

01

## Relembrando e projeto simples

### Transcrição

Conseguimos reorganizar todos os elementos do site "Alurinha" utilizando o `flexbox`. Nesta aula vamos relembrar as principais propriedades que aprendemos ao longo do curso!

Primeiro, vamos observar o arquivo `index.html`:

```
<main class="flexContainer">

  <div class="flexItem primeiraRow"></div>
  <div class="flexItem primeiraRow"></div>
  <div class="flexItem primeiraRow"></div>
  <div class="flexItem primeiraRow"></div>

  <!-- <div class="flexItem segundaRow"></div>
  <div class="flexItem segundaRow"></div>
  <div class="flexItem segundaRow"></div>
  <div class="flexItem segundaRow"></div> -->

</main>
```

Nesse código existe a tag `main`, que engloba diversos elementos filhos.

Vamos trabalhar em cima do seguinte projeto:



Para colocar os objetos lado a lado é necessário introduzir na tag `main` um `display:flex`:

```
.flexContainer {
  display: flex;
}
```

Para espacar os itens utilizamos o `justify-content: space-between` que introduz espaço entre os objetos:

```
.flexContainer {
  display: flex;
  justify-content: space-between;
}
```

E teremos:



Lembrando que é possível sempre recorrer a consultas na documentação do **Flexbox** acessando o seguinte [link](https://css-tricks.com/snippets/css/a-guide-to-flexbox/) (<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>). A documentação é um recurso útil para liquidar com dúvidas, por exemplo, nela encontramos os valores que a propriedade `justify-content` recebe:



Para adicionar um espaço no entorno dos itens utilizamos o `justify-content: space-around`:

```
.flexContainer {
  display: flex;
  justify-content: space-around;
}
```

E teremos:



Para alinhar os objetos vamos utilizar o `align-items` e como queremos que eles estejam no centro acrescentamos o valor `center`:

```
.flexContainer {
  display: flex;
  justify-content: space-around;
  align-items: center;
}
```

Ainda, um conceito importante de ser revisto é o `flex-direction`. O `flex-direction` quando adicionado naturalmente distribui os elementos em linhas, mas acrescentando a ele o valor `columns` a colocação dos itens passa a ser ordenada por colunas. Para verificar o funcionamento do `flex` vamos adicionar um comentário no `justify-content` e `align-items` para ignorar estes recursos:

```
.flexContainer {
  display: flex;
  /*justify-content: space-around;
  align-items: center;*/
  flex-direction: column;
}
```

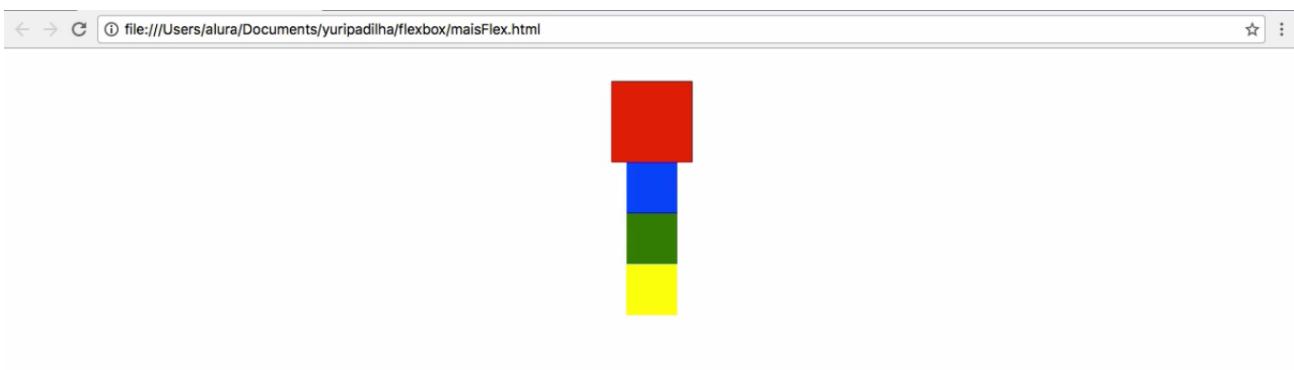
E assim teremos:



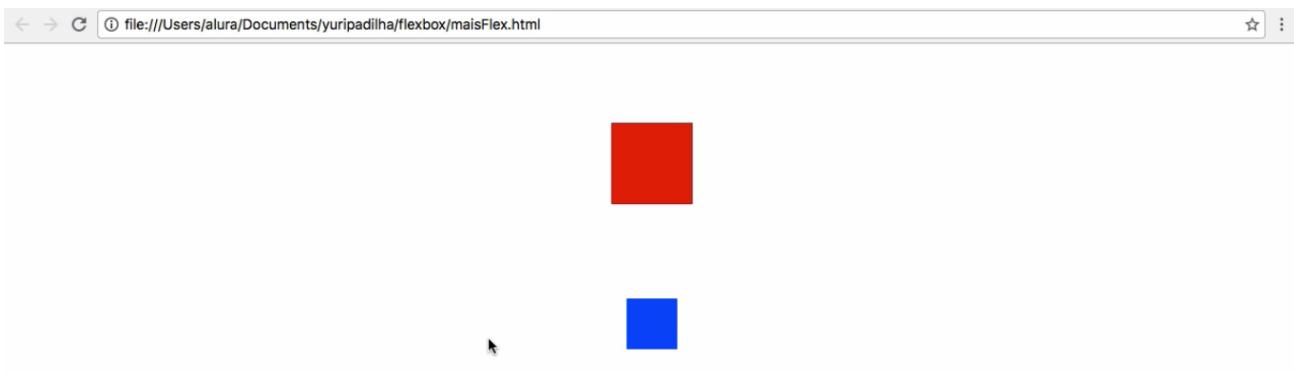
E se deixarmos o comentário apenas no `justify-content`?

```
.flexContainer {
  display: flex;
  /*justify-content: space-around;*/
  align-items: center;
  flex-direction: column;
}
```

Fazendo isso os elementos serão alinhados horizontalmente e no centro:



E se deixarmos todos os itens ativados? O que acontecerá é que os itens ficarão na mesma posição e isso ocorre pois o `flex-container` possui um espaço de mobilidade limitado, portanto, para rearrumar os objetos é preciso aumentar a altura, por exemplo, inserir um `height` de `600px`.



Relembrando as diversas maneiras com as quais podemos utilizar o `flex` chegamos ao fim da revisão!