**ANOTAÇÕES CURSO DE DESENVOLVEDOR WEB.**

**Repositórios públicos:**

**github.com/gustavoguanabara**

**gustavoguanabara.github.io**

**Referencias on-line:**

**Referencia MDN**

**W3C Standards**

**Whatwg living standard**

**W3Schools**

**LIVROS:**

**HTML5 Mark pilgrim**

**CSS eric meyer**

**Flexbox in CSS Estelle weyl**

**Grid Layout eric meyer**

**HTML5 e CSS3 Elizabeth e bruce**

**HTML E CSS altabooks projete e contrua**

**Use a cabeça HMLS e CSS altabooks (indicado para quem esta começando).**

**4 livros Mauricio Samy Silva HTML5**

**Curso de design Gráfico David Dabner**

**Design Senac bob e magie**

**Psicologia das cores e pensar em tipos Ellen lupton**

**Flexbox explained -- CSS Grid explained Jorge Montoya e Stephen Burge (2 livros).**

**Smashing HTML5 Bill Sanders**

**Smashing CSS3 Eric meyer**

**HISTÓRIA DA INTERNET.**

**Internet surgiu na Guerra fria quando o EUA PRECISOU CRIAR A ARPANET para se comunicar com seus postos militares. Depois que a demanda aumentou mudou para internetworking abreviando para internet.**

**Youtube/TNQsmPf24go - documentário sobre como funciona a internet.**

**COMO FUNCIOA A INTERNET:**

**Digitos binários (BIT): 0 e 1 -🡪 Como o aparelho eletrônico entende as coisas.**

**8 bits - BYTE**

**01000001 = letra A**

**Tabela de representação Código multibyte: UTF-8.**

**Múltiplos de bytes: 1024 bytes = 1kb referencia é sempre 210 .**

**1024kb = 1MB 1024 MB = 1GB 1024GB = TB 1024 TB = 1 PETABYTE; EXABYTES; ZETABYTES; YOTABYTES; e assim vai.**

**MB é diferente de Mb: MB é megabytes e Mb é Megabits. Cuidado.**

**Bytes é associado a armazenamento (por exemplo: um computador tem 16GBYTES de memória.)**

**Bits é associado a transmissão (por exemplo: Minha internet é de 350 Megabits).**

**Modem é um aparelho que faz modulção (tranformar onda quadrada para senoidal) e demodulação (transformar a onda senoidal em onda quadrada) de ondas.**

**Servidor DNS: Sistema de nome de domínio.**

**Dominio: Nome único; pago anualmente; tld´s**

**Hospedagem: Espaço para armazenar arquivos; Pago mensalmente; Espaço, memória, recursos. (tem que escolher bem).**

**Endereço completo se chama Url.**

**Dentro da Url tem partes:**

**Domínio: algumacoisa.com (.com é gtld ou seja domínios genéricos) o .BR é para identificar Brasil e outros países tbm tem os seus. Gtld é de genérico e CCtld é de countrycode**

**WWW. É um sub-domínio.**

**E a / é chamado de caminho.**

**Não se diz que programa em HTML e css, pois não são linguagem de programação.**

**HTML: Hypertext Markup Language. (linguagem de marcação para hipertexto)**

**Focada em conteúdo. (texto;imagem;vídeo;tabela;lista;etc)**

**CSS: Cascading style sheets ( Folhas de estilo em cascata)**

**Foco em design, atrativos visuais (tamanho, cores, sombras, posicionamento)**

**TEM TAMBEM O JAVASCRIPT que cria interações (menus; animações;popups;)**

**Pra que servem? Os três compõem a construção de um site.**

**HTML**

**Exemplo de titulo: conteúdo**

**Tags são coisas colocada entre sinais de < > chamdo de abertura de tag**

**Exemplo: <h1> Exemplo de título </h1> nesse caso <h1> é abertura de tag e </h1> é fechamento de tag.**

**Exemplo de parágrafo:**

**<p> exemplo de parágrafo </p>**

**Foto:**

**<img src= “foto.png” alt=“Exemplo de foto”> nesse caso o img meta hr não tem fechamento.**

**Src e alt são chamados de parâmetro e foto.png ou exemplo de foto são chamados de valor.**

**CSS**

**H1 { font-family: Arial;**

**font-size: 20pt;**

**color: blue color seria a propriedade blue seria o valor sempre esse par**

**Seletor declaração Cada declaração é um par de propriedade e valor.**

**TODA DECLARAÇÃO TEM QUE TER PONTO DE VIRGULA NO FINAL ( ; )**

**Estrutura básica de documento HTML**

**Vou desenvolver um site em html5**

**Todo HTML tem duas áreas bem definidas, head e body.**

**Primeira linha:**

**<!DOCTYPE html>**

**<html lang=”PT-br”>**

**<head>**

**<meta charset=“UTF-8”>**

**<meta name=“viewport” content=“width=device-width,initial-sacle=1.0”>**

**<title>Document</title>**

**</head>**

**<body>**

**<h1>Olá, Mundo!</h1>**

**</body>**

**</html>**

**Como funciona a html e as css?**

**HTML5 e css3 no servidor, manda para o computador, o navegador Le todas as linhas e mostra o resultado no computador.**

**Front-end:**

**Lado do cliente/**

**Desenvolvedor front-end vai fazer coisas que vão rodar para o lado do cliente.**

**Gerar arte visual para o usuário do cliente**

**Back-end:**

**Server side/ Lado do servidor**

**Desenvolvedor Back-end vai fazer a interação do código com o servidor.**

**Quais linguagem funcionam em servidor?**

**PHP;Javascript para servidor; Csharp; Phyton; Ruby e Java.**

**Saber ser desenvolvedor front-end e Back-end te torna um desenvolvedor fullstack.**

**DIREITO DE IMAGEM**

**Cuidado com o uso de imagens direto da internet. Buscar as imagens com licença para uso.**

**Sites recomendados: Unsplash; Pexels;**

**Formatos de imagens mais usados:**

**Jpeg (compacta a imagem)**

**png (também é compactado mas permite transparência/fundo transparente).**

**Tomar cuidado com o tamanho da imagem, em megas para não atrapalhar no site.**

**Para colocar a imagem no site, basta escrever img + enter. Para selecionar o arquivo imagem é Ctrl + espaço. Arquivo que esta em alguma pasta img + enter + ctrl + espaço + a pasta + arquivo. Para selecionar imagem da internet é img + enter + ctrl + espaço + cola da URL da imagem.**

**Para adicionar o ícone ou favicon é preciso pegar a imagem desejada, transforma-la em favicon no favicon.io e add o arquivo ícone na pasta do site. Após add na pasta, digitar acima do title “link” e selecionar o favicon. Após selecionar, apagar o que está selecionado automaticamente, ctrl + espaço e seleciona o arquivo ícone que está na pasta.**

**Hierarquia de títulos**

**H1 até H6**

**H1 é titulo principal**

**H2 é subtítulo**

**H3 é mais um e assim vai...**

**Semantica, significado, HTML:**

**Tomar cuidado com tags obsoletas. Olhar no site w3c**

**https://www.w3.org/TR/2014/REC-html5-20141028/obsolete.html**

**Estilo; forma: CSS.**

**Links externos**

**Para linkar com uma pagina externa sem abrir uma nova aba, seguinte comando: seleciona o texto link, ctrl+shift+p --- enter---a---enter, colar URL completa.**

**Para linkar com uma pagina externa abrindo uma nova aba, seguinte comando: seleciona o texto link, ctrl+shift+p --- enter---a---enter, colar URL completa, seguido de** Target:\_blanck Rel:External.

**Links Internos**

**Para criar links internos, é preciso colocar a “pagina arquivo” na mesma pasta. Faça isso pelo Visual Studio Code em novo arquivo (nome da pagina.html). Na primeira pagina usar o comando +shift+p --- enter---a---enter. Usar o segundo comando ctrl+espaço e selecionar a pagina 2 + rel: “next”.**

**Fazer o mesmo na pagina 2 se quiser criar um link para voltar na pagina 1 (rel: “prev”).**

**Quando for colocar um link externo de algum patrocinador por exemplo, fazer o comando: seleciona o texto link, ctrl+shift+p --- enter---a---enter, colar URL completa, seguido de** Target:\_blanck Rel:nofollow **.**

**Para um link interno que está em uma pasta fora da pagina atual, basta usar (../) para voltar uma pasta e selecionar o link correto.**

**PRESSIONAR ALT + O CURSOR ONDE QUER ALTERAR PARA ALTERAR MAIS DE UMA COISA AO MESMO TEMPO.**

**Iana/.org/assigments -🡪 lista de todos os midiastypes.**

**MODULO 2 – CSS**

**AULA 1 -**

**Aula sobre Cores no Design**

**1. Introdução ao tema:**

* **O capítulo 13 será focado em cores, indo muito além do básico (ex: representar cor em código).**
* **O objetivo é entender como escolher cores de forma eficaz para projetos web e design em geral.**

**2. Materiais necessários:**

* **Ter à mão: caderno para anotações, PDF do curso e computador.**

**3. A importância das cores:**

* **Cores afetam nossas emoções e decisões, mesmo sem percebermos.**
* **Estudos indicam que até 90% das decisões de compra estão relacionadas à aparência visual (especialmente cores).**
* **A Psicologia das Cores é fundamental para quem desenvolve sites, mesmo que não seja designer.**

**4. Exemplos de uso de cores por empresas:**

* **Azul: usado por Facebook, Twitter, LinkedIn, HP, Intel — transmite confiança, profissionalismo e segurança. Tem alta aceitação (≈ 45%) e baixa rejeição (≈ 1%).**
* **Vermelho: remete a emoção, amor, energia, mas também perigo e raiva. Usado por marcas como Netflix e McDonald’s. Deve ser usado com cuidado.**
* **Amarelo: transmite otimismo e felicidade. Combinado com vermelho, é eficaz para marcas de comida (McDonald’s, por exemplo).**
* **Marrom e dourado: associam-se a luxo, estabilidade e elegância (ex: sites de restaurantes sofisticados).**
* **Verde: relacionado a natureza, saúde e dinheiro, como usado pelo Greenpeace. Menos popular entre adolescentes do sexo feminino.**
* **Rosa: remete a doçura, romance e cuidados, sendo comum em cosméticos e produtos femininos.**
* **Roxo: transmite criatividade, poder e sabedoria, mas não prende tanto a atenção — usado em bancos e projetos inovadores.**

**5. Cuidados com contraste de cores:**

* **Evite fundo preto com texto branco em sites com muito texto (causa cansaço visual).**

**A combinação mais confortável é texto preto sobre fundo branco.**

**6. Projetos monocromáticos:**

**Algumas marcas utilizam variações de uma única cor (ex: Milka com tons de roxo).**

**7. Exemplos de aplicação prática:**

**Sites como o da Paola Krause (joias), Burger King, Nubank, e Google (Plan Your Planet) mostram como cores bem aplicadas criam identidade e sensações nos usuários.**

**Conclusão:**

**As cores não são só estética — elas têm um poder emocional e comercial. Saber usá-las com consciência é essencial para quem desenvolve sites. O próximo vídeo abordará combinações e contraste de cores.**

**AULA 2-**

**Aula: Como Representar Cores com CSS**

**1. Introdução:**

* **Continuação do Capítulo 13, agora focado em como representar cores no CSS.**
* **O professor reforça: não pule as aulas — entender cores é essencial para criar sites visualmente agradáveis e funcionais.**

**🧠 Quatro formas de representar cores no CSS:**

**1. Por nome (ex: blue, white, black):**

* **Método mais simples, porém limitado.**
* **Reconhece apenas 147 nomes de cores.**

**2. Por código hexadecimal (ex: #0000FF, #FFFFFF):**

* **Cada cor é formada por três pares de dígitos: R (vermelho), G (verde) e B (azul).**
* **Exemplo: #0000FF = 0 de vermelho, 0 de verde e 255 de azul (cor azul).**
* **Usa valores de 00 a FF (hexadecimal = base 16).**

**3. Por função RGB (ex: rgb(0, 0, 255)):**

* **Representa a quantidade de vermelho, verde e azul usando números decimais de 0 a 255.**
* **É equivalente ao hexadecimal, mas mais “legível”.**

**4. Por função HSL (ex: hsl(240, 100%, 50%)):**

* **Baseada em:**
  + **Hue (matiz): tonalidade da cor (0–360 graus).**
  + **Saturation (saturação): intensidade da cor (0–100%).**
  + **Lightness (luminosidade): clareza/escuro da cor (0–100%).**

**🧪 Transparência nas cores:**

**1. rgba() e hsla()**

* **Extensões do rgb() e hsl() com o quarto parâmetro a (alpha) que representa a transparência.**
  + **Exemplo: rgba(0, 255, 0, 0.5) = verde com 50% de opacidade.**
  + **Exemplo: hsla(240, 100%, 50%, 0.3) = azul com 30% de opacidade.**

**💻 Ferramentas e dicas práticas:**

**1. Visual Studio Code (VSCode):**

* **Ao passar o mouse sobre um valor de cor, o editor abre um seletor de cor interativo.**
* **Permite:**
  + **Alterar tonalidade, saturação, luminosidade e transparência.**
  + **Alternar entre os formatos: nome, hexadecimal, rgb, hsl.**

**2. GIMP (editor de imagem gratuito):**

* **Foi utilizado para mostrar valores RGB de uma cor.**
* **Excelente para quem quer explorar cores fora do navegador.**

**📌 Observações importantes:**

* **Todas essas formas representam a mesma cor, só mudando o formato.**
* **White (branco) pode ser:**
  + **white**
  + **#FFFFFF**
  + **rgb(255, 255, 255)**
  + **hsl(0, 0%, 100%)**

**🔜 Próximo passo:**

* **Na próxima aula será abordado o tema "harmonização de cores", ou seja, como escolher boas combinações de cores para o seu site.**
* **Serão usados slides e explicações visuais para facilitar o entendimento.**

**✅ Anote no seu caderno:**

* **As quatro formas de representar cores: nome, hex, rgb(), hsl()**
* **As versões com transparência: rgba() e hsla()**
* **Ferramentas: VSCode para testar cores e GIMP para análise de cores RGB.**

**AULA 3- Resumo – Aula: Harmonia de Cores no Design de Sites**

**Conceitos Fundamentais:**

**Círculo Cromático:**

* **Base do estudo de harmonia de cores.**
* **Apresenta as cores organizadas em uma roda, facilitando a identificação de combinações.**
* **Simplificado para 12 cores principais para facilitar o entendimento.**

**Classificações de Cores:**

* **Cores Primárias:  
  Amarelo, Vermelho e Azul.  
  Formam um triângulo equilátero no círculo cromático.**
* **Cores Secundárias:  
  Laranja, Violeta e Verde.  
  São obtidas a partir da mistura das primárias.**
* **Cores Terciárias:  
  Mistura de primárias com secundárias (ex: amarelo-esverdeado, vermelho-alaranjado).  
  Sempre se nomeiam com a cor primária primeiro.**

**Temperatura das cores:**

* **Cores frias:  
  Tons de azul, verde, violeta. Transmitem calma, frescor.**
* **Cores quentes:  
  Tons de vermelho, laranja, amarelo. Transmitem energia, calor.**
* **🎯 Tipos de Harmonia de Cores:**

**1. Cores Complementares:**

* **São opostas no círculo cromático (ex: azul ↔ laranja, vermelho ↔ verde).**
* **Geram forte contraste e são ideais para chamar atenção.**

**2. Cores Análogas:**

* **São vizinhas no círculo (ex: azul, azul-esverdeado, verde).**
* **Criam transições suaves, com pouco contraste e muito equilíbrio.**

**3. Análogas + Complementar:**

* **Uma cor principal + duas análogas + uma oposta (complementar).**
* **Cria equilíbrio + destaque.**

**4. Análogas Relacionadas:**

* **Duas cores próximas e uma terceira com leve contraste (ex: laranja, vermelho-alaranjado, vermelho).**
* **Harmonia sutil com leve variação.**

**5. Cores Intercaladas:**

* **Seleciona uma cor, pula uma ou duas, e escolhe as seguintes (ex: amarelo, laranja, vermelho).**
* **Gera uma paleta de contraste moderado.**

**6. Cores Triádicas:**

* **Três cores igualmente espaçadas no círculo (ex: amarelo, vermelho, azul).**
* **Criam equilíbrio e vivacidade.**

**7. Cores em Quadrado:**

* **Quatro cores espaçadas igualmente no círculo (ex: amarelo, vermelho-alaranjado, violeta, azul-esverdeado).**
* **Paleta rica e vibrante.**

**8. Cores Tetrádicas:**

* **Duas combinações de cores complementares (ex: verde e vermelho + azul e laranja).**
* **Formam retângulo no círculo cromático.**

**9. Monocromia:**

* **Usa uma única cor variando saturação e luminosidade.**
* **Gera uniformidade e elegância (usado em degradês e fundos).**

**Dicas Práticas para Criar Paletas:**

* **Parta de uma cor principal, que represente bem o produto ou marca.**
* **Uma paleta ideal tem de 3 a 5 cores (além de branco e preto).**
* **Se o cliente já tiver uma logo, a cor principal da identidade visual pode ser usada como base da paleta.**
* **Cada escolha deve considerar emoção, contraste e legibilidade.**
* **AULA 4 –** Resumo – Aula: Aplicando Harmonia de Cores na Prática
* Objetivo da aula: **Ensinar a usar ferramentas** gratuitas **e** intuitivas **para criar paletas de cores harmoniosas, baseadas nos conceitos aprendidos anteriormente (código de cores, círculo cromático, harmonia, contraste etc.).**

**Ferramentas apresentadas:**

**1.** [**Adobe Color (color.adobe.com)**](https://color.adobe.com/)

* **Gratuito e intuitivo.**
* **Gera paletas com base em:**
* Cores complementares
* Análogas
* Triádicas
* Quadradas
* Monocromáticas
* **Permite alterar matiz, saturação, brilho.**
* **Mostra os códigos** hexadecimais **prontos para copiar.**
* **Você pode inserir uma cor manualmente (ex: #b8fe44) e ele gera uma paleta automaticamente.**
* **Recurso de** “Extrair tema”**: você envia uma imagem (ex: logotipo) e ele extrai uma paleta com as cores predominantes.**
* **Recurso de** “Extrair degradê”**: gera gradientes automaticamente com base em imagens.**

**2.** [**Paletton (paletton.com)**](https://paletton.com/)

* **Ideal para simular** aplicações reais **de paletas (ex: visualização em sites).**
* **Permite criar paletas:**
* **Monocromáticas**
* **Adjacent (análogas)**
* **Triádicas**
* **Quadradas**
* **Possui uma aba de** exemplo de site **com prévia do design usando as cores escolhidas.**
* **Mostra os** códigos de cores **RGB/HEX prontos para uso.**
* **Muito útil para quem quer validar visualmente o impacto das cores.**

**3.** [**Coolors (coolors.co)**](https://coolors.co/)

* **Gera paletas aleatórias (pressionando a barra de espaço).**
* **Possui recurso de bloqueio: você pode travar uma cor que gostou, e a ferramenta sugere cores que combinem com ela.**
* **Permite importar imagens e extrair paletas ou gradientes.**
* **Fácil de usar para quem está perdido e quer inspiração rápida.**
* **Exemplos de aplicação prática:**
* **Usar logo de um cliente → extrair paleta oficial.**
* **Gerar degradê de fundo baseado em imagem.**
* **Criar contraste e harmonia com base em círculo cromático.**
* **Obter o código CSS pronto (ex: background-color: hsl(214, 98%, 54%);).**

**Dicas importantes:**

* **Para usar essas ferramentas corretamente, é** obrigatório **entender os conceitos ensinados nas aulas anteriores (cores complementares, análogas, triádicas etc.).**
* **Você pode gerar paletas com base em** imagens, ideias, códigos ou estilo desejado**.**
* **Uma boa paleta deve ter** 3 a 5 cores principais**, sem contar branco e preto.**

**CONFIGURAÇÕES GLOBAIS**

**USA-SE ( \*{} ) PARA CONFIGURAÇÕES GLOBIAS COMO A ALTURA DO GRAFIENTE DE COR.**

**AULA 5-**

**Resumo – Aula: Tipografia (Parte 1 - Teoria)**

**Objetivo da aula:  
Explicar o que é tipografia, sua história, importância no design e como diferentes fontes transmitem emoções. Essa é a parte teórica (1/2) do conteúdo sobre fontes.**

**📚 O que é tipografia?**

* **Tipografia é a arte e o estudo da criação, organização e aplicação de tipos (letras) com fins de leitura e comunicação visual.**
* **O termo vem do grego:**
  + **"Typos" = impressão**
  + **"Graphia" = escrita  
    → Tipografia = escrita impressa**

**📜 Breve história da tipografia**

1. **Antes da imprensa:**
   * **Livros eram copiados à mão por monges copistas, também chamados de amanuenses.**
   * **O processo era lento, caro e exclusivo da elite.**
2. **Invenção da imprensa:**
   * **Johannes Gutenberg, no século XV (cerca de 1450), inventou a prensa com tipos móveis.**
   * **Isso revolucionou a produção de livros: a tipografia passou a ser reproduzível e mais acessível.**
   * **Ainda que chineses tenham criado algo similar antes, Gutenberg é conhecido como o pai da imprensa ocidental.**

**🔠 Importância da tipografia no design**

* **As fontes, assim como as cores, comunicam sensações e emoções.**
* **Um site pode ter um ótimo conteúdo, mas falhar no impacto visual se não usar fontes adequadas.**

**🎭 Fontes e emoções (exemplos práticos)**

1. **Palavra: AMOR**
   * **Fonte mais suave e cursiva comunica afeto e delicadeza.**
   * **Fonte rígida e com traços retos não transmite bem essa ideia.**
2. **Palavra: IMPORTANTE**
   * **Fonte grossa, firme, reta → transmite urgência, seriedade, atenção.**
   * **Fonte leve e descontraída não combina com o propósito.**
3. **Termo: FEITO À MÃO**
   * **Fonte que simula escrita manual comunica autenticidade e personalização.**
   * **Fonte tradicional digitalizada não transmite o mesmo sentimento.**

**📌 Conclusão da aula**

* **Tipografia não é só estética: ela influencia diretamente a experiência do usuário.**
* **Boas fontes, combinadas com boas cores e boas imagens, compõem a base visual de qualquer projeto web.**
* **No próximo vídeo, será abordada a anatomia da letra (partes que compõem um tipo) e as categorias de fontes (ex: serifadas, sem serifa, manuscritas, etc.).**

**AULA 6 –**

Revisão da Parte 1

* **Tipografia é a arte de criar e aplicar fontes.**
* **Fontes transmitem** emoções **e influenciam a** legibilidade**.**
* **O cérebro** não lê letra por letra**, mas a** palavra como um todo**.**

Anatomia dos tipos (partes que compõem as letras)

Alturas e medidas

* Altura X**: altura das letras minúsculas, baseada no “x”.**
* Altura das maiúsculas**: linha limite superior das letras maiúsculas.**
* Ascendente**: parte superior que ultrapassa a altura x (ex: “b”, “k”).**
* Descendente**: parte inferior que ultrapassa a linha base (ex: “g”, “q”).**
* Altura do corpo**: soma da altura x + ascendentes + descendentes.**

Partes geométricas das letras

* Serifa**: pequenos traços no fim das hastes. Ajudam a guiar o olho na leitura.**
* Haste**: linha principal vertical ou diagonal (ex: “l”, “k”).**
* Filete**: traço que une duas hastes.**
* Arco**: curva ligada à haste (ex: “n”).**
* Esporão**: saliência que apoia uma letra (ex: “b”).**
* Vértice**: encontro de dois traços em ângulo (ex: “v”).**
* Terminal**: extremidade solta de uma letra (ex: perna do “r”).**

Partes com nomes “humanos”

* Braço**: traço que se projeta para cima ou para os lados (ex: “k”).**
* Perna**: traço descendente (ex: “k”).**
* Pé**: base inferior (ex: “l”).**
* Barriga**: arco curvado que forma um “balão” (ex: “b”).**
* Olho**: espaço interno fechado (ex: “o”, “g”).**
* Orelha**: pequeno gancho em letras como “g”.**
* Cauda**: extensão decorativa na parte inferior (ex: “g”, “y”).**
* Espinha**: curva em “s”.**

Glifo, Fonte e Família Tipográfica

* Glifo**: cada caractere (letra, número, símbolo).**
* Fonte**: conjunto de glifos com estilo unificado.**
* Família tipográfica**: variações da mesma fonte (ex: Light, Regular, Bold, Extra Bold).**

Categorias de fontes

Serifadas

* **Têm** serifas**.**
* **Boa legibilidade em** textos impressos longos**.**
* **Ex:** Times New Roman**.**

Sem serifa (Sans-serif)

* **Visual mais** limpo e moderno**.**
* **Indicadas para** textos em tela**.**
* **Ex:** Open Sans**,** Arial**.**

Monoespaçadas (Monospace)

* **Cada caractere ocupa o** mesmo espaço horizontal**.**
* **Úteis para** código-fonte **e** exibição técnica**.**
* **Ex:** Courier New**.**

Script / Handwrite

* **Simulam** escrita manual**.**
* **Usadas em títulos e peças visuais com** toque humano/artístico**.**
* **Ex:** Pacifico**,** Dancing Script**.**

Display / Comemorativas

* **Fontes decorativas para uso** pontual **(títulos, logos, cartazes).**
* **Não seguem necessariamente padrões anatômicos.**
* **Ex: fontes de filmes como** Jurassic Park**.**

Recomendações práticas

* **Para** textos longos em tela **→** Sans-serif**.**
* **Para** impressos **→** Serifadas**.**
* **Evite fontes com** pouco contraste **ou muito decorativas em textos corridos.**
* **Use** fundo claro + texto escuro **para leitura (ou o inverso em ambientes escuros).**
* **Use fontes comemorativas** apenas para enfeite ou chamadas visuais**.**

**Aula 6 –**

**AULA 7 –**

## 📐 Unidades de medida para fontes

### 🔸 1. Medidas Absolutas

**São fixas, não se adaptam ao tamanho da tela:**

* **cm → centímetro**
* **mm → milímetro**
* **in → polegada (1 in = 2.54 cm)**
* **px → pixel (✅ *recomendado*)**
* **pt → ponto (muito usado em Word, NÃO recomendado para telas)**
* **pc → pica (medida tipográfica, obsoleta para web)**

⚠️ Não recomendadas**: cm, mm, in, pt, pc**✅ Recomendado pela W3C**: px**

### 🔹 2. Medidas Relativas

**Se ajustam dinamicamente conforme o contexto ou o elemento pai:**

* **em → relativo ao** tamanho da fonte atual
* **1em = tamanho atual (ex: se base = 16px, então 2em = 32px)**
* **rem → relativo ao** tamanho da fonte raiz **(<html>)**
* **vw → relativo à largura da viewport (ex: 10vw = 10% da largura da tela)**
* **vh → relativo à altura da viewport**
* **% → relativo ao elemento pai**

✅ Recomendado pela W3C**: em (melhor para sites responsivos)**

## 🔁 Diferença entre em e rem

| **Unidade** | **Base** | **Exemplo (base = 16px)** |
| --- | --- | --- |
| **em** | **Elemento pai** | **2em = 32px (se pai = 16px)** |
| **rem** | **Elemento raiz (<html>)** | **Sempre relativo à base de 16px** |

## 📏 Exemplo prático no CSS

**p {**

**font-size: 1em; /\* mesmo que 16px (padrão) \*/**

**}**

**h1 {**

**font-size: 2em; /\* 32px, pois é o dobro \*/**

**}**

**.small {**

**font-size: 0.75em; /\* 75% do tamanho base \*/**

**}**

**.texto {**

**font-size: 18px; /\* tamanho fixo \*/**

**}**

## 🎯 Recomendações práticas da W3C

* **Para** telas **(sites): use** px **ou** em
* **Evite medidas tipográficas (pt, pc) e físicas (cm, in)**
* **Use em/rem para** responsividade **e** acessibilidade
* **O tamanho** padrão **de fonte nos navegadores é 16px ou 1em**
* **Evite valores quebrados com vírgula. Use** ponto **(0.5, não 0,5)**

## 🛠️ Vantagens de usar em

* **Escalável em diferentes dispositivos**
* **Facilita design responsivo**
* **Boa prática para acessibilidade**

### ✅ Conclusão

**Agora você já sabe:**

* **Como usar font-size corretamente**
* **A diferença entre unidades absolutas e relativas**
* **Que px e em são os padrões recomendados**
* **Como aplicar esses conhecimentos em CSS**

**AULA 8 –**

**1. Peso da fonte – font-weight**

* **Define o** quão fina ou grossa **será a fonte.**
* **➤ Palavras-chave:**
* **lighter → mais leve**
* **normal → padrão (equivale a 400)**
* **bold → negrito (equivale a 700)**
* **bolder → mais negrito**
* **➤ Valores numéricos:**
* **Varia de 100 a 900, em múltiplos de 100**
* **100 = ultra leve**
* **400 = normal**
* **700 = bold**
* **900 = extra bold**
* ⚠️ Nem todas as fontes suportam todos os pesos.

2. **Estilo da fonte – font-style**

* **Define o** estilo **da fonte:**
* **normal → padrão**
* **italic → itálico**
* **oblique → inclinado (menos comum)**
* **⚠️ Nem todas as fontes possuem versões itálicas.**

3. Sublinhado – **text-decoration**

* **Define** efeitos decorativos **no texto:**
* **underline → sublinha o texto**
* **Pode ser combinado com outros estilos:**
* **h1 {**
* **font-style: italic;**
* **font-weight: bold;**
* **text-decoration: underline;**
* **}**

4. **Propriedade shorthand – font**

* **Permite condensar várias propriedades de fonte em uma** única linha**.**
* **✅** Ordem correta:
* **font: font-style font-weight font-size font-family;**
* **➤** Exemplo completo:
* **h1 {**
* **font: italic bold 2em 'Work Sans', sans-serif;**
* **}**
* **⚠️** Ordem obrigatória! **Se inverter, pode dar erro.**

Exemplo prático com variações:

* **/\* Forma longa \*/**
* **h2 {**
* **font-style: normal;**
* **font-weight: 700;**
* **font-size: 24px;**
* **font-family: 'Arial', sans-serif;**
* **}**
* **/\* Forma curta (shorthand) \*/**
* **h2 {**
* **font: normal 700 24px 'Arial', sans-serif;**
* **}**

5. **Fontes com múltiplos pesos e estilos**

* **Algumas fontes (como Arial, Verdana, Times) têm poucas variações.**
* **Outras, como Work Sans (usada neste capítulo), têm variações de 100 até 900.**
* **Para funcionar, essas variações precisam ser** importadas corretamente **(o Guanabara mostra como fazer isso na próxima aula).**

**✅** Conclusão

* **Agora você já sabe:**
* **Usar font-weight com valores numéricos ou palavras-chave**
* **Aplicar font-style e text-decoration**
* **Criar atalhos com o font shorthand**
* **Combinar tudo em uma única linha, de forma organizada.**

**AULA 9 –**

1. Por que usar fontes externas

* **Fontes padrão do sistema (Arial, Times, Verdana etc.) são** limitadas**.**
* **Usar fontes externas** enriquece o visual **do site.**
* **O** Google Fonts **é uma das principais opções, gratuito e confiável.**

2. Acessando o Google Fonts

* **Acesse:** [**https://fonts.google.com**](https://fonts.google.com/)
* **No site você pode:**
  + **Pesquisar fontes por nome**
  + **Filtrar por categoria (serifada, sem serifa, manuscrita, monoespaçada etc.)**
  + **Visualizar diferentes estilos**
  + **Ver prévias com seu próprio texto**
  + **Selecionar famílias de fonte e estilos específicos**

3. Como importar uma fonte do Google Fonts

### ➤ Passo a passo:

1. **Escolha a fonte e o estilo (ex: Kaushan Script - Regular)**
2. **Clique em** "Select style"
3. **Vá até a aba de incorporação (Embed)**
4. **Copie o código dentro de @import url(...)**
5. **Cole esse código no início do <style> ou do seu arquivo CSS externo**

**@import url('https://fonts.googleapis.com/css2?family=Kaushan+Script&display=swap');**

4. Usando a fonte no CSS

**Depois de importar a fonte, aplique normalmente com font-family:**

**h1 {**

**font-family: 'Kaushan Script', cursive;**

**font-size: 3em;**

**}**

5. Importando múltiplas fontes

**Você pode importar várias fontes com diferentes pesos e estilos (ex: Work Sans de 100 a 900).  
Escolha os estilos desejados e o Google Fonts gerará a @import completa.**

6. Boas práticas com fontes

* Use no máximo 3 ou 4 famílias tipográficas **diferentes por site**
* **Sempre inclua uma fonte** genérica de fallback **no final:**

**font-family: 'Work Sans', Arial, sans-serif;**

* **Evite exageros para não comprometer a** performance e legibilidade

7. O que fazer se a fonte não estiver no Google Fonts

**Se o cliente tiver uma fonte comprada ou exclusiva:**

* **No** próximo vídeo**, o curso ensina como usar arquivos .ttf, .woff etc.**
* **Isso exige subir a fonte para o projeto e usar @font-face no CSS**

Dica final

* **Use a ferramenta de** visualização do Google Fonts **para testar fontes com seu texto.**
* **Prefira usar a** opção @import **para incluir diretamente no CSS.**
* Não use muitas fontes **diferentes ou seu site ficará visualmente poluído.**

**AULA 10 –**

Objetivo da aula

* **Aprender a utilizar fontes personalizadas que** não estão no Google Fonts**, como fontes compradas ou baixadas de sites como** [**daFont.com**](https://www.dafont.com/)**.**

1. Quando usar essa técnica

* **Quando o cliente possui uma** fonte específica em TTF, OTF ou WOFF
* **Quando a fonte** não existe no Google Fonts
* **Quando deseja um visual** muito personalizado **(ex: fontes com ícones, corações etc.)**

2. Sites confiáveis para baixar fontes

* [**www.dafont.com**](https://www.dafont.com/)
* [**www.1001fonts.com**](https://www.1001fonts.com/)
* [**www.fontspace.com**](https://www.fontspace.com/)
* **⚠️** Cuidado com:
* **Fontes** sem acentos **(verifique os glifos antes de usar)**
* **Fontes** somente em maiúsculas
* **Formatos incompatíveis com navegadores antigos**

3. Formatos de arquivos de fonte

| * **Formato** | * **Nome técnico** | * **Compatibilidade** |
| --- | --- | --- |
| * **.ttf** | * **TrueType** | * **Boa** |
| * **.otf** | * **OpenType** | * **Muito boa** |
| * **.woff** | * **Web Open Font Format** | * **Melhor para web** |
| * **.woff2** | * **Versão comprimida** | * **Melhor ainda** |
| * **.svg** | * **Vetorial (antigo)** | * **Obsoleto** |
| * **.eot** | * **Embedded OpenType** | * **IE antigo** |

4. Estrutura recomendada de pastas

* **projeto/**
* **│**
* **├── index.html**
* **├── style.css**
* **├── fonts/**
* **│ ├── fonte1.ttf**
* **│ ├── fonte1.otf**

5. Como usar **@font-face**

* **No seu CSS (ou dentro da <style> no HTML):**
* **@font-face {**
* **font-family: 'love';**
* **src: url('fonts/LoveLarry.otf') format('opentype'),**
* **url('fonts/LoveLarry.ttf') format('truetype');**
* **font-weight: normal;**
* **font-style: normal;**
* **}**
* **Depois, use normalmente:**
* **h1 {**
* **font-family: 'love', serif;**
* **font-size: 3em;**
* **}**

6. Dicas importantes

* **O nome após font-family (ex: 'love') é** personalizável**.**
* **Sempre informe o** tipo de formato **usando format('...').**
* **Inclua uma** fonte genérica de fallback **no final.**
* **Se estiver usando arquivo CSS externo, use** caminho relativo**:**
* **url('../fonts/nome\_da\_fonte.ttf')**

7. Pode misturar com Google Fonts?

* **✅ Sim! Você pode importar uma fonte do Google** e também usar fontes personalizadas **no mesmo projeto.**

Resumo prático

* **Use @font-face para carregar fontes baixadas**
* **Mantenha os arquivos organizados em uma pasta fonts/**
* **Teste em diferentes navegadores**
* **Prefira usar fontes do Google Fonts quando possível (mais leves e padronizadas)**

**AULA 11-**

## Descobrindo Fontes Usadas em Qualquer Site

### Objetivo da aula

**Aprender a** identificar as fontes **utilizadas em um site que você está visitando, para inspiração ou uso próprio, com o auxílio de uma extensão do Google Chrome.**

## 1. Quando usar essa técnica

* **Para** descobrir **qual fonte foi usada em um site bonito.**
* **Para** reutilizar **essa fonte no seu próprio projeto (se for gratuita ou estiver no Google Fonts).**
* **Para** copiar a estética **de títulos, menus, textos ou cores.**

## 2. Ferramenta usada

Extensão para Google Chrome: [WhatFont ou Fonts Ninja]

**✅ Recomendado pelo curso:** Fonts Ninja

Passo a passo para instalar:

1. **Acesse:** [**chrome.google.com/webstore**](https://chrome.google.com/webstore)
2. **Pesquise por Fonts Ninja**
3. **Clique em** "Usar no Chrome" **e depois em** "Adicionar extensão"
4. **Um ícone verde com uma máscara ninja vai aparecer na barra do navegador**

## 🔹 3. Como usar o Fonts Ninja

1. **Acesse qualquer site que deseja investigar.**
2. **Clique no ícone do** Fonts Ninja**.**
3. **Passe o mouse sobre os textos.**
4. **O plugin mostra:**
   * Nome da fonte
   * Peso **(ex: bold, light)**
   * Tamanho da fonte (em px)
   * Cor do texto
   * Espaçamento entre letras
   * **Indicação se é fonte** real ou imagem

**📌** Se for imagem: **o plugin** não funciona **(ex: logos, banners com texto em imagem).**

## 🔹 4. Exemplos de uso

* **No site da Hostnet, foi mostrado que** só uma fonte **era usada, com variações de peso e tamanho.**
* **No site do Curso em Vídeo, identificou fontes como Berkshire Swash e Josefin Sans.**

## 🔹 5. Como usar a fonte identificada

* **Copie o nome da fonte.**
* **Vá até:**
  + [**Google Fonts**](https://fonts.google.com/) **– procure e incorpore diretamente**
  + [**Dafont**](https://www.dafont.com/) **– baixe a fonte (se disponível)**
* **Se encontrar o arquivo .ttf ou .otf, use @font-face como aprendido na aula anterior.**

## 🔹 6. Extras

* **Você pode testar tamanhos e pesos dentro do plugin.**
* **Pode** copiar o código da cor **da fonte diretamente da extensão.**
* **Pode também ver a** família inteira **da fonte em uso.**

## 📌 Importante lembrar

* **O Fonts Ninja tem uma versão paga com recursos extras (ex: experimentar a fonte no computador), mas** a versão gratuita já cumpre bem o papel **de identificação.**
* **Se o texto for imagem, você não conseguirá capturar a fonte dessa forma.**
* **Sempre teste a fonte no** [**Google Fonts**](https://fonts.google.com/) **ou no site oficial da fonte para verificar a** licença de uso**.**

**AULA 12-**

**🎯** Objetivo da aula

* **Aprender a** descobrir qual fonte foi usada **em um texto contido dentro de uma imagem, usando inteligência artificial e visão computacional.**

**🔹** 1. Contexto

* **No vídeo anterior aprendemos a descobrir fontes em** textos de sites (HTML) **usando a extensão** Fonts Ninja**.**
* **Agora, o foco é em** textos contidos dentro de imagens **(como logos, panfletos, banners, etc).**

2. Sites utilizados

* **✅** 1) [**WhatFontIs.com**](https://www.whatfontis.com/)
* Mais eficiente **e com** melhores resultados
* **Permite ajustar contraste, brilho, separar letras grudadas e até aplicar redução de ruído**
* **Mostra** fontes pagas e gratuitas **semelhantes**
* **Permite testar mais de uma área da imagem**

2) [**FontSquirrel.com**](https://www.fontsquirrel.com/matcherator)

* **Mais simples**
* **Identifica algumas fontes, mas** com menor precisão
* **Também mostra sugestões gratuitas e pagas**

3) [**MyFonts.com**](https://www.myfonts.com/WhatTheFont/)

* **Interface parecida com os outros**
* **Foco maior em fontes** comerciais
* Boa opção secundária**, caso os outros não identifiquem**

3. Como usar os sites (passo a passo comum)

* **Tenha a** imagem com o texto **salvo no seu computador**
* **Acesse um dos sites listados**
* **Faça o** upload da imagem
* **Selecione a área da imagem com o texto desejado**
* **Ajuste brilho, contraste e ruído (se necessário)**
* **Separe letras grudadas manualmente (se necessário)**
* **Preencha os campos informando as letras identificadas**
* **O site exibirá sugestões de fontes** iguais ou semelhantes
* **Verifique se a fonte é** gratuita **ou** paga

Dicas importantes

* Nem sempre a fonte é identificada corretamente. **Teste os três sites se necessário.**
* **Prefira imagens com:**
* Alto contraste
* Pouco ruído
* Textos nítidos e separados
* **Se a fonte for paga, o site mostra o preço e o link para compra.**
* **Se quiser reutilizar a fonte, procure no** [**Google Fonts**](https://fonts.google.com/) **ou** [**DaFont**](https://www.dafont.com/) **para ver se há alternativas gratuitas.**
* Evite fontes que não tenham acentos**, se o seu site for em português.**
* **Utilize a técnica aprendida anteriormente com @font-face para aplicar fontes baixadas.**

**🧾** Resumo final

| * **Site** | * **Destaque Principal** |
| --- | --- |
| * **WhatFontIs** | * **Melhor precisão, opções avançadas de ajuste** |
| * **FontSquirrel** | * **Simples e rápido, mas menos preciso** |
| * **MyFonts** | * **Foco comercial, útil para fontes populares** |

**AULA 12 –**

Encerramento do Capítulo 14 – Fontes e Alinhamento de Texto

**✅** Resumo do que foi aprendido no capítulo:

* **O que é** tipografia **e o que são** famílias tipográficas
* **Diferença entre** fontes serifadas, sem serifa e manuscritas
* **Como aplicar** fontes genéricas
* **Como usar** Google Fonts
* **Como importar** fontes externas personalizadas **com @font-face**
* **Como identificar fontes de** sites (com texto) **e** imagens (OCR)
* **Uso de** peso**,** estilo **e** shorthand font
* **E agora:** alinhamento e indentação de texto

Tipos de alinhamento com CSS

* **Utiliza-se a propriedade text-align. Os valores mais comuns são:**

| * **Valor** | * **Descrição** |
| --- | --- |
| * **left** | * **Alinha o texto à esquerda *(padrão)*** |
| * **right** | * **Alinha o texto à direita** |
| * **center** | * **Centraliza o texto** |
| * **justify** | * **Justifica o texto dos dois lados** |

**Exemplo:**

* **h1 {**
* **text-align: center;**
* **}**
* **h2 {**
* **text-align: right;**
* **}**
* **p {**
* **text-align: justify;**
* **}**

Recuo de parágrafo

* **Usa-se a propriedade text-indent para** iniciar um parágrafo com recuo **(como na formatação de redação).**
* **p {**
* **text-indent: 30px;**
* **}**
* **❌** Não use mais a tag **<center>**
* **A tag <center> é** obsoleta**.**
* **A centralização e demais formatações** devem ser feitas no CSS**, não no HTML.**
* **HTML atual é** semântico**, ou seja, estrutura e significado; estética é com CSS.**
* **🧠** Importante lembrar
* **Alinhamento padrão do navegador: text-align: left**
* **Alinhamento é considerado** estético**, logo, deve estar no CSS**
* **Boas práticas indicam** separar conteúdo e estilo.

**AULA 13 –**

**Sem resumo por enquanto.**

**Tudo que em HTML é id – em CSS é #.**

**Tudo que em HTML é Class – em CSS é . (ponto)**

**AULA 14 –**

Resumo – Seletores Personalizados (Capítulo 15)

#### Diferença entre id e class

* id **é um identificador** único **para um elemento.**
  + **Só pode haver** um elemento com o mesmo id **em um documento HTML.**
  + **É referenciado no CSS com #, exemplo: #titulo.**
* class **é usada para** agrupar vários elementos com estilo comum**.**
  + **Pode ser usada** em múltiplos elementos**.**
  + **É referenciada no CSS com ., exemplo: .destaque.**

#### Regras de nomeação

* **Evite dar nomes baseados** na aparência **(.vermelho, .azul).**
* **Prefira nomes baseados** na função **do elemento (.basico, .avancado, .destaque).**

#### Aplicando múltiplas classes

* **Um elemento pode ter** mais de uma classe**:**
* **<h2 class="avancado destaque">CSS Avançado</h2>**
* **No CSS, cada classe define estilos independentes:**
* **.avancado { color: red; }**
* **.destaque { background-color: rgba(255,255,0,0.4); }**

#### Misturando id e class

* **É possível usar id e class** no mesmo elemento**.**
* **O id tem** maior prioridade **sobre a class quando houver conflito de estilos.**
* **Exemplo:**
* **<h1 id="principal" class="titulo">Título</h1>**
* **.titulo { color: blue; }**
* **#principal { color: red; } /\* prevalece \*/**

#### Hierarquia de seletores

* **Estilos mais específicos (como id)** sobrescrevem **os mais genéricos (class, tag).**
* **Há uma** herança **natural: por exemplo, tudo que estiver em body será herdado, a menos que sobrescrito.**

#### Criando estilos com .class no CSS

**.basico { color: #60b25c; }**

**.intermediario { color: #ffcc00; }**

**.avancado { color: #cc3333; }**

#### Exemplo prático em HTML

**<h2 class="basico">HTML Básico</h2>**

**<h2 class="intermediario">HTML Intermediário</h2>**

**<h2 class="avancado destaque">HTML Avançado</h2>**

#### text-indent e alinhamento de texto

* **text-align: left | center | right | justify → alinha o texto.**
* **text-indent: 30px → aplica** recuo no início de parágrafo **(como em redações).**

**AULA 15 –**

**✅** Resumo – Pseudo-classes em CSS (Capítulo 15)

**O que são pseudo-classes?**

* Pseudo-classes **definem** estados específicos **de um elemento HTML (ex: quando o mouse está sobre ele, quando está ativo, vazio, etc.).**
* **Usam o símbolo : no CSS.**
* Exemplo:
* **div:hover {**
* **background-color: yellow;**
* **}**
* **Significa: “quando o mouse estiver sobre uma div, ela ficará amarela”.**

Exercício prático com **<div>**

* **🔸 Etapas do exercício:**
* **Criadas 3 <div> com width e height de 200px, background-color cinza claro e display: inline-block.**
* **Cada <div> é identificada com id diferente (div1, div2, div3).**
* **Aplicado :hover para mudar a cor ao passar o mouse:**
* **div:hover {**
* **background-color: yellow;**
* **}**

Exemplo interativo – Mostrar conteúdo escondido ao passar o mouse

* **Uma <div> com um texto visível e um <p>** escondido **com display: none.**
* **Ao passar o mouse na <div>, o parágrafo escondido aparece com cor personalizada.**
* CSS:
* **div p {**
* **display: none;**
* **}**
* **div:hover p {**
* **display: block;**
* **color: white;**
* **background-color: red;**
* **}**
* **Esse efeito simula o funcionamento de** menus dropdown **(exibidos ao passar o mouse).**

**Dicas importantes:**

* **div:hover permite criar efeitos de destaque, animações e menus interativos.**
* **É possível aplicar pseudo-classes em:**
* **a:hover, input:focus, li:nth-child(2), etc.**
* **Use display: none e display: block para esconder e mostrar elementos.**
* **Para selecionar** filhos **de um elemento, usa-se elemento filho no seletor CSS:**
* **div p { ... }**

**Conclusão:**

* **Você aprendeu:**
* **O que são pseudo-classes.**
* **Como usar :hover para destacar elementos.**
* **Como esconder e exibir elementos dinamicamente com CSS.**
* **A criar interações visuais simples** sem JavaScript**.**

**AULA 16 –**

Resumo – Capítulo 15 (Parte final): Pseudo-classes e Pseudo-elementos

* **🔹 Revisão até aqui:**
* **Já vimos seletores básicos, identificadores (id) e classes.**
* **No vídeo anterior, usamos a pseudo-classe :hover.**
* **Agora, aprendemos mais pseudo-classes e introduzimos os pseudo-elementos.**

Exercício prático com links

* **Foram criados** 3 links**:**
* **Repositório no GitHub**
* **Canal do YouTube**
* **Site do Curso em Vídeo**
* **Esses links foram usados para aplicar diferentes** pseudo-classes**:**
* **🔸 Pseudo-classes aprendidas:**
* **a:visited → Estiliza links já visitados.**
* **a:hover → Quando o mouse passa por cima do link.**
* **a:active → Quando o link é clicado (ativo).**

Exemplo prático:

* **a {**
* **color: red;**
* **text-decoration: none;**
* **font-weight: bold;**
* **}**
* **a:visited {**
* **color: darkred;**
* **}**
* **a:hover {**
* **color: black;**
* **text-decoration: underline;**
* **}**
* **a:active {**
* **color: yellow;**
* **}**

**Estilizando conteúdos especiais com classes:**

* **Foi criada uma** classe .especial **para links destacados.**
* **Essa classe recebe estilos próprios e também** pseudo-elementos **(::before e ::after).**

Exemplo:

* **a.especial {**
* **background-color: darkred;**
* **color: white;**
* **}**
* **a.especial::before {**
* **content: "🔗 ";**
* **}**
* **a.especial::after {**
* **content: " ↩";**
* **}**

Pseudo-elementos

* **Utilizam** dois-pontos duplos ::**.**
* **Servem para inserir** conteúdo adicional via CSS**, sem alterar o HTML diretamente.**
* **Exemplo:**
* **a::after {**
* **content: " (link)";**
* **color: gray;**
* **}**

Símbolos importantes aprendidos

* **#id → seletor por** identificador
* **.classe → seletor por** classe
* **: →** pseudo-classe **(ex: :hover, :visited, :active)**
* **:: →** pseudo-elemento **(ex: ::before, ::after)**
* **> → filho direto de um elemento (ex: div > p)**

**Conclusão**

* **Agora você sabe usar e combinar id, class, pseudo-classes e pseudo-elementos.**
* **Aprendeu a aplicar efeitos visuais com hover, active, visited e a enriquecer o conteúdo com ::before e ::after.**
* **No próximo capítulo, o foco será em** box model**: tamanhos, margens, bordas e espaçamentos.**

**AULA 17 –**

**O que é o modelo de caixas?**

Todo **elemento visível em HTML** é considerado uma **caixa (box)**.

Essas caixas são usadas para organizar o layout do site.

A organização dessas caixas segue o chamado **box model**, ou modelo de caixas.

**Partes da anatomia de uma caixa**:

**Conteúdo**: o texto ou imagem visível.

**Width (largura)** e **Height (altura)**:

width: define a **largura** da caixa.

height: define a **altura** da caixa.

**Border (borda)**:

É uma linha que contorna o conteúdo da caixa.

Fica imediatamente colada ao conteúdo, por padrão.

**Padding (preenchimento ou acolchoamento)**:

Espaço entre o **conteúdo** e a **borda**.

Afasta o conteúdo da borda internamente.

**Margin (margem)**:

Espaço **externo** da borda.

Separa a caixa de outros elementos ao redor.

**Outline (contorno)**:

Traçado externo à margem (sem ocupar espaço).

Usado para destaque visual, mas não afeta o layout.

**Tipos de caixas**

**Box-level (caixas em bloco)**:

Ocupam **100% da largura** disponível.

Sempre começam em uma **nova linha**.

Exemplos: div, h1 a h6, p, header, section, form, footer.

**Inline-level (caixas em linha)**:

Ocupam **apenas o espaço necessário** para o conteúdo.

**Não quebram linha**.

Permitem outros elementos ao lado.

Exemplos: span, a, strong, em, input, select, label, button.

**Dicas importantes**

O **modelo de caixas é fundamental** para controlar a posição, tamanho e espaçamento dos elementos.

Saber diferenciar **box-level** de **inline-level** é essencial para criar layouts responsivos e bem organizados.

O **VS Code ajuda na digitação** de propriedades como width, height, padding, margin, border, etc.

Atenção à ortografia de termos em inglês como height (altura) e width (largura), que confundem b**astante.**

**Conclusão**

A aula foi **100% teórica**, focada em **entender o conceito visual e estrutural** das caixas no CSS.

No próximo vídeo, o professor começará a parte **prática**, construindo um site de exemplo para aplicar o Box Model.

**AULA 18 –**

## Resumo – Aula Prática: Modelo de Caixas (Parte 1)

### Objetivo da aula:

Demonstrar na prática como funcionam os conceitos do box model, especialmente:

* Caixas box-level e inline-level
* Propriedades: width, height, padding, border
* Ferramenta: DevTools do Chrome para visualização e ajustes em tempo real

### Criação do exercício

1. Criada a pasta ex021
2. Criado o arquivo caixa01.html
3. Elementos usados no HTML:
   * <h1>: exemplo de box-level
   * <p>: também box-level
   * <a>: exemplo de inline-level

### Diferença prática entre Box-level e Inline-level

* Box-level (ex: <h1>, <p>):
  + Ocupa 100% da largura
  + Começa sempre em nova linha
* Inline-level (ex: <a>, <span>):
  + Ocupa apenas o espaço do conteúdo
  + Pode ficar ao lado de outros elementos

### Uso do DevTools (Ferramentas do desenvolvedor do Chrome)

* Acessado via botão direito > inspecionar
* Utilizado para:
  + Visualizar o box model do elemento
  + Ver propriedades aplicadas (user-agent)
  + Testar alterações de estilo sem modificar o código original
  + Analisar visualmente margin, border, padding, content

### Estilização no CSS

Exemplo aplicado ao <h1>:

h1 {

width: 300px;

height: 300px;

background-color: lightgray;

border-width: 10px;

border-style: solid;

border-color: darkslategray;

}

Exemplo aplicado ao <a>:

a {

border-width: 10px;

border-style: solid;

border-color: red;

}

### Aplicação de padding (preenchimento interno)

h1 {

padding-top: 10px;

padding-right: 10px;

padding-bottom: 10px;

padding-left: 10px;

}

* Ordem correta: top, right, bottom, left (sentido horário)
* Pode ser testado em tempo real usando o DevTools, aumentando e diminuindo com a rodinha do mouse

### Dicas práticas com DevTools

* Alterar propriedades com a rodinha do mouse
* Desligar temporariamente estilos para testes
* Copiar estilos ajustados no DevTools e colar no CSS original

### Próximo conteúdo (próxima aula):

* Veremos:
  + margin (margem externa)
  + outline (contorno)
  + Simplificações de código CSS
  + Continuação com o mesmo exercício

**AULA 19 –**

## Resumo – Capítulo 16: Modelo de Caixas (Parte 3)

### Objetivo da aula:

Dar continuidade ao exercício prático do box model, aprofundando:

* margin, outline, padding, border
* Shorthands (atalhos) para simplificar o CSS
* Compreensão visual com **DevTools**
* Centralização de elementos
* Manipulação de display

### Revisando os conceitos anteriores:

* Vimos height, width, padding, border aplicados a <h1> e <a>
* Diferença entre elementos **box-level** (<h1>, <p>) e **inline-level** (<a>, <span>)

### Novas propriedades abordadas:

#### margin – Espaço externo da caixa

* Criada com margin-top, margin-right, margin-bottom, margin-left
* Ordem correta (sentido horário): **top, right, bottom, left**
* Pode ser usada para **afastar elementos entre si**
* Também permite **centralizar horizontalmente** com:
* margin: 0 auto;

#### outline – Contorno externo (visual, não afeta layout)

* Usado para destacar caixas **fora da borda**, dentro da margin
* Propriedades:
* outline-width: 5px;
* outline-style: dashed;
* outline-color: salmon;
* Pode ser simplificado com shorthand:
* outline: 5px dashed salmon;

### Shorthands (atalhos de CSS):

#### ✔️ Para border:

border: 10px solid darkslategray;

(Substitui border-width, border-style, border-color)

#### ✔️ Para padding ou margin:

* **4 valores**: top, right, bottom, left
* padding: 10px 20px 30px 40px;
* **2 valores**: top & bottom, right & left
* margin: 20px 10px;
* **1 valor**: todos os lados iguais
* padding: 10px;

### 🧭 Centralização horizontal com margin

* Para centralizar uma **caixa com largura definida**:
* margin-left: auto;
* margin-right: auto;

ou simplesmente:

margin: 0 auto;

### 📏 Mudando tipo de caixa com display

* Padrões:
  + <h1>: display: block (ocupa largura total)
  + <a>: display: inline (só ocupa o necessário)
* Pode ser alterado:
  + display: block; para tornar um inline em bloco
  + display: inline-block; para misturar comportamentos (inline com tamanho configurável)

### 🧠 Resumo da anatomia de uma caixa:

+--------------------------+

| margin (externo) |

| +--------------------+ |

| | outline | |

| | +--------------+ | |

| | | border | | |

| | | +----------+ | | |

| | | | padding | | | |

| | | | content | | | |

| | | +----------+ | | |

| | +--------------+ | |

| +--------------------+ |

+--------------------------+

### 📌 Dicas finais do professor:

* **Volte na teoria** se algo ficou confuso
* **Use o DevTools** para testar as propriedades e visualizar em tempo real
* **Pratique bastante**: CSS é domínio visual e técnico
* Não tenha pressa: **aprender programação é uma jornada**

**AULA 20 –**

## Resumo – Capítulo 16: Estruturação com caixas semânticas

### Objetivo da aula:

Mostrar **como aplicar o modelo de caixas** para **construir um site real**, usando **tags semânticas** para dar organização e significado ao conteúdo da página.

### 🧱 Antes (HTML4): Tudo era <div>

Antigamente a estrutura de um site era feita com:

<div id="header"></div>

<div id="menu"></div>

<div id="content"></div>

<div id="footer"></div>

Ou seja, tudo era **div** com id para diferenciar as seções.

### ✅ Agora (HTML5): Tags Semânticas

Com HTML5, usamos **tags com significado próprio**, que melhoram:

* A **acessibilidade**
* O **SEO** (indexação por mecanismos de busca)
* A **organização visual e lógica** do código

### 🧩 Tags semânticas principais:

| Tag | Função |
| --- | --- |
| <header> | Cabeçalho da página ou seção |
| <nav> | Navegação (menus/links) |
| <main> | Conteúdo principal da página |
| <section> | Seção temática do conteúdo |
| <article> | Conteúdo independente |
| <aside> | Conteúdo lateral/periférico |
| <footer> | Rodapé |

### 💻 Exemplo de estrutura de site:

<body>

<header>

<h1>Meu site</h1>

<nav>

<a href="#">Link 1</a>

<a href="#">Link 2</a>

</nav>

</header>

<main>

<section id="noticias">

<article>

<h2>Notícia sobre o Brasil</h2>

<p>Conteúdo da notícia...</p>

<aside>Escrito por José da Silva</aside>

</article>

<article>

<h2>Outra notícia</h2>

<p>Texto...</p>

</article>

</section>

</main>

<footer>

<p>Desenvolvido pelo Curso em Vídeo</p>

</footer>

</body>

### 🎨 CSS para estilizar a estrutura:

Exemplo de estilos aplicados:

body {

font-family: Arial, sans-serif;

background-color: #eeeeee;

margin: 0;

}

header, main, footer {

background-color: white;

margin: 10px;

padding: 10px;

}

footer {

background-color: black;

color: white;

text-align: center;

padding: 5px;

}

article {

background-color: lightgray;

padding: 5px;

margin-bottom: 10px;

}

aside {

background-color: gray;

color: white;

padding: 5px;

}

### 🔄 Tags semânticas ≠ estilos prontos

Essas tags **não mudam a aparência por padrão**. Você precisa aplicar o **CSS** para definir cores, tamanhos e posições.

### ⚙️ Exemplos de personalizações feitas na aula:

* Alteração de font-family no <body>
* Uso de padding e margin para espaçamentos
* Cores de fundo para destacar as caixas
* Centralização com text-align: center
* Separação de links do <nav> com margin-right
* Estilização de links com text-decoration e color
* Interatividade com :hover (sublinhado ao passar o mouse)

### 🧠 Conclusões da aula:

* Agora você sabe **como montar um site** com estrutura real
* **Não use só** div, use **tags semânticas**
* **Organize o conteúdo antes de pensar no visual**
* Use DevTools para ajustar detalhes e testar
* Pratique montando sua própria estrutura!

📌 Dica final do professor Guanabara:

"Não copie meu layout. Use os exemplos como referência, mas crie **seu próprio site** com sua lógica e estilo. O importante é **entender a estrutura e praticar**!"

**AULA 21 –**

**Resumo – Capítulo 16: Sombras com box-shadow**

Objetivo da aula:

Aprender a usar a propriedade box-shadow para adicionar sombras externas e internas aos elementos, deixando o site mais moderno e com profundidade visual — sem exageros!

1. O que é box-shadow?

É a propriedade CSS usada para adicionar **sombras a caixas (blocos)**.

📌 Sintaxe básica (short-hand):

box-shadow: deslocamentoX deslocamentoY espalhamento embaçamento cor;

✅ Exemplo:

box-shadow: 2px 2px 7px 1px rgba(0, 0, 0, 0.3);

2. Significado dos valores

Ordem dos parâmetros:

deslocamento horizontal → (positivo = direita, negativo = esquerda)

deslocamento vertical → (positivo = baixo, negativo = cima)

blur-radius → embaçamento da sombra (opcional)

spread-radius → quanto a sombra se expande (opcional)

cor → geralmente rgba(0, 0, 0, 0.x) para transparência

🛠️ Exemplo completo:

box-shadow: 5px 5px 15px 2px rgba(0, 0, 0, 0.2);

3. Como criar sombras na prática

Você pode:

Escrever o código manualmente, ou

Usar o **DevTools (Google Chrome)** para configurar visualmente:

Clique no elemento com botão direito → **Inspecionar**

Vá em **Styles** → Clique em + → Selecione box-shadow

Use o **editor gráfico** de sombra que aparece (drag-and-drop)

Copie o resultado gerado e cole no CSS

4. Dicas importantes sobre sombras

Use sombras **sutis** e com **transparência**.

Não use cores sólidas como red ou blue para sombra (fica artificial).

Prefira tons de **preto com opacidade**:

rgba(0, 0, 0, 0.2)

Uma sombra deve ser percebida, não gritar.

Use inset para **sombras internas**:

box-shadow: inset 0px 0px 5px rgba(0, 0, 0, 0.3);

5. Exemplo aplicado na aula:

nav {

background-color: #333;

padding: 10px;

box-shadow: 2px 2px 7px rgba(0, 0, 0, 0.3);

}

6. Corrigindo espaçamentos e rodapés com DevTools

O professor também mostrou como identificar e resolver um problema de **margem automática do parágrafo dentro do** <footer>:

footer p {

margin: 0;

padding: 4px;

}

Isso garantiu que o espaço visível fosse **apenas o** padding **do footer** e não uma margem automática do parágrafo.

7. Atalho: Copiando do DevTools

Após criar a sombra graficamente no DevTools:

Copie o valor de box-shadow

Cole diretamente no seu código CSS

Resumo visual da ordem dos valores do box-shadow:

box-shadow: X Y Blur Spread Color;

Exemplo:

box-shadow: 3px 3px 6px 0px rgba(0, 0, 0, 0.2);

Ou com sombra interna:

box-shadow: inset 3px 3px 6px rgba(0, 0, 0, 0.2);

Conceitos que você aprendeu na aula:

O que é box-shadow

Como aplicar sombras externas e internas

Como usar DevTools para criar sombras visualmente

Como evitar sombras exageradas

Como configurar transparência na sombra

Como resolver conflitos de margem e padding no footer

**AULA 22 –**

Resumo – Capítulo 16: Vértices Arredondados com border-radius

🎯 Objetivo da aula:

Aprender a arredondar bordas de caixas (divs, seções, menus etc.) usando border-radius, além de combinar essa técnica com sombras (box-shadow) para criar efeitos visuais elegantes e discretos.

🔄 1. O que é border-radius?

A propriedade border-radius serve para **arredondar os cantos** de uma borda de qualquer elemento em HTML.

🧾 Exemplo básico:

box {

border-radius: 20px;

}

Isso aplica 20 pixels de raio em todos os cantos da caixa.

📐 2. Formas de aplicar border-radius

▶️ Um valor só:

Aplica o mesmo raio para os 4 cantos.

border-radius: 10px;

▶️ Dois valores:

Aplica raio alternado:

border-radius: 10px 20px;

Interpretação:

1º valor: superior esquerdo + inferior direito

2º valor: superior direito + inferior esquerdo

▶️ Quatro valores:

Aplica um valor diferente para cada canto no sentido horário:

border-radius: 10px 20px 30px 40px;

Ordem:

canto superior esquerdo

canto superior direito

canto inferior direito

canto inferior esquerdo

🎨 3. Usando porcentagem para criar círculos

Criando uma bola perfeita:

div {

width: 100px;

height: 100px;

border-radius: 50%;

}

Quando altura e largura são iguais e o border-radius é 50%, forma-se um círculo perfeito.

🧪 4. Testando com o DevTools (Chrome)

No painel **"Elements" → "Styles"**, é possível:

Usar controles visuais para alterar cada vértice separadamente

Ver o efeito em tempo real

Ajustar interativamente valores como top-left, bottom-right etc.

🔄 5. Combinando border-radius com box-shadow

Você pode aplicar os dois juntos:

box {

border-radius: 10px;

box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.2);

}

🧠 Dica de design:

Use **arredondamento e sombra discretos**

Menos é mais → bordas suaves e sombras com leve transparência (ex: rgba(0,0,0,0.2))

🧱 6. Exemplo prático (com sombra discreta):

main {

border-radius: 10px;

box-shadow: 2px 2px 7px rgba(0, 0, 0, 0.2);

}

🔘 7. Exemplo de bola com div:

<div id="bola"></div>

#bola {

width: 100px;

height: 100px;

background-color: white;

border-radius: 50%;

margin: 10px auto;

}

📌 8. Conclusões importantes

border-radius pode ser escrito com **1, 2 ou 4 valores**

Pode usar **porcentagens**, como 50%, para efeitos circulares

Pode aplicar sombras junto com bordas arredondadas

**Use com moderação**: design bonito é design discreto

Com o **DevTools** você pode ajustar visualmente todos os cantos

AULA 23 –

Aqui está o **resumo completo da aula final do Capítulo 16** do curso de HTML e CSS do Curso em Vídeo, focada em **bordas decoradas personalizadas com CSS**, utilizando imagem.

✅ Resumo – Bordas decoradas com CSS (border-image)

🎯 Objetivo da aula:

Ensinar como criar e aplicar **bordas personalizadas** usando **imagens PNG** com border-image, explicando também o uso do programa **GIMP** para desenhar as bordas.

🖌️ 1. Por que usar bordas decoradas?

O professor **não gosta muito do recurso**, mas mostra como usá-lo caso **você precise num projeto**.

Não é recomendado para uso excessivo ou design moderno, mas pode ter aplicação em projetos específicos ou criativos.

🛠️ 2. Criando a imagem da borda no GIMP

Criar imagem de **80x80px**, com **fundo transparente**.

Desenhar os **vértices (cantos)** da borda com bolinhas ou formas.

Usar **régua do GIMP** para dividir a imagem em 3 partes (guia de terços).

Pintar os cantos (superior esquerdo, superior direito, inferior direito, inferior esquerdo).

Exportar a imagem em .png.

💾 3. Inserindo a borda no CSS

p {

border: 20px solid black; /\* obrigatório \*/

border-image-source: url('borda.png');

border-image-slice: 27;

}

border precisa existir, mesmo que com cor transparent.

border-image-source: define a imagem da borda.

border-image-slice: fatiamento da imagem. O número ideal depende do design da imagem.

Pode ser ajustado com **DevTools** para visualização.

🧩 4. Ajuste fino com DevTools

O professor usa o **Inspecionar Elemento (DevTools)** para:

Ajustar valores de slice.

Testar efeitos visuais.

Ver como os cortes afetam os cantos e lados da borda.

🧾 5. Simplificação com shorthand

Você pode usar a shorthand border-image:

border-image: url('borda.png') 27 repeat;

Substitui as três linhas (source, slice, repeat).

A propriedade repeat pode ser stretch, repeat ou round.

⚠️ 6. Cuidados importantes

Sempre defina border (espessura e tipo) antes de usar border-image.

Bordas com imagem **não são compatíveis com todos os navegadores**.

Use transparent na borda de fallback:

border: 20px solid transparent;

🧠 7. Conclusão

Recurso útil em **casos especiais** (ex: bordas temáticas, infantis, personalizadas).

Pouco usado em sites modernos, mas pode ter valor estético em projetos criativos.

A prática com GIMP + HTML + CSS é importante para entender como fatiar corretamente a imagem.

**AULA 24 –**

**Resumo –** Capítulo 16: Bordas Decoradas (Curso em Vídeo - HTML e CSS)

1. **Introdução**
   * O professor menciona que não gosta muito de bordas decoradas, mas reconhece que podem ser úteis em projetos específicos.
2. **Criação de Borda Personalizada com GIMP**
   * Ensina a criar uma imagem pequena (80x80 px) com fundo transparente.
   * Mostra como desenhar os cantos da borda com formas circulares usando ferramentas de seleção e preenchimento.
   * Exporta a imagem como PNG.
3. **Implementação no HTML/CSS**
   * Cria um arquivo HTML simples com um elemento <p> com borda.
   * Utiliza a propriedade border-image-source apontando para a imagem criada.
   * Usa border-image-slice para fatiar a imagem e posicioná-la corretamente nas bordas do elemento.
   * Ajusta os valores com DevTools para calibrar o fatiamento da imagem.
4. **Dicas Adicionais**
   * Pode-se usar border-image-repeat: stretch | round | repeat.
   * Utiliza shorthand com border-image para simplificar o código.
   * Recomenda manter a propriedade border com valor transparente para compatibilidade e fallback.

**AULA 25 –**

## Resumo – Desafio 10 (Capítulo 16)

**Curso em Vídeo – HTML e CSS com Gustavo Guanabara**

### 🧪 O que será feito no desafio?

* Criação de um **site completo** com:
  + Cabeçalho (header)
  + Menu de navegação
  + Conteúdo com **listas**, **vídeos**, **links**, **imagens**
  + Layout **responsivo** (adapta-se ao tamanho da tela)
* Exemplo mostrado: site com imagem da criadora do Android, vídeo do YouTube adaptável, menus funcionais e elementos de conteúdo estruturado.

### 📁 Onde encontrar o desafio?

1. Acesse: [gustavoguanabara.github.io](https://gustavoguanabara.github.io/)
2. Vá em: **Curso de HTML e CSS**
3. Clique em: **Resolva os desafios**
4. Escolha o **Desafio 10**
5. Baixe o **pacote ZIP** com arquivos, imagens e textos.

### 🧠 Dica do professor

* **Tente fazer o desafio sozinho** antes de assistir à explicação.
* Se travar, não tem problema: o **Capítulo 17** trará o **passo a passo completo** para a construção do projeto.
* A ideia é **te ensinar a pensar como desenvolvedor**, não apenas copiar código.

**MODULO 3 –**

**AULA 1 –**