Anotações Módulo 02

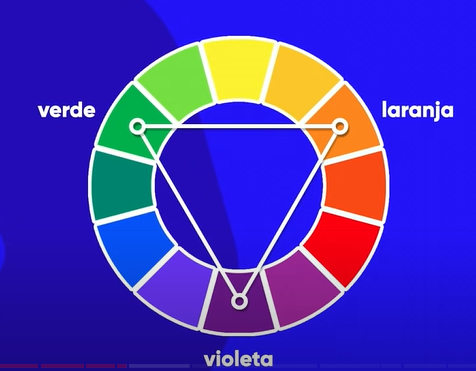
Acessar o pdf 13 para tirar dúvidas sobre a psicologia das cores.

**Círculo Cromático:** Serve para verificar a harmonia das cores.

**Cores Primárias:**



**Cores secundárias:**

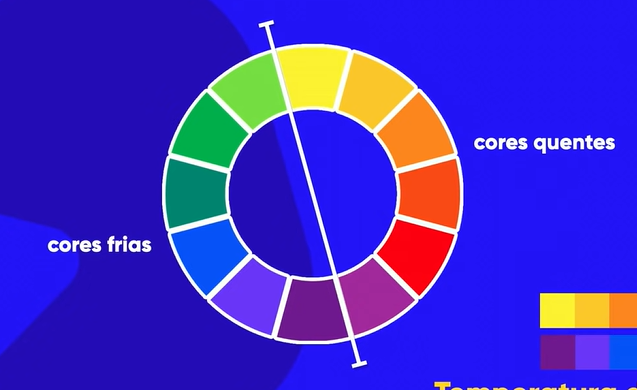


**Cores Terciárias:** São todas as outras cores, que são misturas das cores primárias e secundárias.

Ex de cores terciárias:



**Temperatura das cores:**



**Cores complementares:** Quando quiser usar uma cor que contrasta com outra, ou seja, que se destaque a cor da letra perante ao fundo por exemplo. Ver no círculo cromático a cor exatamente oposta a sua posição.

No exemplo abaixo, a cor que mais tem que contraste com o violeta é o amarelo.



**Cores Análogas:** São as cores parecidas, que não tem contraste uma com a outra. No círculo cromático, para saber a cor análoga da cor escolhida, é só ver as cores as lado da escolhida. Veja o exemplo abaixo:



**Cores Análogas e uma complementar:** Consiste em selecionar cores que são análogas e dão uma suavidade a paleta de cores, e uma completamente o oposto para dar um contraste. Exemplo:



**Cores Análogas relacionadas:** É uma técnica utilizada quando se quer um pequeno contraste em relação à paleta de cores. Nesse caso é escolhido duas cores, uma ao lado da outra, pula uma cor e utiliza a próxima. Veja o exemplo abaixo:

**Cores Intercaladas:** São parecidas com a opção anterior, mas com um contraste maior.



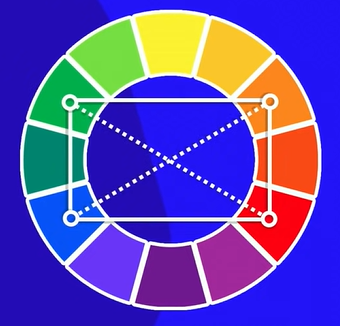
**Cores triádicas:** Consiste em escolher uma cor, pular as próximas 3 selecionar a próxima e assim continua. Exemplo:



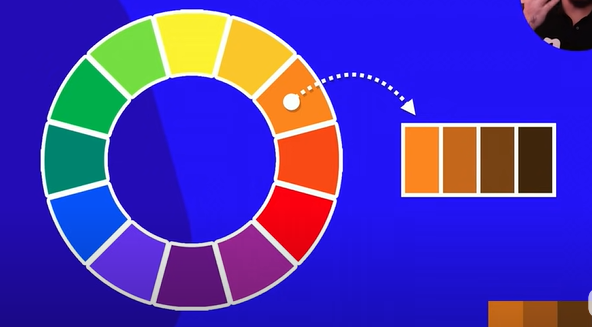
**Cores em quadrados:** Consiste em escolher uma cor, pular 2 e selecionar a próxima. Exemplo:



**Cores Tetrádicas:** Consistem em selecionar uma cor e próxima será a sua complementar. Selecionar uma outra cor qualquer e a próxima será a sua complementar. Exemplo:



**Monocromia:** Consiste em utilizar somente uma cor e alterar seu brilho e saturação, isso fará sejam geradas novas cores. Cria uma efeito de degrâde. Exemplo:



**Aula 09 e 10 – Estudo Básico de Tipografia**

**Anatomia do tipo:** Se refere a como vai ser escrito cada tipo de letra, por que cada fonte tem seus diferenciais.

**ITENS MÉTRICOS DE UMA FONTE**

**Altura-x:** A altura de todas as letras minúsculas de uma fonte são baseadas na letra x, ou seja, quando alguém vai desenvolver uma nova fonte, a primeira letra a ser desenhada é o x, pois todas as outras serão baseadas na sua altura, isso é chamado de altura-x. Abaixo esta uma imagem dessa representação com a fonte Times New Roman.



**Altura das maiúsculas:** Criado proporcionalmente a partir da altura das minúsculas.



**Ascendente e Descendente:** Algumas letras minúsculas como “b” e “k” representados no exemplo, apresentam um vazamento para cima da medida padrão da altura-x, essa medida é chamado de ascendente. Já as letras “g” e “Q” apresentam um vazamento para baixo da medida padrão, essa medida é chamada de descendente.



**Altura do Corpo:** É o somatório entre a altura-x, ascendente, descendente e a diferença da altura da maiúsculas representados acima, ou seja, o resultado é tamanho da altura do corpo da fonte.

**Importante:** Para realizar diversas configurações em uma fonte em CSS é necessário saber a altura do corpo.

**Itens Anatômicos de uma fonte**

**Importante:** Nem todas as fontes tem os componentes explicados a seguir.

**Serifa:** Serve basicamente para ajudar na leitura do usuário. A serifa faz com que nosso cérebro crie uma linha imaginária, mesmo sem percebermos, que acaba guiando o nosso cérebro e a facilite a leitura. No exemplo abaixo mostra esses traços circulados que são as serifas, e o tracejado seria a linha que nosso cérebro cria inconsciente para ajudar na leitura.



**Importante:** Há mais coisas no vídeo da aula que acredito que não precise anotar por não ser tão importante, mas caso queira rever o vídeo para relembrar:

Módulo 02, aula 10 - <https://www.youtube.com/watch?v=YZfzstEquas&list=PLHz_AreHm4dlUpEXkY1AyVLQGcpSgVF8s&index=10&ab_channel=CursoemVídeo>

**Categoria de Fontes**

**Serifadas:** Fontes que apresentam serifa.



**Sans-Serif:** Fontes que não apresentam serifa.



**Monoespaçadas:** Pode ser com serifa e sem serifa. São fontes em que todas as letras ocupam o mesmo espaço dentro do texto, independente da letra ou tamanho. Veja no exemplo abaixo a letra “l” e “a”, apesar do “l” ser menor em questão de espaçamento lateral, nessa fonte estão todos com o mesmo tamanho.



**Handwrinting, script ou scriptada:** Fontes que simulam escrita à mão, como se fosse em um papel.



**Display ou fontes comemorativas:** Fontes que não se preocupam em obedecer as regras anatômicas (No caso aqui no documento só explique a serifa, mas no vídeo tem mais regras descritas). Ás vezes não tem nenhuma dessas regras ou muito pouco.



**Aula 12**

**Medidas absolutas:**

cm, mm, in(polegada), px, pt(ponto), pc(paica)

**Medidas Relativas:**

em(Medida relativa ao altura-m, é uma medida em relação a letra "M" maiuscula), ex(Medida relativa ao altura-x), rem, vw(largura da viewport), vh(altura da viewport), %

O tamanho padrão das fontes nos navegadores normalmente é 16px.

Geralmente 16px é = 1em

É recomendável usar a medida "em" por que ela trabalha com porcentagem em cima dos pixels, fazendo com que o tamanho da fonte se ajuste em diversas telas.

**Importante:**

**Weight:** peso;

**Height:** altura;

**Width:** largura.

**Box Model**

Todo elemento visível em um site é uma caixa (Box model no caso), o exemplo abaixo mostra exatamente o exemplo de uma caixa do <H1>.



Então quando eu crio um seletor em CSS para esse h1, eu estou configurando a caixa que está exibindo esse elemento. Essas caixas tem dois tamanhos que podem ser configuráveis, sendo eles: **Height (altura) e Width (largura).**



**Border:** Propriedade em CSS que serve para criar bordas em volta do elemento que quiser, ou seja, a caixa. Veja que no exemplo abaixo a borda por padrão fica bem próxima do pontilhado da box model.



**Padding:** Tradução de padding pode ser entendida como preenchimento, ela serve para configurar esse espaço entre a borda e a box model, ou seja, é um **espaço interno** entre a caixa do elemento e a borda. Veja o exemplo abaixo:



**Margin:** A propriedade margin diferente do padding configura o espaço externo de uma box model. Veja que o elemento tem uma borda, porém o espaço que está sendo configurado é o externo a box model, isso faz com os elementos não fiquem colados um nos outros. Veja o exemplo abaixo:

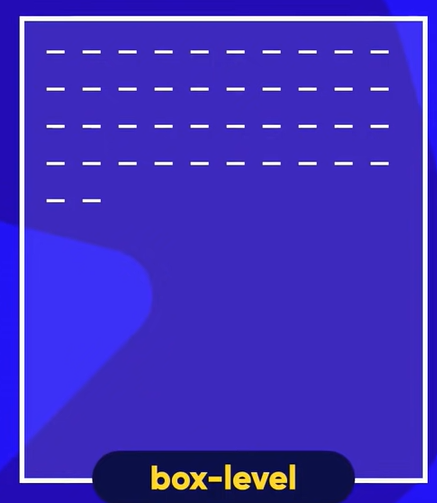


**Outline:** É um traçado feito igual o border, porém o border é feito dentro do box model do elemento e o outline é fora do box model. No exemplo abaixo o outline é o traçado amarelo.

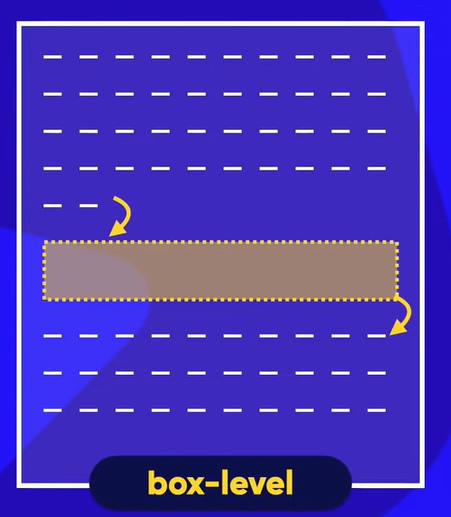


**Tipos de caixa**

**Box-level:**

****

Imagine que essa seja uma tela de navegador e as linhas tracejadas são parágrafos, que também é considerado uma caixa, e então eu queira inserir um outro tipo de elemento (tag), ou seja, para o navegador isso será considerado uma nova caixa. O que vai acontecer é que automaticamente há uma quebra de linha, onde é inserido a nova caixa que ocupa a largura de toda a linha, e após isso há outra quebra de linha para continuar o parágrafo, já que parágrafos também são caixas. Veja no exemplo abaixo:

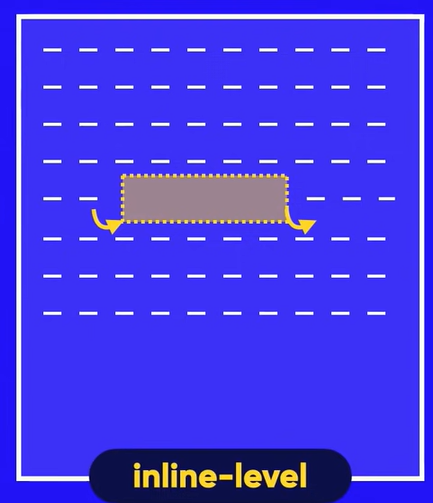
****

**Exemplo de tag box-level:** A tag <div> é uma tag box-level, uma das tags mais famosas em HTML. Será utilizada quando quiser criar um box que ocupe a largura inteira e gere quebra de linha. Veja mais algumas tags abaixo:



**Inline-level:**

Seguiremos o mesmo exemplo do box-leve para explicar o inline-level. Nesse caso não haverá quebra de linha, o conteúdo será escrito na frente na mesma linha onde termina o parágrafo, além disso a caixa não ocupará a largura de toda a linha, somente o necessário para que seu conteúdo caiba dentro da caixa, e o parágrafo continua exatamente após o término da caixa. Veja o exemplo abaixo:

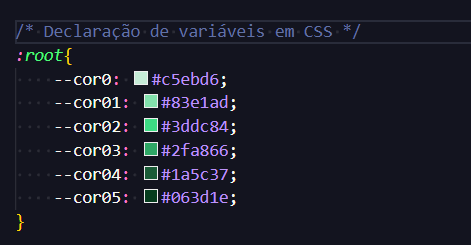


**Exemplo de tag inline-level:** A tag <span> é uma tag inline-level, utilizada quando você quiser que dois elementos, ou duas box-models, estejam imediatamente do lado uma da outra. Veja outros exemplos abaixo:



**Declarações de Variáveis em CSS**

Há uma maneira de declarar variáveis em CSS. Essas variáveis recebem algum estilo do CSS e dessa forma a linguagem reconhece aquela variável como se fosse o estilo em sim. Exemplo abaixo:



Nesse exemplo as variáveis estão recebendo cores, então quando eu quiser utilizar uma cor posso simplesmente usar o nome da variável declarada. Observe a imagem abaixo e **se atente a sintaxe**:

