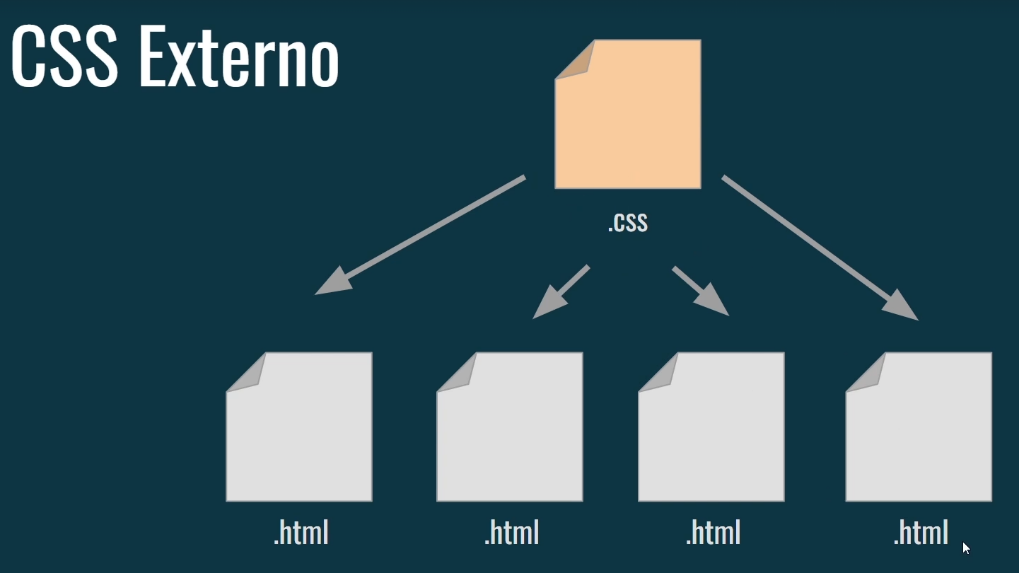
Para uma única página, é recomendável usar o modo interno. Basta colocar a tag <style> dentro do head e definir os seletores do CSS.

A tag style lida com o CSS da página



1. **Externo**

Para muitas páginas, usar o externo, que é basicamente criar um arquivo próprio para o CSS e nele definir todos os seletores dos arquivos HTML.

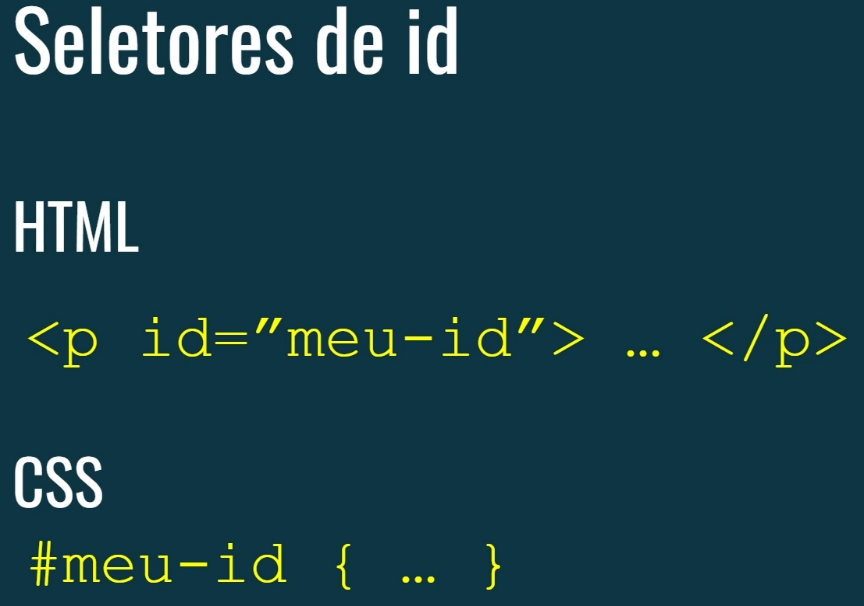


Uma alteração no CSS é refletida em todos os arquivos HTML. Para vincular a página HTML com o arquivo CSS, **é preciso utilizar a tag <link>** no head:

<link rel="stylesheet" href="introducao.css">

**“rel = stylesheet”** é obrigatório e href é o nome do arquivo CSS.

Através dos atributos “class” e “id”, é possível atribuir nomes às tags do HTML com o intuito de facilitar a alteração de suas propriedades no CSS.



* **Qual é a diferença entre class e id?**

O id é único por página. Não é possível ter mais de um id com o mesmo nome em uma página. Já a classe é como se fosse um nome, possível de ser repetido.

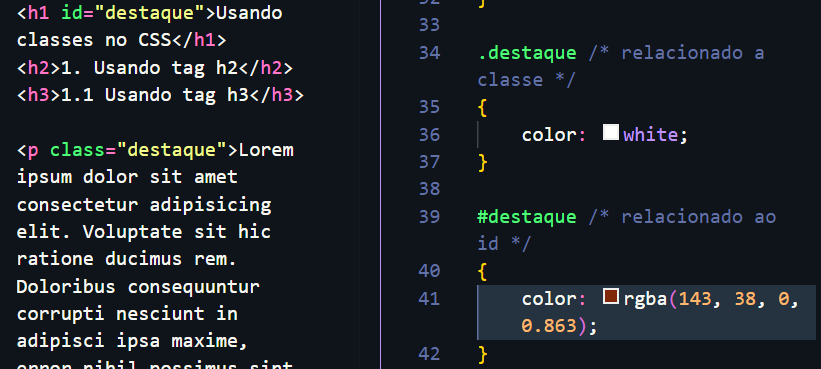
**ID:** é acessado por “#”.

**Atalho pra criar div:** se você digitar div#div1, por exemplo, o VS Code irá gerar automaticamente: <div id=”div1”></div>

**Class:** é acessada por “.”

É possível atribuir duas classes a um mesmo objeto. Basta colocar, dentro das aspas, o nome de uma delas e em seguida o nome da outra.

Destaque e size sendo usadas ao mesmo tempo no parágrafo.



**- Multiplicando parágrafos automaticamente – Exemplo:**

**Comando**: p{\*PARÁGRAFO}\*n + TAB

Sendo “n” o número de vezes que você quer que o parágrafo se repita.

**Exemplo:**

Tela preta com letras brancas

Descrição gerada automaticamente

**Resultado:**

**Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente**

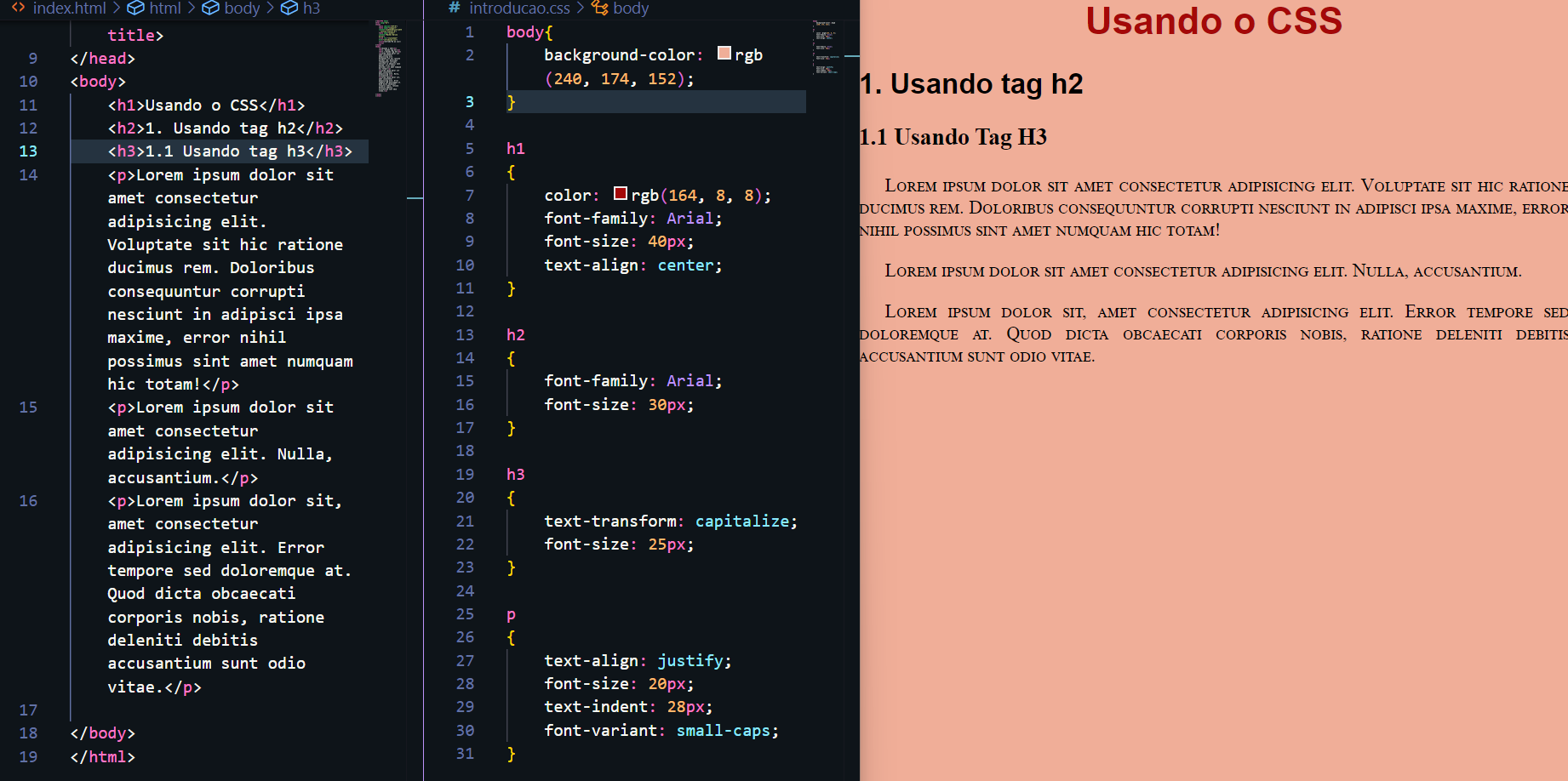
1. **Propriedades**

* **Fontes**
* **Font-family:** define o nome da fonte (Arial, Calibri etc). Como o funcionamento dessa propriedade depende que o usuário final tenha a fonte escolhida baixada, é possível passar mais de um valor para essa propriedade, pois, caso o usuário não tenha a primeira fonte, a segunda opção pode ser usada.

Exemplo:



* **Font-size:** pode ser medida em pixel (px).
* **Font-style:** italic, oblique
* **Font-weight:** bold, normal
* **Font-variant**
* **Text**
* **Text-align:** justify, right, left, center, descentralizar etc
* **Text-decoration:** underline, line-through (colocar risco no texto), overline, none (remove sublinhados dos links)
* **Text-indent:** útil para parágrafos. Colocar o valor em pixels que você deseja espaçar do ínicio da página até a primeira palavra do texto.
* **Text-transform:** uppercase, lowercase, capitalize (a primeira letra de cada palavra em maiúsculo)



* **Display**
* **Float**
* Interface gráfica do usuário

  Descrição gerada automaticamente com confiança médiaValores: right, left ou none (padrão)
* É usada para posicionar elementos na tela.
* Quando um elemento é definido como float, ele é retirado do seu fluxo normal e colocado ao longo do lado direito ou esquerdo do seu container, permitindo que os outros elementos possam organizar em torno dele.
* A propriedade float pode ser usada em elementos em bloco ou em linha.
* Não é mais tão usada no dia a dia para estruturar layouts
* **Elementos abaixo da div2 tentam se encaixar no conteúdo dele:Interface gráfica do usuário

  Descrição gerada automaticamente**
* **#div3 não faz mais parte de “container”:**

Interface gráfica do usuário

Descrição gerada automaticamente

Outros exemplos:

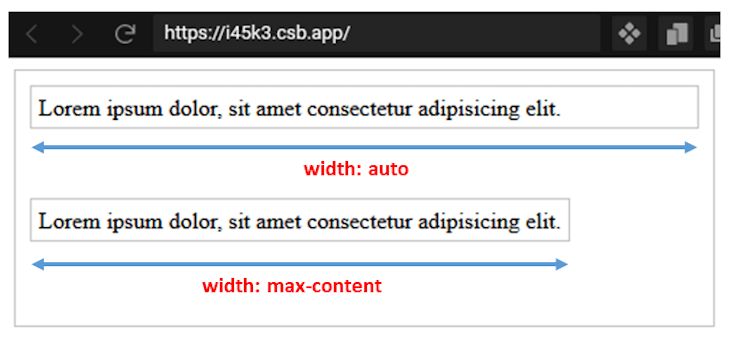
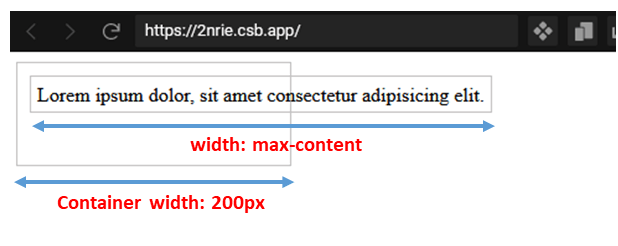
Texto

Descrição gerada automaticamenteTexto

Descrição gerada automaticamente

* Texto

  Descrição gerada automaticamente com confiança baixa**Clear**
* **Valores**: right, left ou both
* Limpa elementos que possuem a propriedade float nos valores right ou left
* **Fit, min e max-content**
* **min-content**: é o menor tamanho que uma caixa pode ter sem acontecer overflow em seu conteúdo. A palavra mais longa dentro do conteúdo define o tamanho da caixa;
* **max-content**:representa o tamanho que uma caixa precisa para conter todo o seu conteúdo sem “quebrar” (wrap) ou “estourar” (overflow).

* **fit-content**: a caixa usa o espaço disponível, mas nunca se expande além de max-content e nunca diminui além de min-content.

Texto

Descrição gerada automaticamente

* **Overflow**
* Texto

  Descrição gerada automaticamenteÉ uma propriedade do CSS utilizada para tratar conteúdos que ultrapassam os limites de um elemento.
* **overflow: hiden**: todos os conteúdos que ultrapassam os limites de um elemento são escondidos na tela.

Texto

Descrição gerada automaticamente

* **overflow: scroll**: similar o overflow-hidden, porém permite que o resto do conteúdo seja exibido através do uso de uma barra de rolagem. Coloca barras de rolagem vertical e horizontal, mesmo que uma delas não seja usada

Texto

Descrição gerada automaticamente

* Interface gráfica do usuário, Texto

  Descrição gerada automaticamente**overflow: auto:** coloca barra de rolagem apenas no eixo em que é necessário haver uma (horizontal ou vertical).
* **overflow: visible:** padrão da propriedade overflow.
* **Position**
* Serve para posicionar elementos na tela, como o nome indica.
* **position: static**: é o valor padrão. A ordem dos elementos no browser é a mesma ordem dos elementos no HTML.
* **position:relative**: trabalha em conjunto com outras propriedades como left e top. No exemplo abaixo, a div3 se afasta 15px da esquerda e 15px da direita, deslocando-se.

Interface gráfica do usuário

Descrição gerada automaticamente

* **position:fixed:** sempre se relaciona com o viewport, ou seja, com o browser. O bloco que tem essa propriedade sai do fluxo normal do container e começa a se relacionar com o browser.

Interface gráfica do usuário

Descrição gerada automaticamente

Com positions, a ordem das divs na tela não é necessariamente a mesma da ordem do HTML. Uma div que está no início da tela pode estar sendo declarada no final do HTML, mas, com a propriedade positions, ela é elevada ao topo (com fixed, por exemplo).

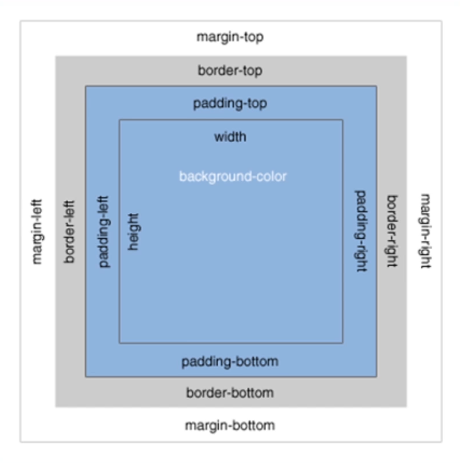
Elementos podem se sobrepor. Para tratar isso, existe a propriedade z-index:

* **z-index**
* Tela de computador com fundo azul

  Descrição gerada automaticamenteControla “quem está na frente de quem” no ponto de vista do usuário.
* Valor padrão: 0
* Quanto maior o número do index, mais à frente o elemento está.
* Aceita números negativos

1. **Comentário**

/\* Essa é a estrutura de um comentário no CSS \*/

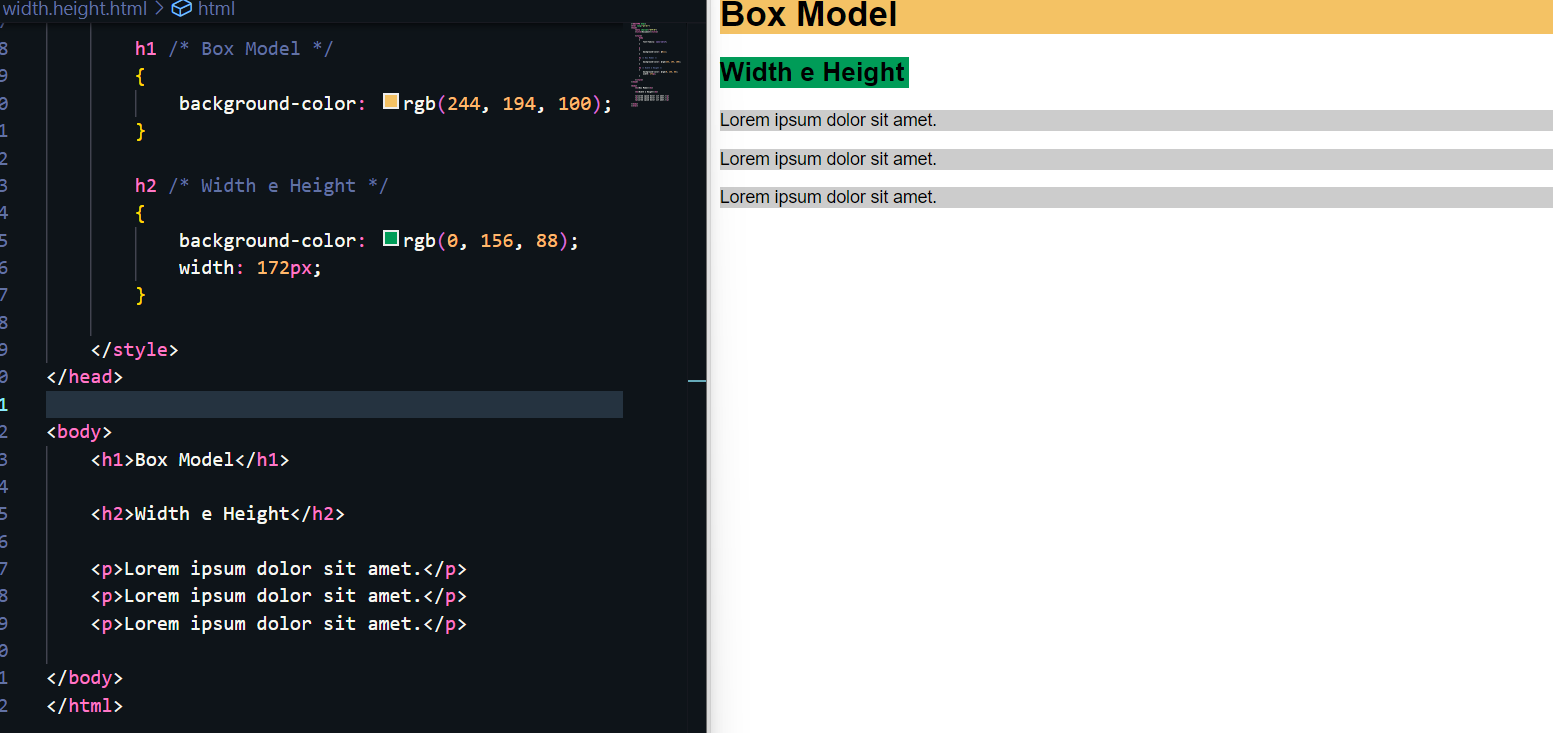
1. **Box Model**

Nunca use propriedades do Box Model em elementos de linha.

* **Width**

Determina a largura da área de conteúdo de um elemento. A área de conteúdo fica dentro do preenchimento, da borda, e da margem de um elemento. Medida em pixels ou porcentagem.

O valor de width determinado é proporcional ao tamanho do body.

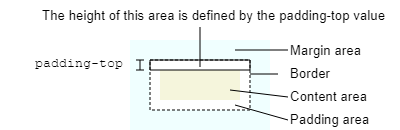


* **Height**

Determina a altura de um objeto, como uma imagem, vídeo, iframe etc. Definida em pixels por padrão. Também é proporcional ao body.

* **Padding**

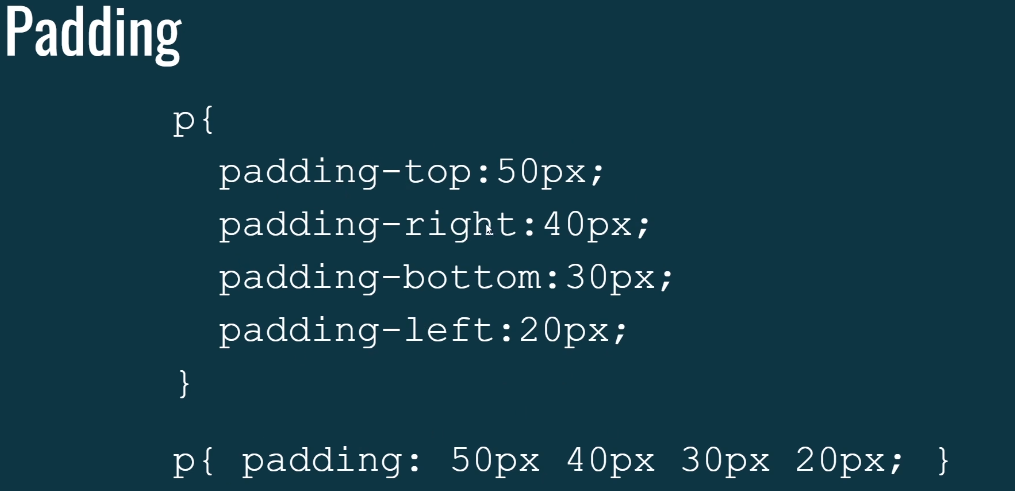
Determina a distância entre o texto e suas bordas, tanto a superior e a inferior quanto as laterais. Medida em pixels.



O Padding pode ser segregado em:

* Padding-top;
* Padding-right;
* Padding-bottom;
* Padding-left

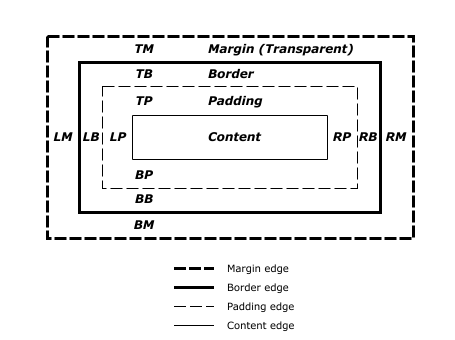
É importante lembrar dessa ordem (associar ao sentido de um relógio, sentido horário).



* **Margin**

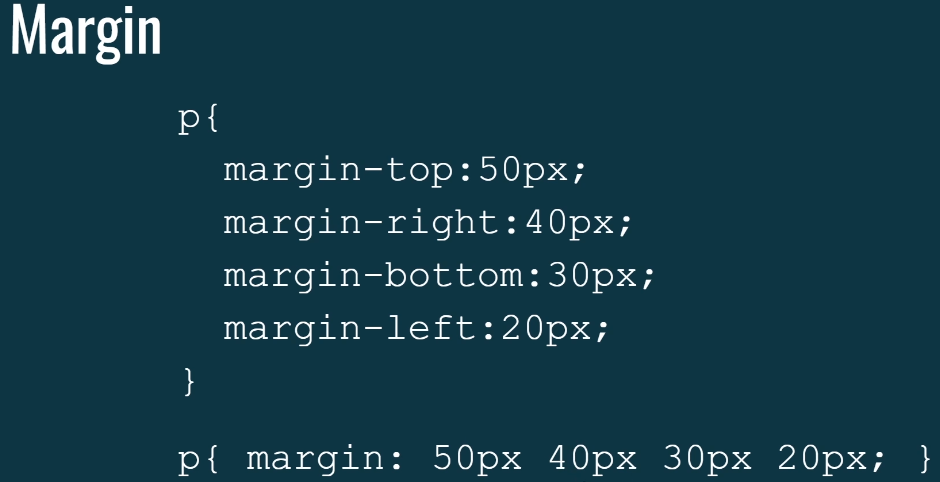
Similar ao padding, porém determina a distância entre as bordas e o resto do body. Medida em pixels.

Margin-auto: centraliza o conteúdo.



Margin pode ser segregado em:

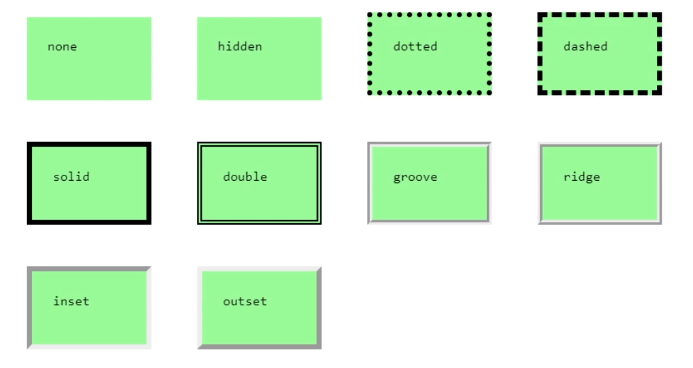
* Margin-top;
* Margin-right;
* Margin-bottom;
* Margin-left

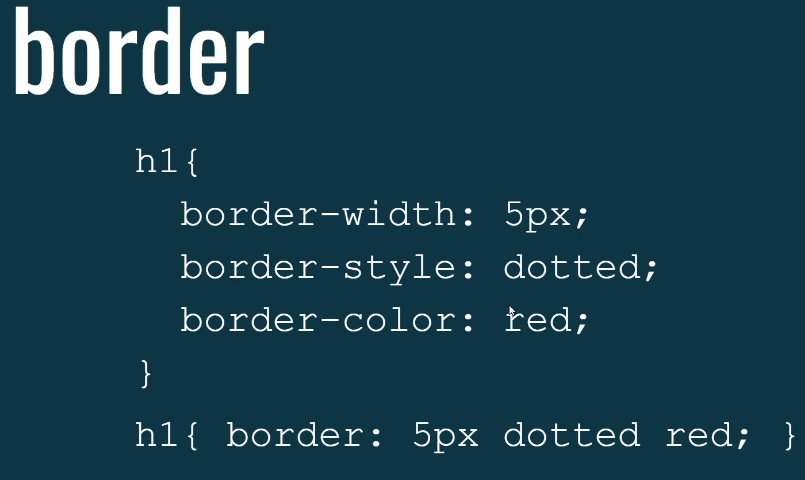


* **Border**

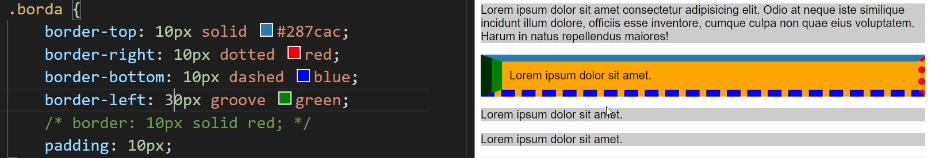
Atributos: espessura (width), estilo (style) e cor (color).

* Estilos:

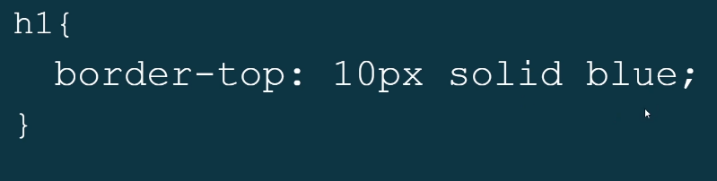




* É possível que cada lado da borda tenha um valor diferente para seus atributos.

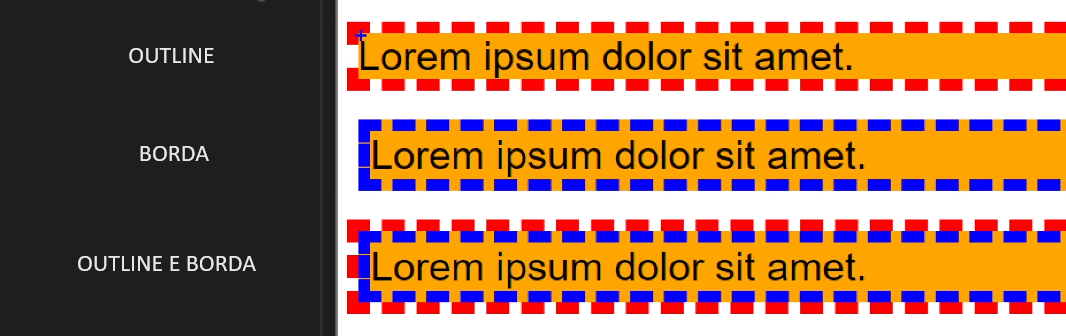


* Em uma linha é possível definir a formatação da borda. **Ordem**: width, style, color.



* **Outline**

A diferença entre outline e borda é que, ao contrário da borda, o outline não afeta o tamanho do elemento.



* **Margin auto**

É utilizada para centralizar um elemento na página horizontalmente. Ao definir "margin: auto;" para os lados direito e esquerdo de um elemento, o navegador automaticamente calcula a margem necessária para centralizá-lo na página.

1. **list-style**

Para formatar uma lista no CSS, pegar o tipo dela (ul/ol) e em seguida o elemento “li” para começar a formatação.

**list-style-type:** define a figura que irá “ordenar” a lista.

**5.1. Listas não ordenadas**

* **list-style-type:** Por padrão, é um círculo preto (disc), mas pode ser do tipo square ou circle ou none, por exemplo.
* **list –style-position:** joga a lista mais para a direita (inside) ou esquerda (outside) da tela.

ul li {

list-style-type: square;

list-style-position: inside

}

**5.2. Listas ordenadas**

* **List-style-type:** por padrão, é o decimal, mas pode serupper/loweralpha ou latin (ordena a lista por letras maiúsculas ou minúsculas), upper/lower-roman (ordena por números romanos maiúsculos ou minúsculos)

ol li {

list-style-type: upper-alpha inside;

}

1. **Pseudo-classes em links**

Em qualquer pseudo-classe, não é permitido haver espaço antes ou depois dos dois pontos (“:”)

**a:visited** : formata links que já foram visitados.

**a:hover** : formata o background do texto clicável.

1. **Texto com sombra**

A propriedade “**text-shadow**” permite o sombreamento de textos pelo CSS. Essa propriedade aceita 4 parâmetros:

1. **Deslocamento horizontal** (**obrigatório**): Especifica o quanto a sombra é deslocada para a direita (valor positivo) ou para a esquerda (valor negativo). Exemplo: 2px.
2. **Deslocamento vertical** (**obrigatório**): Especifica o quanto a sombra é deslocada para baixo (valor positivo) ou para cima (valor negativo). Exemplo: 2px.
3. **Desfoque (opcional):** Determina o raio de desfoque da sombra. Quanto maior o valor, mais desfocada será a sombra. Se omitido, o valor padrão é 0, o que significa uma sombra nítida. Exemplo: 5px.
4. **Cor (opcional):** Define a cor da sombra. Pode ser especificada em qualquer formato de cor válido em CSS (nome da cor, hexadecimal, rgb(), rgba(), hsl(), etc.). Se omitido, o valor padrão é a cor do texto. Exemplo: rgba(0, 0, 0, 0.5).

Texto

Descrição gerada automaticamenteTexto, Logotipo

Descrição gerada automaticamente