

Guia de CSS – Propriedades e Funções

1. Seletores

Definem quais elementos do HTML serão estilizados.

- * → seleciona todos os elementos.
 - element → exemplo: p { } estiliza todos <p>.
 - .classe → estiliza todos com class="classe".
 - #id → estiliza o elemento com id="id".
 - element element → seletor descendente.
 - element > element → seletor filho direto.
 - element + element → próximo irmão.
 - element ~ element → todos os irmãos seguintes.
 - [atributo] → exemplo: input[type="text"].
 - :hover, :focus, :active → pseudo-classes de interação.
 - :nth-child(n), :first-child, :last-child.
 - ::before, ::after → pseudo-elementos.
-

2. Cores e Fundo

- color → cor do texto.
 - background-color → cor de fundo.
 - background-image → imagem de fundo.
 - background-size → cover, contain, valores fixos.
 - background-repeat → repetir (repeat-x, repeat-y, no-repeat).
 - background-position → posição do fundo.
 - background-attachment → fixed, scroll.
 - opacity → transparência.
-

3. Texto

- font-family → fonte usada.
- font-size → tamanho da fonte.
- font-weight → peso (normal, bold, 100-900).
- font-style → italic, normal.
- line-height → altura da linha.

- letter-spacing → espaço entre letras.
 - word-spacing → espaço entre palavras.
 - text-align → alinhamento (left, right, center, justify).
 - text-decoration → none, underline, line-through.
 - text-transform → uppercase, lowercase, capitalize.
 - white-space → quebra ou não de linha.
 - text-shadow → sombra no texto.
-

4. Caixas e Layout

- width, min-width, max-width.
 - height, min-height, max-height.
 - margin → espaçamento externo.
 - padding → espaçamento interno.
 - border → borda (largura, estilo, cor).
 - border-radius → cantos arredondados.
 - box-shadow → sombra externa.
 - overflow → hidden, scroll, auto.
 - display → block, inline, inline-block, flex, grid, none.
 - position → static, relative, absolute, fixed, sticky.
 - top, right, bottom, left → posição.
 - z-index → profundidade de camadas.
-

5. Flexbox

- display: flex → ativa flexbox.
 - flex-direction → row, row-reverse, column, column-reverse.
 - justify-content → flex-start, center, space-between, space-around.
 - align-items → alinhamento no eixo cruzado.
 - align-content → quando há várias linhas.
 - flex-wrap → quebra de linha (wrap, nowrap).
 - flex → define proporções.
 - order → muda a ordem dos elementos.
-

6. Grid Layout

- display: grid.
 - grid-template-columns, grid-template-rows.
 - gap, row-gap, column-gap.
 - grid-column, grid-row.
 - justify-items, align-items.
 - justify-content, align-content.
 - grid-template-areas.
-

7. Animações e Transições

- transition → tempo e propriedades de transição.
 - transition-duration, transition-delay, transition-timing-function.
 - @keyframes → define animações.
 - animation-name.
 - animation-duration.
 - animation-delay.
 - animation-iteration-count → número de vezes.
 - animation-direction → normal, reverse, alternate.
-

8. Transformações

- transform: translate(x,y).
 - transform: rotate(45deg).
 - transform: scale(1.5).
 - transform: skew(20deg).
-

9. Responsividade

- @media (max-width: 768px) → regras para telas menores.
 - vw, vh → porcentagem da tela.
 - em, rem → unidades relativas ao texto.
 - clamp(min, preferido, max) → tamanho fluido.
-

10. Funções úteis

- `calc()` → cálculos (`width: calc(100% - 50px)`).
 - `var()` → usar variáveis CSS (`var(--cor-primaria)`).
 - `rgba()` e `hsla()` → cores com transparência.
 - `url()` → carregar imagens.
 - `min()`, `max()`, `clamp()`.
-

■ Conceitos do CSS – Explicados

1. Seletores

- ◆ Servem para **identificar quais elementos HTML** receberão o estilo.
 - `element` → pega todos elementos do tipo (ex: `p` pega todos parágrafos).
 - `.classe` → pega todos com determinada classe.
 - `#id` → pega o elemento com aquele ID.
 - `:hover`, `:focus` → estilizam quando há interação (ex: mouse em cima).
 - `::before`, `::after` → adicionam conteúdo "falso" antes/depois do elemento.

➡ **Conceito:** Seletores são o **endereço** para o CSS saber **onde aplicar as regras**.

2. Cores e Fundo

- ◆ Controlam **aparência visual** (cores, imagens de fundo).
 - `color` → muda a cor do texto.
 - `background-color` → cor do fundo.
 - `background-image` → insere imagem como fundo.
 - `background-size` → define como a imagem se ajusta.
 - `opacity` → deixa elemento transparente.

➡ **Conceito:** Tudo que define **a cor, textura e transparência** de elementos.

3. Texto

- ◆ Controlam **como o conteúdo textual é exibido**.
 - `font-family` → define a fonte.
 - `font-size` → tamanho do texto.
 - `line-height` → espaçamento entre linhas.

- text-align → alinhamento (esquerda, centro, direita).
- text-transform → transforma maiúsculo/minúsculo.
- text-shadow → aplica sombras no texto.

➡ **Conceito:** Define a **tipografia**, espaçamento e legibilidade.

4. Caixas e Layout

- ◆ Base do **modelo de caixa do CSS** (box model).

Todo elemento é uma caixa com:

- width / height → dimensões.
- margin → espaço **externo** (afasta de outros elementos).
- padding → espaço **interno** (afasta conteúdo da borda).
- border → borda ao redor.
- box-shadow → sombras.
- display → comportamento da caixa (block, inline, flex, etc.).
- position → como a caixa é posicionada na página (absolute, fixed, etc.).

➡ **Conceito:** Controla **espaçamento, tamanho, bordas e posicionamento** de elementos.

5. Flexbox

- ◆ Sistema de **layout moderno** para alinhar elementos em linha ou coluna.
- display: flex → ativa o flexbox.
- flex-direction → direção (linha ou coluna).
- justify-content → alinhamento no eixo principal.
- align-items → alinhamento no eixo cruzado.
- flex-wrap → quebra de linha automática.

➡ **Conceito:** Torna o layout **flexível e adaptável**, ideal para centralizações e alinhamentos.

6. Grid Layout

- ◆ Sistema avançado para **dividir a tela em colunas e linhas**.
- display: grid → ativa grid.
- grid-template-columns → define colunas.
- grid-template-rows → define linhas.

- gap → espaçamento entre itens.
- grid-area → posiciona elementos em áreas específicas.

➡ **Conceito:** Organiza a página em **grades**, facilitando layouts complexos.

7. Animações e Transições

- ◆ Fazem elementos **mudarem suavemente**.
 - transition → define mudanças animadas (tempo, efeito).
 - @keyframes → cria etapas de uma animação.
 - animation-name, animation-duration → ligam a animação ao elemento.

➡ **Conceito:** Dá **movimento** a elementos, deixando a página dinâmica.

8. Transformações

- ◆ Alteram **forma e posição** de elementos.
 - translate → move o elemento.
 - rotate → gira.
 - scale → aumenta/diminui.
 - skew → inclina.

➡ **Conceito:** Muda a **geometria** do elemento no espaço.

9. Responsividade

- ◆ Faz o site se **adaptar a diferentes telas**.
 - @media → aplica regras só em certas resoluções.
 - vw, vh → medidas relativas à tela.
 - em, rem → medidas relativas ao texto.
 - clamp() → define limites de tamanho adaptável.

➡ **Conceito:** Permite que o site seja **acessível em celular, tablet e desktop**.

10. Funções úteis

- ◆ Pequenas ferramentas do CSS.
 - calc() → permite cálculos.
 - var() → usa variáveis CSS (facilita manter padrões).
 - rgba() / hsla() → cores com transparência.

- `min()`, `max()`, `clamp()` → controlam tamanhos flexíveis.

➡ **Conceito:** Tornam o CSS **mais inteligente e reutilizável**.

✚ Resumindo:

- **Seletores** = onde aplicar.
- **Cores/Fundos** = aparência.
- **Texto** = tipografia.
- **Caixas/Layout** = espaçamento e posicionamento.
- **Flex/Grid** = estrutura da página.
- **Animações/Transições** = movimento.
- **Transformações** = mudanças visuais.
- **Responsividade** = adaptação de telas.
- **Funções** = cálculos e variáveis.