Professor: Giovanni Ribeiro

Youtube: https://www.youtube.com/@pessoadev6982

LinkedIn: https://www.linkedin.com/in/giovannirp/

FRONT-END DESIGN

Conhecendo CSS GRID

O CSS Grid é um sistema de estruturação de layout que o CSS nos fornece. O CSS Grid nos permite configurar layouts em dimensões (linhas e colunas). A junção de linhas e colunas formam uma grade, o que dá o nome a esse sistema (Grid).

Criando a estrutura do **HTML**

```html
<div class="container-grid">
  <img class="img-1" src="http://lorempixel.com.br/200/200/?1" alt="">
  <img class="img-2" src="http://lorempixel.com.br/200/200/?2" alt="">
  <img class="img-3" src="http://lorempixel.com.br/200/200/?3" alt="">
  <img class="img-4" src="http://lorempixel.com.br/200/200/?4" alt="">
  <img class="img-5" src="http://lorempixel.com.br/200/200/?5" alt="">
</div>
```

Resetando espaça mento e criando **100% na imagem**:

```css
* {
  margin: 0;
  padding: 0;
}

img {
  width: 100%;
}
```

Transformando o CSS em Grid:

```css
.grid-container {
  /* transformando em grid */
  display: grid;
}
```

Aqui temos duas colunas

```css
/* declarar quantidade de colunas */
grid-template-columns: 100px 100px;
```

Usando o repeat para fazer a mesma coisa

```css
/* usando o repeat para fazer a mesma coisa */
grid-template-columns: repeat(2, 100px);
```

Alterando para 4 colunas

```css
/* usando o repeat para fazer a mesma coisa */
grid-template-columns: repeat(4, 100px);
```

Porém as imagens não são responsivas, são imagens fixas:

Vamos alterar

```css
/* Alterando para 4 colunas */
grid-template-columns: repeat(4, 20%);
```

O Grid nos permite trabalhar com uma outra unidade de medida que é o **FR fracionária**

Ou seja, FR vai pegar do tamanho cheio da tela

1fr (A unidade fr representa uma fração do espaço disponível no container do grid)

Ele vai se adequando, e deixando todos iguais

Trabalhando com **autofill** ou **autofit**

```
/* colocando regra de tamanho min e max */
/* tamanho min de 150 e tamanho maximo de uma fraçao */
 grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));
```

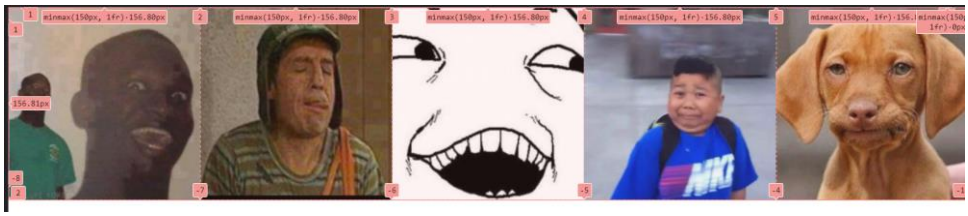
Quando não tem o espaço suficiente auto-fill cria uma coluna a mais

Usando o auto-fit

As imagens vão pegar inteira

A diferença dele, que ele expande as colunas

```
 /* usando o auto-fit */
grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
```



Exercício

Deixar a tela idêntica

Container **máximo** 400px e centralizado.

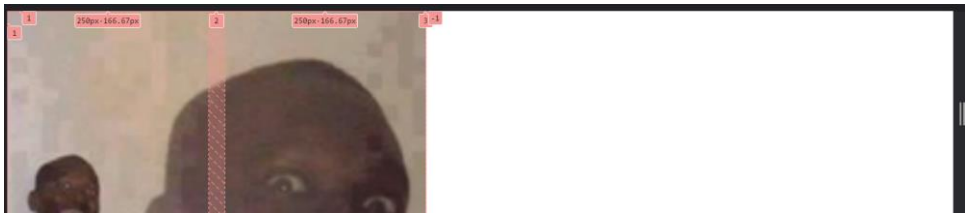
Solução

```css
.grid-container {
  display: grid;
  max-width: 400px;
  margin: 0 auto;
  grid-template-columns: repeat(3, 1fr);
}
```

Usando nova ideia de layout

Aqui estou dizendo que, o elemento .img-1 vai do início até o final

Aqui a grid entende que tem 3 colunas



```css
.grid-container {
    /* usando nova ideia de layout */
    display: grid;
    grid-template-columns: 250px 250px;
    gap: 20px;
}

.img-1 {
  grid-column: 1/3;
}
```

Exercício

Seguindo esse exemplo, deixar img-4 e img-5 pegando tela inteira:

Solução

```css
.img-4 {
  grid-column: 1/3;
}

.img-5 {
  grid-column: 1/3;
}
```

Agora vamos trabalhar com grid área:

Com o grid-areas, conseguimos dar nomes as posições e vincular cada item a posição nomeada;

```
/* grid área */
/* aqui tem 4 linhas */
grid-template-areas:
  "header header" /* os nomes são iguais, por que quero que pega a linha inteira */
  "meio meio2" /* aqui vai ter, duas imagens */
  "meiob meiob2" /* aqui vai ter, duas imagens */
  "baixo baixo" /* os nomes são iguais, por que quero que pega a linha inteira */
;
```

Agora vou chamar grid 1

```
.img-1 {
  grid-area: header;
}
```

A imagem-2 vai ocupar a última linha:

```
.img-2 {
  grid-area: baixo;
}
```

Imagem 3, a gente vai deixar na linha inteira

```
grid-template-areas:
  "header header" /* os nomes são iguais, por que quero que pega a linha inteira */
  "meio meio2" /* aqui vai ter, duas imagens */
  "meiob meiob" /* os nomes são iguais, por que quero que pega a linha inteira */
  "baixo baixo" /* os nomes são iguais, por que quero que pega a linha inteira */
;
```

```
.img-3 {
  grid-area: meiob;
}
```