

Guia Prático de CSS

Display • Position • Flexbox • Grid

Resumo + comandos essenciais + mapa mental

1. DISPLAY — Como o elemento se comporta

- **block**: ocupa a linha toda, quebra linha, aceita largura e altura.
- **inline**: fica na mesma linha, não aceita width/height.
- **inline-block**: fica em linha, mas aceita tamanho e margens.
- **flex**: ativa o flexbox.
- **none**: esconde o elemento.

Comandos básicos:

```
display: block;  
display: inline;  
display: inline-block;  
display: flex;  
display: none;
```

2. POSITION — Onde o elemento fica

- **static**: padrão.
- **relative**: referência para filhos.
- **absolute**: posicionado dentro de um pai relative.
- **fixed**: preso na tela.
- **sticky**: normal até rolar, depois gruda.

Comandos básicos:

```
position: static;  
position: relative;  
position: absolute;  
position: fixed;  
position: sticky;
```

```
top: 0;  
left: 0;  
right: 0;  
bottom: 0;
```

3. FLEXBOX — Organizar em uma direção

- Organiza elementos em linha ou coluna.
- Facilita centralização.
- Controla alinhamento e espaçamento.

Comandos principais:

```
display: flex;  
flex-direction: row | column;  
justify-content: center | space-between | space-around;  
align-items: center | flex-start | flex-end;  
flex-wrap: wrap;
```

4. GRID — Organizar em linhas e colunas

- Cria layouts completos.
- Trabalha com linhas e colunas.
- Ótimo para páginas e galerias.

Comandos principais:

```
display: grid;  
grid-template-columns: 1fr 1fr 1fr;  
grid-template-rows: auto;  
gap: 10px;  
place-items: center;
```

Mapa Mental — CSS Layout

CSS



■■■ DISPLAY

■ ■■■ block

■ ■■■ inline

■ ■■■ inline-block

■ ■■■ flex

■ ■■■ none



■■■ POSITION

■ ■■■ static

■ ■■■ relative

■ ■■■ absolute

■ ■■■ fixed

■ ■■■ sticky



■■■ FLEXBOX

■ ■■■ display: flex

■ ■■■ flex-direction

■ ■■■ justify-content

■ ■■■ align-items



■■■ GRID

■■■ display: grid

■■■ grid-template-columns

■■■ grid-template-rows

■■■ gap