**Principais tags CSS**

**Cores e Fundos**

* background-color: Define a cor de fundo do elemento.
* color: Define a cor do texto.

**Fontes e Textos**

* font-family: Define a família da fonte (Ex: "Times New Roman", Arial).
* font-size: Define o tamanho da fonte.
* font-style: Define o estilo da fonte (Ex: italic, normal).
* font-weight: Define o peso (espessura) da fonte (normal, bold, lighter, bolder ou valores numéricos de 100 a 900).
* text-transform: Transforma o texto (uppercase, lowercase, capitalize).
* text-align: Alinha o texto (left, center, right, justify).
* text-indent: Define o espaço no início do parágrafo.
* text-decoration: Adiciona efeitos ao texto (underline, line-through, overline, none).
* letter-spacing: Define o espaçamento entre as letras (positivo para aumentar, negativo para reduzir).
* line-height: Define a altura da linha, controlando o espaçamento vertical entre as linhas de texto.

**Dimensões e Espaçamentos**

* width: Define a largura do elemento.
* height: Define a altura do elemento.
* padding: Define o espaçamento interno do elemento.
* margin: Define o espaçamento externo do elemento.
* margin-top: Define o espaçamento superior de um elemento.

**Bordas e Sombras**

* border: Define todas as propriedades da borda (largura, estilo e cor).
* border-top-width: Define a espessura da borda superior.
* border-right-style: Define o estilo da borda direita.
* border-left-color: Define a cor da borda esquerda.
* border-bottom: Define a borda inferior.
* border-radius: Define bordas arredondadas.
* outline: Adiciona uma borda externa ao elemento.
* box-shadow: Adiciona sombras ao elemento.

**Layout e Posicionamento**

**Flex Container**

* flex-direction: Define a direção principal do layout dos itens do contêiner flexível;
  + row: (Padrão) Organiza os itens na horizontal, da esquerda para a direita.
  + row-reverse: Organiza os itens na horizontal, da direita para a esquerda.
  + column: Organiza os itens na vertical, de cima para baixo.
  + column-reverse: Organiza os itens na vertical, de baixo para cima.
* flex-wrap: Define os itens podem quebrar para a próxima linha quando não couberem no contêiner.
  + nowrap: (Padrão) Todos os itens permanecem em uma única linha.
  + wrap: Os itens quebram para a próxima linha se necessário.
  + wrap-reverse: Os itens quebram para a próxima linha, mas em ordem reversa.
* flex-flow: Atalho para definir flex-direction e flex-wrap ao mesmo tempo.
  + flex-flow: row wrap: Isso significa que os itens ficarão organizados em linha (row) e podem quebrar para uma linha (wrap).
* justify-content: Controla o alinhamento dos itens ao longo do eixo principal.
  + flex-start: (Padrão) Alinha os itens ao início do eixo principal.
  + flex-end: Alinha os itens ao final do eixo principal.
  + center: Centraliza os itens ao longo do eixo principal.
  + space-between: Distribui os itens igualmente, com espaço apenas entre eles.
  + space-aroud: Distribui os itens com espaço ao redor de cada um.
  + space-evenly: Distribui os itens com espaços iguais entre e nas extremidades.
* align-items: Controla o alinhamento dos itens ao longo do eixo perpendicular (vertical, se flex-direction for row).
  + stretch: (Padrão) Os itens esticam para ocupar todo o espaço disponível no eixo secundário.
  + flex-start: Alinha os itens ao início do eixo secundário.
  + flex-end: Alinha os itens ao final do eixo secundário
  + center: Centraliza os itens no eixo secundário.
  + baseline: Alinha os itens de acordo com a linha base do texto dentro deles.
* align-content: Controla o espaçamento entre linhas dentro do contêiner quando há múltiplas linhas (quando flex-wrap: wrap).
  + flex-start: Alinha as linhas ao topo do contêiner.
  + flex-end: Alinha as linhas à parte inferior do contêiner.
  + center: Centraliza as linhas com espaço apenas entre elas.
  + space-between: Distribui as linhas com espaço apenas entre elas.
  + space-aroud: Distribui as linhas com espaço ao redor de cada uma.
  + stretch: (Padrão) As linhas esticam para ocupar o máximo de espaço vertical.
* gap: Define o espaçamento entre os itens dentro do contêiner flexível.
  + gap: 10px: Define um espaço de 10px entre os itens.
  + row-gap: 10px: Define o espaço entre as linhas.
  + column-gap: 10px: Define o espaço entre as colunas.

**Flex Items**

* flex-basis: Define o tamanho inicial do items antes de aplicar o flex-grow ou flex-shrink. funciona como um “tabanho base” do item.

O item terá 200px de largura (se o flex-direction for row) ou 200px de altura (se for column).

* Valor especial: auto (padrão) O tamanho será baseado no contúdo do item.
* flex-grow: Define o quanto um item pode crescer em relação aos outros, se houver espaço extra disponível.
  + Valor 0 (padrão): O item não cresce além do tamanho definido.
  + Valor 1 ou maior: O item pode crescer para ocupar o espaço extra.
* flex-shrink: Define o quanto um item pode encolher caso o espaço seja reduzido.
  + Valor 1 (padrão): O item pode encolher, se necessário.
  + Valor 0: O item não encolhe.
  + Valor maior que 1: O item encolhe mais rapidamente que os outros.
* flex: Atalho para redefinir flex-grow, flex-shrink e flex-basis de uma só vez.
  + Sintaxe:

flex: <grow> <shrink> <basis>

* + Exemplo:

flex: 1 1 200px

* order: Controla a ordem de exibição dos itens, independentemente da order do HTML.
  + O padrão é 0, mas pode ser qualquer número positivo ou negativo.
  + Exemplo:

.item1 { order: 2; }

.item2 { order: -1; }

.item3 { order: 1; }

Ordem final na tela: .item2, .item3, .item1.

* align-self: Permite que um único item tenha um alinhamento diferente dos outros, ignorando align-items do contêiner.
  + auto: Usa o valor de align-items do contêiner (Padrão).
  + flex-start: Alinha o item ao topo do eixo secundário.
  + flex-em;d: Alinha o item à parte inferior do eixo secundário.
  + center: Centraliza o item no eixo secundário.
  + baseline: Alinha o item com a linha base do texto dos outros.
  + stretch: Estica o item para ocupar todo o espaço disponível.
  + Exemplo:

.item3 { align-self: flex-end; }

O .item3 ficará alinhado ao final do eixo secundário, diferente dos outros.

**Listas**

* list-style-type: none: Remove os marcadores de listas (ul e ol).

**Backgroud**

* background-size: Define o tamanho da imagem de fundo.
* background-repeat: Controla a repetição da imagem de fundo.

**Efeitos**

* hover: O hover é um pseudo-seletor usado para aplicar efeitos quando o usuário passa o mouse sobre um elemento.

**Layout e Interatividade**

* pointer-events: Define como um elemento responde às interações do mouse ou toque (auto, none, all, visiblePainted).
* cursor: Permite personalizar o ícone do ponteiro do mouse quando ele passa sobre um elemento (default, pointer, text, wait, help, not-allwed).

**Transform**

* translate (x, y): Move o elemento para a direita e para baixo, sendo negativos para esquerda e direita.
* rotate(45deg): Gira o elemento 45 graus no sentido horário.
* scale(x, y): Redimenciona, aumenta e diminui um elemento em largura e altura. Se não dizer qual eixo e escrever apenas um número, ele redimensiona ambos eixos.
* skewX(30deg): Inclina o elemento 30 graus no eixo horizontal (como se fosse esticado lateralmente).

**Você pode combinar várias transformações em uma única linha:**

transform: translate(50px, 20px) rotate(45deg) scale(1.2) skewX(10deg);

**Centralizar itens**

**\* {**

margin: 0;

padding: 0;

box-sizing: border-box;

**}**

**img / div {**

display: flex;

align-items: center;

justify-content: center;

background-color: black;

height: 100vh;

**}**

**h1, p {**

text-align: center;

**}**

**img2 {**

height: 200px;

display: block;

margin-right: auto;

margin-left: auto;

**}**

**img3 {**

height: 200px;

position: absolute;

top: 50%; / left: 50%;

transform: translateY(-50%); / transform: translateX(-50%);

**}**