

Grid

O Grid Layout é um sistema de layout bidimensional, ou seja, ele lida com dois eixos, o eixo horizontal e o eixo vertical.

Para adicionar o grid container a um elemento, basta adicionar a propriedade `display: grid`.

Algumas de suas principais propriedades são:

1. **grid-template-columns**: Define o número e o tamanho das colunas do grid, podendo ser definido em pixels, porcentagem, fr, etc.

Ex: `grid-template-columns: 100px 200px 300px;` que define 3 colunas de tamanhos diferentes.

A unidade `fr` segue a mesma lógica do `flex-grow`, onde o espaço disponível é dividido entre as colunas de acordo com o valor de `fr`.

`grid-template-columns: 1fr 2fr 3fr;` define 3 colunas de tamanhos diferentes, onde a primeira coluna ocupará 1/6 do espaço disponível, a segunda 2/6 e a terceira 3/6.

2. **grid-template-rows**: Define o número e o tamanho das linhas do grid, podendo ser definido em pixels, porcentagem, fr, etc.

Ambas as propriedades podem ser definidas de maneira mais dinâmica, utilizando a função `repeat()`. Também podem receber múltiplas unidades de medidas. Ex:

```
grid-template-columns: repeat(2, 50px) 1fr 2fr
```

`100px;` define 5 colunas, onde as duas primeiras possuem 50px, a terceira e quarta 1/3 e 2/3 respectivamente do espaço restante, e a quinta 100px.

3. **grid-template-areas**: Define um template de áreas para o grid, onde cada área é nomeada, e os elementos filhos do grid podem ser posicionados nessas áreas. Ex:

```
grid-template-areas:  
  'header header header'  
  'nav main main'  
  'nav footer footer';
```

Onde o grid terá 3 linhas e 3 colunas, e os elementos filhos podem ser posicionados nas áreas definidas através da propriedade `grid-area`.

[MDN - CSS Grid Layout](#)

[Codepen - Grid teplate areas](#)

Transições e Animações

Transições

As transições são utilizadas para criar efeitos/animações de transição entre dois estados de um elemento, por exemplo, mudar a cor de um botão quando o mouse passa por cima dele.

Ela pode ser aplicada a uma propriedade específica, ou a todas as propriedades. Podemos definir a duração, o tempo de espera, a função de tempo, e o atraso da transição.

```
div {  
  transition: <property> <duration> <timing-function> <delay>;  
}
```

[Documentação MDN - Transições](#)

Animações

As animações são utilizadas para criar efeitos mais complexos, onde é possível definir vários estados intermediários, e também é possível definir a duração, o tempo de espera, a função de tempo, e o atraso da animação.

São utilizadas através da propriedade `@keyframes`, onde definimos algo como uma função, que controla os estados intermediários da animação.

```
@keyframes spin {  
  0% {  
    transform: rotate(0deg);  
  }  
  
  100% {  
    transform: rotate(360deg);  
  }  
}
```



```
div {  
  animation: <duration> <name> <iteration-count> <direction>;  
}
```

Onde a propriedade `animation` recebe a duração, o nome da animação, a quantidade de vezes que a animação será executada, e a direção da animação.

[Documentação MDN - Animações](#)

[Codepen spinner](#)

Media queries

São responsáveis por aplicar estilos diferentes conforme as características do dispositivo que está exibindo a página, por exemplo, a largura da tela.

É um elemento chave para design responsivo, onde o layout da página se adapta conforme o dispositivo que está exibindo a página.

É muito comum utilizar media queries para definir estilos diferentes para dispositivos com telas menores, como smartphones e tablets, baseado na largura da viewport

```
/* Define um estilo para telas menores que 600px */  
@media (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

```
/* Define um estilo para telas maiores que 600px */
@media (min-width: 600px) {
  body {
    background-color: lightblue;
  }
}
```

É possível utilizar várias media queries em conjunto, e também é possível utilizar operadores lógicos como `and`, `or` e `not`.

```
/* Define um estilo para telas maiores que 600px e menores que 800px */
@media (min-width: 600px) and (max-width: 800px) {
  body {
    background-color: lightblue;
  }
}
```

Existem alguns breakpoints comuns que são utilizados para design responsivo, como por exemplo:

```
/* Extra small devices (phones, 600px and down) */
@media (max-width: 600px) {
}

/* Small devices (portrait tablets and large phones, 600px and up) */
@media (min-width: 600px) {
}

/* Medium devices (landscape tablets, 768px and up) */
@media (min-width: 768px) {
}

/* Large devices (laptops/desktops, 992px and up) */
@media (min-width: 992px) {
}

/* Extra large devices (large laptops and desktops, 1200px and up) */
@media (min-width: 1200px) {
}
```

Links úteis sobre media queries

[W3Schools - Media query breakpoints](#)

[MDN - Guia de media queries](#)

[MDN - Usando media queries](#)

