Display Grid

* **Bidimensional: alinha em colunas e linhas**

display: grid;

**tem que haver um container que envolva os itens que se quer alinhar**

grid-template-columns: 2fr 4fr 2fr;

**define as colunas, e quanto de espaços elas vão ocupar**

**as propriedades justify-content só vão ser aplicadas de o container tiver espaço sobrando, ou seja, quando estiver utilizando fr**

grid-template-columns: auto auto;

**largura automática de acordo com o tamanho do conteúdo**

grid-template-rows: 1fr 1fr;

**espaço ocupado pelas linhas**

grid-row-gap: 20px;

**dá espaçamento entre as linhas**

grid-column-gap: 20px;

**espaçamento entre as colunas**

gap: 20px 40px;

**atalho para os dois. Primeiro row depois column**

**Alinhamentos do container**

justify-content: center;

**alinha os itens na horizontal (apenas se tiver espaço), se não estiver funcionando, verificar o tamanho do container**

align-content: center;

**alinha do conteúdo na vertical (apenas se tiver espaço)**

align-items: flex-start;

**alinhamento na vertical**

place-content: center space-evenly;

**atalho para o align e o justify (nessa ordem), se colocar só uma propriedade aplica as dois**

**Alinhamento dos itens**

grid-column-start: 1;

grid-column-end: 5;

**com 4 frações definidas no grid-template-columns, esse item ocupa toda a tela (na horizontal)**

grid-row-start: 2;

grid-row-end: 4;

**com 3 frações definidas nas rows, o item ocupa todo o espaço (na vertical)**

justify-self: end;

**alinhamento na horizontal (em relação a fração)**

**Grid template áreas**

grid-template-areas:

    'header header header'

    'sidebar content content'

    'footer footer footer';

**No elemento pai**

**Só fazer isso não vai funcionar, tem que colocar isso nos elementos filhos**

grid-area: header;

**nesse exemplo o header vai ocupar toda a tela (na parte de cima), porque foi definido um grid-template-columns de 1fr 1fr 1fr**

grid-area: sidebar;

**o sidebar vai ocupar apenas 1 fração das colunas**

**CSS TRiCKS**