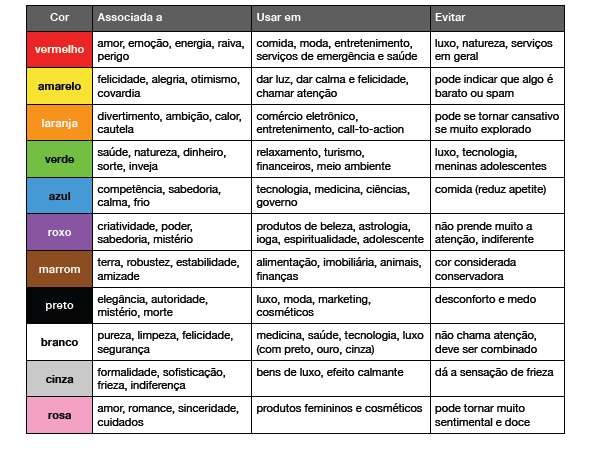


Capitulo 13 a Capitulo 17 dos PDFs!

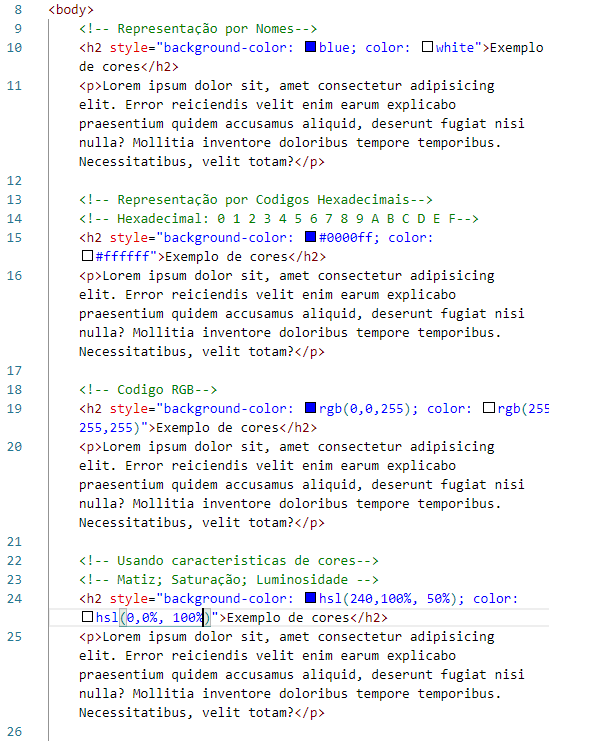
**O poder das cores**

\* Não está proibido usar em sites outras cores!

É recomendado para muita leitura usar o padrão: Letra preta e Fundo branco.

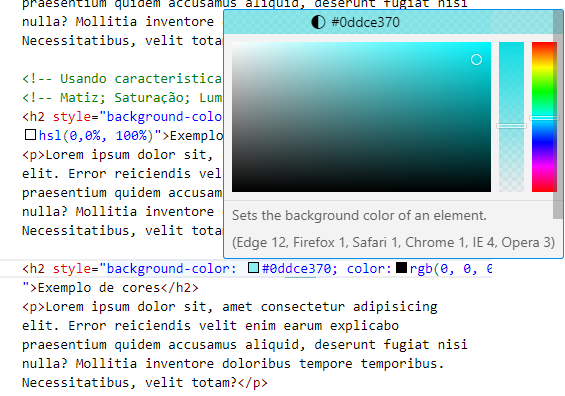
Todas as cores é formada por conjunto de vermelho, verde e azul! (*se eu coloco: #0000ff – significa que coloco 00 de vermelho, 00 de verde e ff de azul*)

Então as formas que podemos representar as cores é: Por nomes; Por Hexadecimais; Por RGB e Por HSL



Para mudar no VSCODE posso só passar por cima do nome e escolher a cor que quero!

Se eu triscar em cima, ele muda a maneira que eu quero (exemplo rgb, hexadecimal e etc)



**Harmonia de Cores**

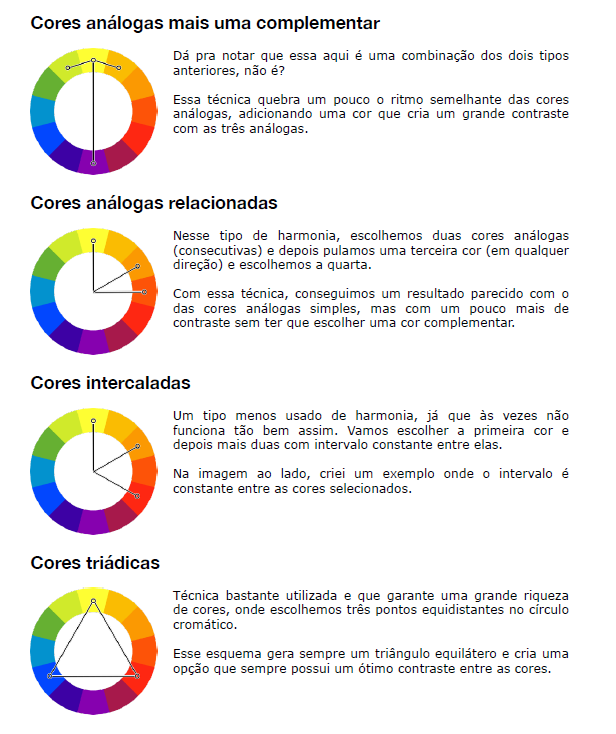
Circulo Cromático:



\***Geralmente 3 a 5 cores no máximo!**

\*Quando tiver logo, observar a cor primaria da logo para utilizar no site!

* Cores complementares: Cores opostas. (melhor contraste)
* Cores análogas: Cores vizinhas (a da esquerda e direita) – Harmonia.
* Monocromia: Modificar a saturação e brilho da cor!



[Roda de cores, um gerador de paleta de cores | Adobe Color](https://color.adobe.com/pt/create/color-wheel) – Auxilia escolher cores

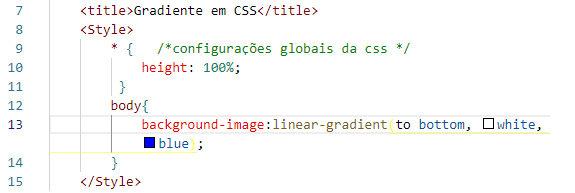
[Paletton - The Color Scheme Designer](https://paletton.com/#uid=1000u0kllllaFw0g0qFqFg0w0aF) -mostra exemplo de site

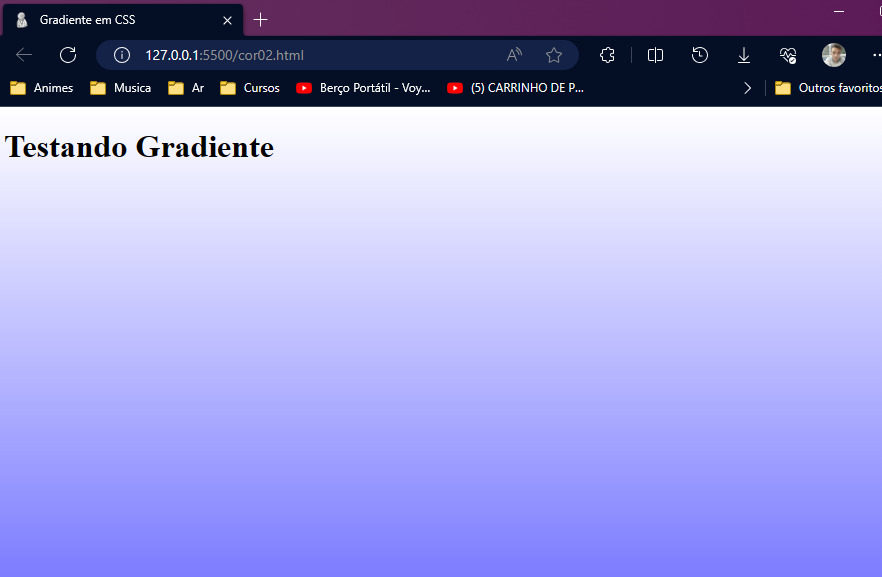
[Create a Palette - Coolors](https://coolors.co/afd2e9-9d96b8-c1292e-2e3532-22031f) – ajuda fixando cor

**Usando gradientes em CSS**

Para gradiente você deve usar a função Background-image: linear-gradiant (to right, cor1, cor2); \* Right= direita; \*left= esquerda; \*top =cima; \*bottom= baixo

Configurações globais da CSS \*{





Para deixar razoavelmente pode usar o site da adobe.

E em vez de usar linear- gradiente pode utilizar o radial-gradient



E depois da cor eu posso colocar o % dela por exemplo blue 1%, e ele deixa o azul bem fraco!

Exemplo prático:

* **No Html**:

1. <main> divisão especial, indica que o conteúdo é principal.



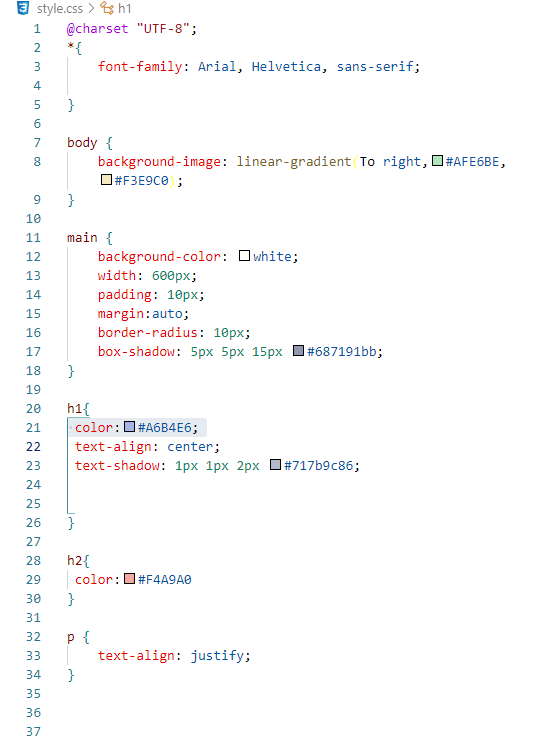
* **No CSS:**

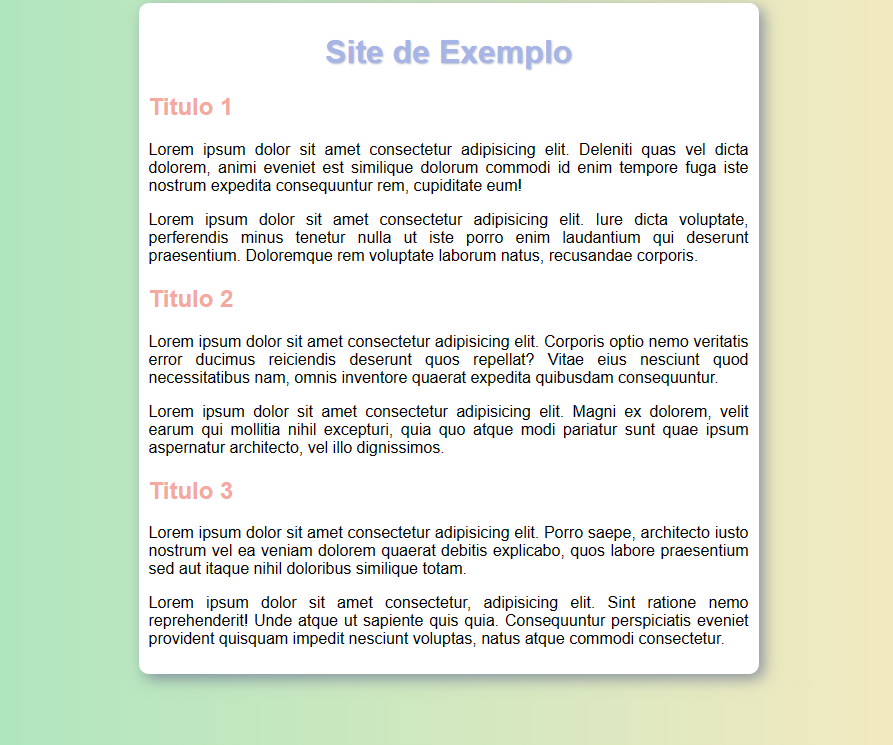
1. width: largura da caixa
2. padding: espaço lateral do texto dentro da caixa
   1. superior é de 10px
   2. direito é de 5px
   3. inferior é de 15px
   4. esquerdo é de 20px
3. margin: define a margens de um elemento ex: margem do texto (auto \* vai centralizar sempre, 3px, ...)
   1. superior é de 10px
   2. direita é de 5px
   3. inferior é de 15px
   4. esquerda é de 20px
   5. auto: centralizar na horizontal
4. text-align: alinhamento do texto (center, end, justify, ...)
5. border-radius: cantos da caixa arredondados
   1. superior esquerdo é de 15px
   2. superior direito é de 50px
   3. inferior direito é de 30px
   4. inferior esquerdo é de 5px
6. box-shadow/text-shadow: sombra da caixa (anexa uma ou mais sombras a um elemento)
7. box-shadow: none|h-offset v-offset blur spread color |inset|initial|inherit;
   1. nome- Valor padrão. Nenhuma sombra é exibida
   2. h-offset-Deslocamento horizontal da sombra
   3. v-offset- Deslocamento vertical da sombra
   4. blur- O raio de desfoque
   5. spread- raio de propagação
   6. color- A cor da sombra
   7. inset- Muda sombra externa para sombra interior
   8. initial- Define para o valor padrão
   9. inherit- Herda esta propriedade do seu elemento pai.

parâmetros: tam. sombra p/ lado, tam. sombra p/ baixo, espalhamento, cor;

Ex: box-shadow: 5px 5px 15px rgba(138, 111, 95, 0.616);

text-shadow: 1px 1px 2px rgba(129, 78, 5, 0.733);





**Fontes**

Tipografia: Arte da escrita!



A - **Altura das maiúsculas**: Altura que as letras maiúsculas vão ocupar. Geralmente um pouco menor que a soma da ascendente com a mediana (< B + D)

B - **Ascendente**: Parte das letras maiúsculas que se ergue acima da linha mediana

C - **Descendente**: Parte das letras minúsculas que passa por baixo da linha de base.

D - **Altura-X**: Também chamada de mediana, define o tamanho das letras minúsculas. Tem esse nome, pois se baseia no tamanho da letra x minúscula.

E - **Corpo**: É a soma de quatro medidas: ascendente + altura-x + descendente + espaço de reserva. É o tamanho total da letra. É o valor que escolhemos ao configurar o tamanho da fonte em um texto.

1 - **Arco**: presente em letras minúsculas. Uma linha curva que nasce em na haste principal.

2 - **Barriga**: curva em uma letra maiúscula ou minúscula, fechada, ligada à haste vertical em dois pontos.

3 - **Braço**: traço horizontal ou inclinado, ligado à haste vertical principal de uma letra maiúscula ou minúscula.

4 - **Cauda**: apêndice do corpo de algumas letras (g, j, J, K, Q, R), que fica abaixo da linha base.

5 - **Enlace**: a forma como uma haste, linha ou filete se liga a um arremate, a uma serifa ou a um terminal. Pode ser angular ou curvilíneo.

6 - **Espinha**: curva e contracurva estrutural da letra S.

7 - **Esporão**: uma projeção que encontramos nas letras b e G.

8 - **Filete**: haste horizontal ou inclinada, fechada nas duas extremidades, por duas hastes ou por uma curva.

9 - **Haste**: traço principal de uma letra, geralmente vertical.

10 - **Olho**: espaço em branco, fechado, dentro de uma letra.

11 - **Orelha**: apêndice presente na letra g, que pode ser em gota, botão, bandeira ou gancho.

12 - **Pé**: terminal ou serifa horizontal que arremata uma perna na parte de baixo.

13 - **Perna**: haste vertical ou inclinada com uma extremidade livre (ou com um pé) e outra extremidade ligada ao corpo da letra.

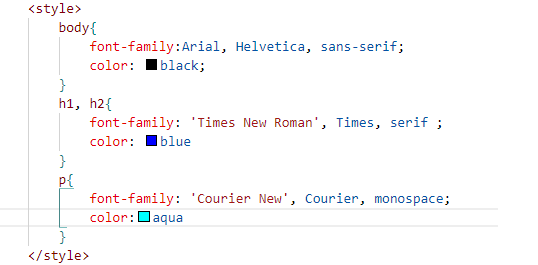
14 - **Serifa**: também chamada de apoio ou patilha. Pequenas retas que ornamentam as hastes de alguns tipos.

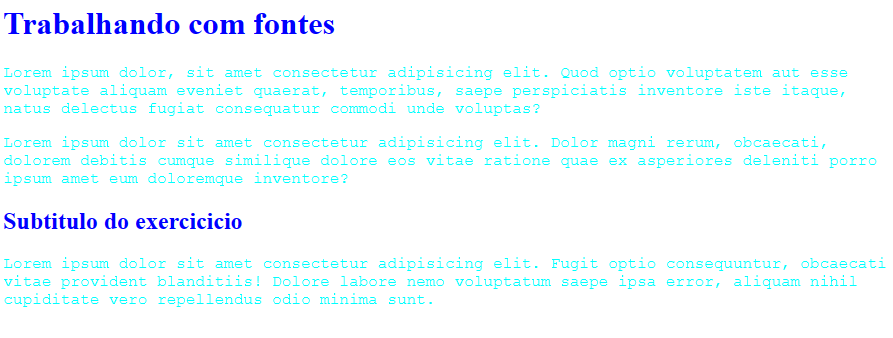
15 - **Terminal**: forma que arremata a extremidade de uma linha curva de uma letra.

16 - **Vértice**: também chamada de ápice. Formada pela convergência de duas hastes que se encontram. Pode ser pontiagudo, oblíquo, plano ou redondo.



* Para **mexer com fonte** utilize o comando: font-family: geralmente o vscode coloca 3 fonte pois caso não tenha ele vai para outra!

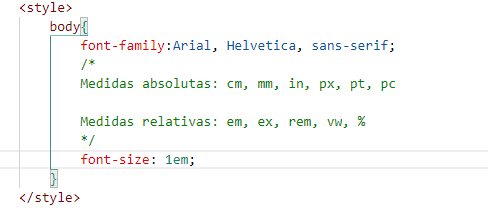




Tamanho: o W3C recomenda o uso do **px** ou do **em**.

* Para **mexer no tamanho** utiliza o font-size, geralmente 16px ou 1em é o tamanho normal da fonte!

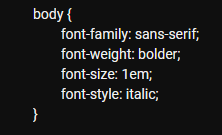
Podemos configurar o tamanho de um título, por exemplo, como sendo 2 vezes maior que a fonte padrão usando o valor 2em para a propriedade.



* **Peso**: para mexer no peso é font- weight: (lighter= mais magro/normal/ bold/bolder) e pode variar também em 100 a 900!

**Estilo**: para mexer no estilo é font-style: italic ou normal.

**Decoração (sublinhado)**: para mexer na decoração é text-decoration: underline;

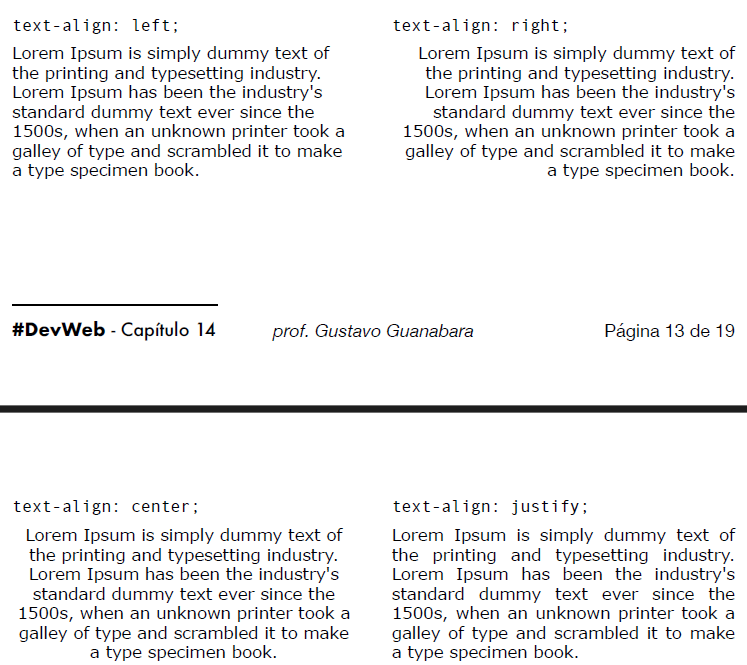


Tem como juntar tudo em um só código:

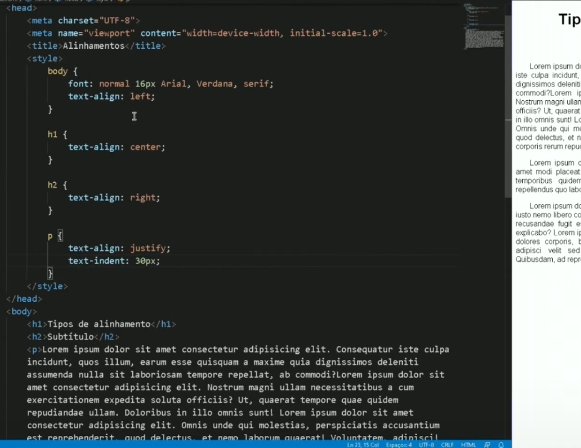
Font: font-style> font-weight> font-size> font-family

*Ex*: font: italic bolder 2em italic (Não precisa ter todas as propriedades, **mas a sequência deve ser seguida**)

* Para **mexer no alinhamento** utiliza o text -aling: (left-esquerda/ right- direita/ center-centralizado/ justify-justificado).



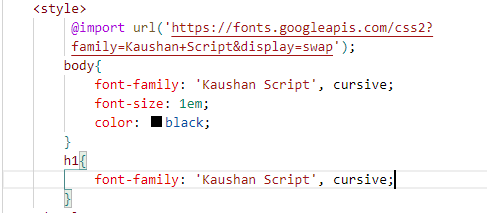
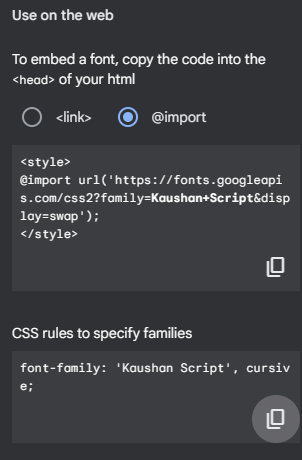
* Para **mexer apenas a primeira** palavra utiliza text-ident: 30px



**Fontes externas**: [Browse Fonts - Google Fonts](https://fonts.google.com/)

Seleciono a família que quero e coloco @import e coloco na primeira linha do style e copia o css rules

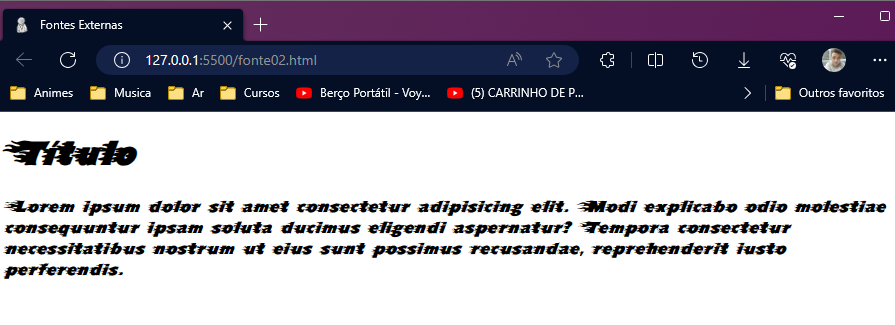
É recomendado **até 4 fontes!**



**Fontes baixadas**: [Blazed | dafont.com](https://www.dafont.com/pt/blazed.font) //otf e ttf geralmente baixa os 2

Baixar a fonte que você quer e coloca @font-face (como regra)





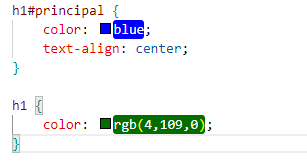
* Para pegar fonte de algum site utiliza o Fonts ninja (extensão)
* Para pegar fonte de uma imagem: [Font Finder 🔎 by What Font Is](https://www.whatfontis.com/), [Free Fonts! Legit Free & Quality » Font Squirrel](https://www.fontsquirrel.com/), [MyFonts | Buy and Download Fonts](https://www.myfonts.com/)

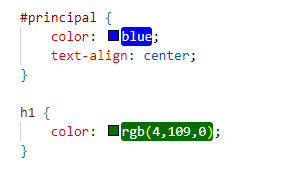
**Seletores personalizados**

Quando dois elementos da mesma classe (ex: h1 e h1) que deseja formatar diferente!

Então é só colocar uma identificação nele. Ex: <h1 id=”principal”> e no css colocar h1#principal { }







\* **Não é preciso colocar h1#principal; pode colocar apenas #principal**

Nota: Uma coisa legal que dá para fazer quando você cria um id, é poder criar uma âncora na página que ao clicar te leva direto ao id específico. Exemplo: Colocar no final da página um <a href="#principal">Clique para voltar ao início</a>

Também dá pra usar isso para fazer um índice no começo da página. Quando tem muito conteúdo às vezes as pessoas não têm paciência de ler até encontrar o que querem, colocar essas âncoras no começo para direcionar pra cada tópico é bacana.

**Segurar alt muda mais de um elemento!**

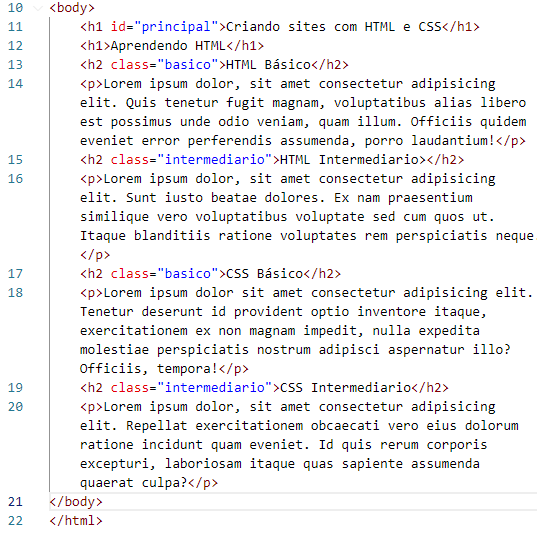
*ID é diferente de class*

ID: identifica um elemento único dentro da página atual. (exemplo: não posso usar o principal que usei no 1 elemento, logo no outro elemento! Isso ta errado)

Class: classe onde um ou mais elemento pertençam.

**Em html é id= em css é #**

**Em html é class= em css é .**





\* **Não é preciso colocar h2.basico; pode colocar apenas .basico**

Se eu quiser criar uma class=”destaque” por exemplo e destacar uma palavra no texto, só colocar a tag <span class=”destaque”>

* **Consegue aplicar mais de uma classe em um elemento! É só separar por espaço** (ex: <h2 class= “intermediário destaque”>
* **Consegue colocar um id e uma classe! Mas id sobrepõe.** (ex: "<h1 id="principal" class="destaque">Criando Sites com HTML e CSS</h1>")

No HTML, temos a tag "<div> </div>", ela não tem uma função específica, serve apenas para demarcar um espaço qualquer dentro da nossa página. Como padrão, se tiver mais de uma, as div's ficam dispostas uma em cada linha, uma embaixo da outra, sendo que elas ocupam uma área que pega a largura inteira do navegador, de ponta à ponta. Exemplo:

<body>

<div>01</div>

<div>02</div>

<div>03</div>

</body>

Mudar a altura e largura das div's

"Height" que é altura

"Width" que é largura

Mudar a posição dos textos nas div's com a Propriedade text-align,

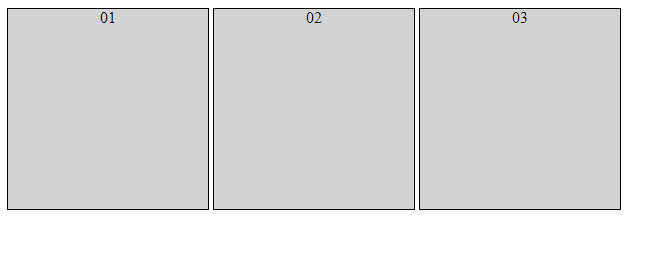
Podemos adicionar bordas às div's com a Propriedade "border" com o valor "solid"

**Nota**: Ao utilizar a Propriedade de Estilo "border", devemos primeiro inserir o valor da espessura da borda em px, depois o tipo de borda (ex: solid) e posteriormente a sua cor. *Exemplo de shorthand para bordas em CSS:*

*" border: 1px solid black; "*

Por padrão as div's ficam alinhadas uma em cima da outra, mas se quisermos que fiquem **uma do lado da outra**, devemos usar "display" com o valor "inline-block", que seria um alinhamento de bloco em linha, assim fica tudo na mesma linha.

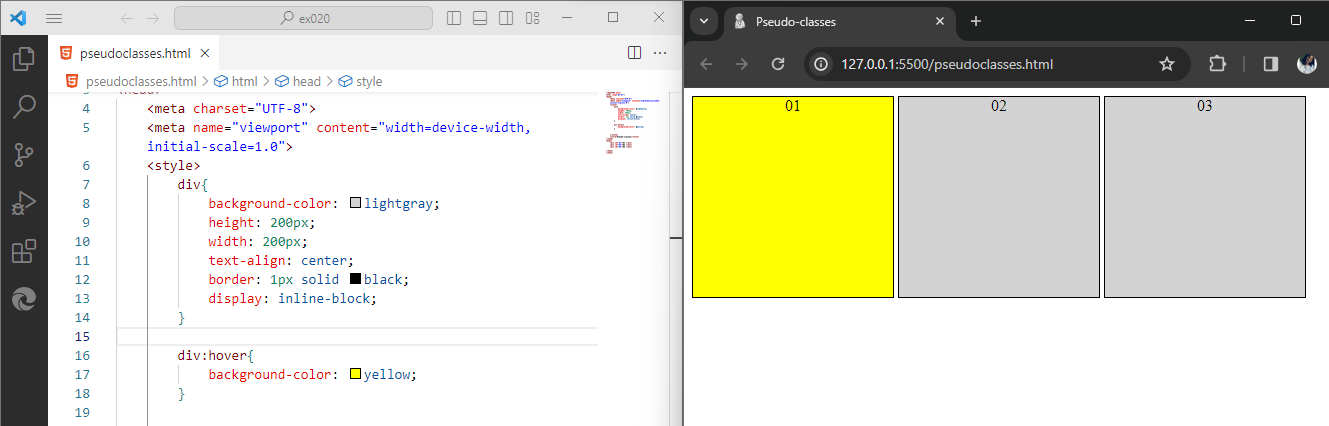




**Pseudo-classes**

São representadas por : no qual tem que esta relacionado a um elemento ou uma classe.

1. :active (seleciona link ativo- clicar no link)
2. :checked (seleciona cada elemento <input> marcado)
3. :enabled
4. :disabled (seleciona cada elemento <input> desabilitado)
5. :corner-present
6. :empty (seleciona cada elemento <p> que não tem filhos)
7. :end
8. :first
9. :hover (quando passa o mouse por cima acontece algo)
   1. ex: div:hover{ background-color:yellow;}
10. :double-button
11. :visited (sites visitados)



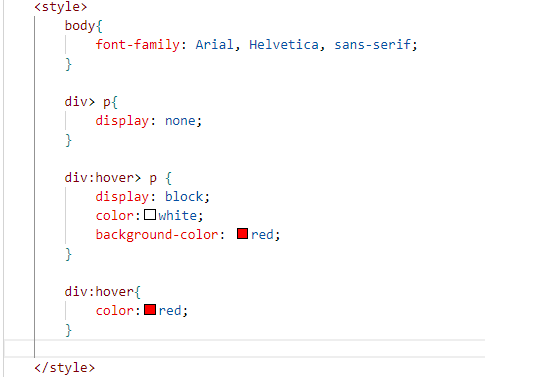
Quando a gente coloca div>p – significa o parágrafo que está dentro da div (p é filho de div)

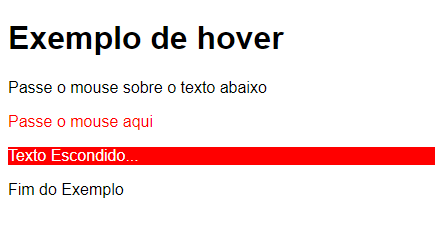
**Para deixar o texto de “p” invisível faz:**

div>P {display:none;}

**Para deixar o texto de “p” visível quando passa o mouse (útil para criar menu)**

div:hover>p{display:block;}





*Por padrão do html quando não visita um site fica azul e quando visita fica roxo*

**Para mudar a cor padrão** faça a{color: red; text-decoration:none font-weight: bold}

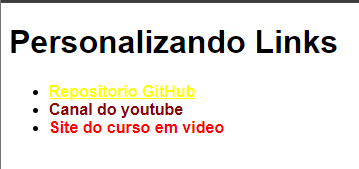
Test-decoration: none- significa para tirar o sobrilhado

**Para mudar a cor visitada** faça a: visited{ color: darkred;}

Se eu quiser que quando passar o mouse em cima eu deixe sublinhado faz a: hover{text-decoratin:underline;}

**Para mudar a cor de um link clicado** faça a:active{ color:yellow;}

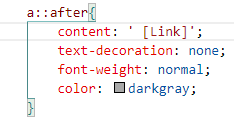




**Pseudo-Elemento**

Eles podem agir nas classes, nos elementos, nos id’s e etc... Eles mexem diretamente no **conteúdo periférico do elemento.** Representado por ::

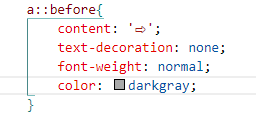
Suponha que queremos que todo link no final tenha uma indicação de que é um link, podemos abrir um seletor personalizado para links e usar o pseudo-elemento (::) " after " para fazer isso.



Content: o valor que aparecera depois de todo o link.



Se quisermos colocar algo antes é só colocar pseudoelemento (::) “before”



Se quisermos colocar que um link é prioridade por exemplo, cria uma classe (class=”...”)



# = ID

. = Class

: = pseudo-class

:: = pseudo-element

> = clildren

**Modelos de Caixas**

Pode ser criado uma hierarquia de caixa (aninhamento).

Ex: <h1> ola, mundo!</h1>

Todos os elementos estão dentro de uma caixa, quando formatamos o elemento, na verdade estamos configurando a caixa que vai estar exibindo o elemento.

Toda caixa possui um tamanho, que é chamado de " **PARTE DE CONTEÚDO** ", esse tamanho é especificado em duas medidas muito importantes, que são:

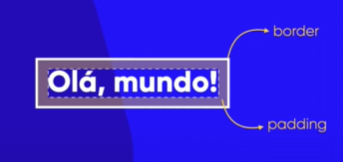
* Altura (" height ")
* Largura (" width")



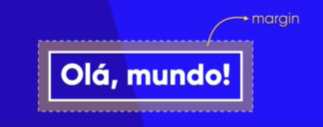
* Borda (“border”) linha em volta da parte de conteúdo.



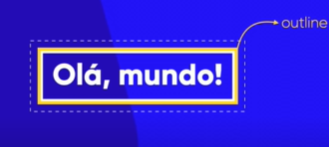
* Se eu quiser afastar a bordar da linha ponteada é com o (“Padding”) (acolchoamento)



* Da borda para dentro tem um espaço, se eu tiver outra caixa em baixo desse “ola, mundo” ele pode ficar grudado nessa linha. Entao eu posso criar (“Margin”). Para externo.



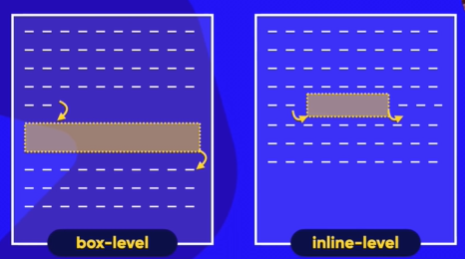
* Dentro da margin eu posso criar outro traço e chamo essa área de (“Outline”).



Sendo assim, ainda referente à anatomia das caixas, todo elemento e consequentemente toda caixa, tem a "Parte de Conteúdo", tem uma "Borda", dessa borda para dentro, temos o "Padding", da borda para fora temos o "Margin" e, ainda dentro do margin e fora da borda, temos a área do "Outline".

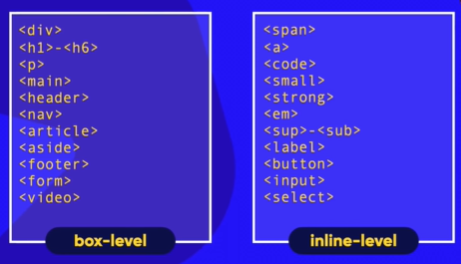
Tipos de caixa:

Existem dois tipos (“box-level/block-level” e “inline-level”.)



**box-level / block-level:** Em uma caixa do tipo "box-level / block-level", o dito elemento sempre irá se iniciar em uma nova linha e como padrão sempre ocupará a largura total da tela ou do elemento onde ele está contido (100% do viewport), sendo que irá pular para a próxima linha antes de continuar o conteúdo posterior e aí sim inserir outros boxes. As caixas desse tipo são isoladas e independentes. Ex: <div> <p> <main> <video> <h1>

**inline-level:** Já uma caixa do tipo "inline-level" não vai começar em uma nova linha, e sim no ponto exato onde foram definidos, no meio de um parágrafo por exemplo. E a largura dele vai ocupar apenas o tamanho relativo ao seu conteúdo, sendo que não irá quebrar a linha ao terminar, continuando o conteúdo de forma direta (em um parágrafo por ex) sem pular para a linha de baixo. Ex: <a> <span> <code> <strong> <em>

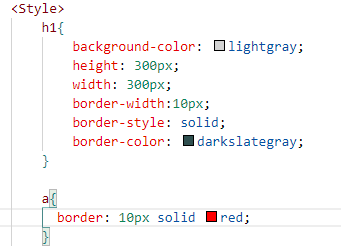


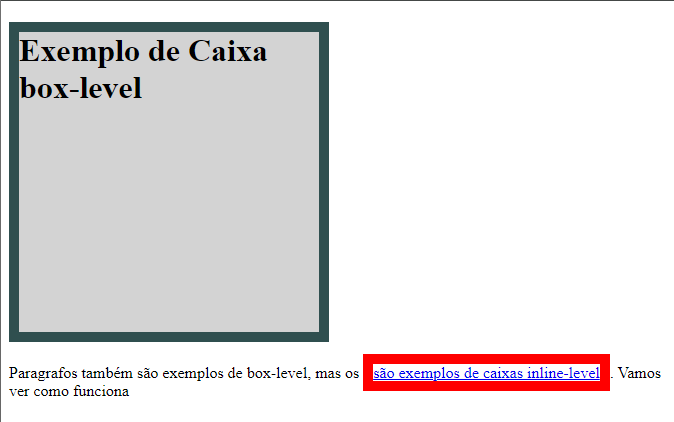
Eu posso inspecionar elemento no site e mexer nas características dentro do devtools para depois mudar no meu site!

Para mexer nas border eu posso usar o border-width que é a largura e border-style que é o estilo da borda (**pode ser solid-solida; dashed- tracejada; dotted- pontilhada; double- dupla; groove- 3d)** e caso queira adicionar cor é border-color.



Fazer tudo junto da borda: border: 10px solid red (**largura, estilo, cor)**

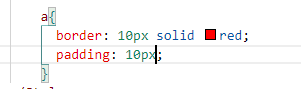




Se eu quiser mexer no “padding” geralmente dividido em quatro parte. (top;right;bottom;left) é recomendado seguir essa ordem

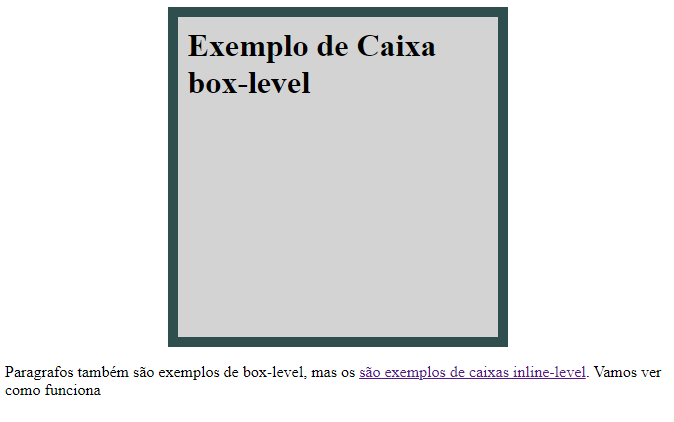
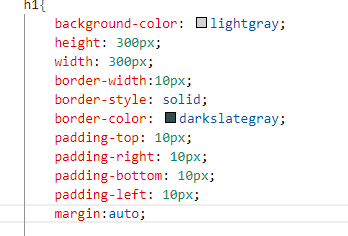


Fazer tudo junto: padding: 15px 10px 15px 10px (sent.horario) ou só colocar (10px -4 lados iguais a 10px) ou (15px 10px – cima baixo 15 e esquerda direita 10)

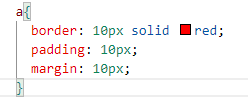


Se eu quiser mexer na “margin” geralmente dividido em quarto parte. (top;right;bottom;left) é recomendado seguir essa ordem // para centralizar um elemento na tela, de forma automática, devemos colocar margin: auto.





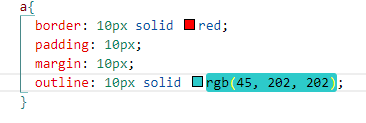
Fazer tudo junto: margin: 15px 10px 10px 15px (sent.horario) ou só colocar (10px -4 lados iguais a 10px) ou (10px auto 10px auto – cima e baixo 10 e centralizado).

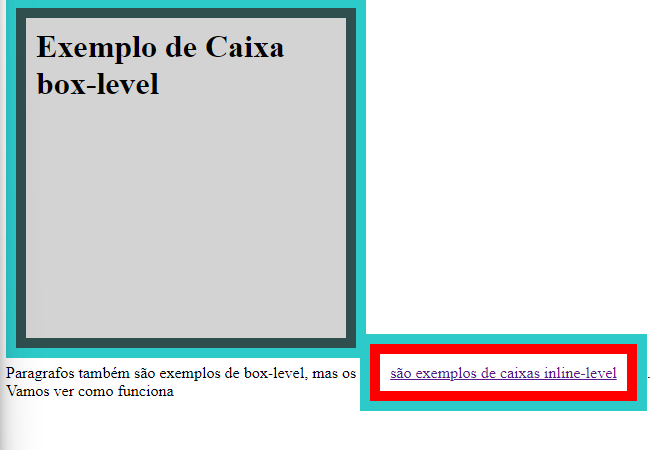


Para mexer nas “outline” eu posso usar o outline-width que é a largura e outline-style que é o estilo da borda (**pode ser solid-solida; dashed- tracejada; dotted- pontilhada; double- dupla; groove- 3d)** e caso queira adicionar cor é outline-color.



Fazer tudo junto da outline: outline: 10px solid green (**largura, estilo, cor)**





**Transformar box-level em inline-level e vice versa**:

Só colocar por exemplo *h1*{ display:inline; \*\* originalmente é block e *a*{display:block; \*\* originalmente é inline.

OBS: display: inline-block; É inline, com algumas características de block. (pode mudar tamanho, altura, largura!)



**Grouping Tags**

Antigamente era só as <div> block-level e a <span> inline-level e com ele colocava id tipo cabeçalho, menu, coteudo e etc.

Só que agora tem as grouping tags de semânticas!

<header> cabeçalho

<nav> para menus, links, navegação

<main> principal

<section> seção

<article> artigo

<aside> periférico ao artigo (ex: escritor / autor)

<footer> rodapé



Separando assim eu posso mexer no CSS



