

DISCIPLINA: PROGRAMAÇÃO DE SISTEMAS WEB

PROFESSOR: CLERIVALDO

ROTEIRO DA ATIVIDADE: FLEBOX CSS

INTRODUÇÃO

O Flexbox é um modelo de layout em CSS que permite criar layouts flexíveis e responsivos. Ele permite que você posicione elementos em um contêiner de forma mais fácil e eficiente do que usando outros modelos de layout em CSS, como o modelo de posicionamento e o modelo de fluxo.

Ele é especialmente útil para criar layouts em que você precisa alinhar e posicionar elementos horizontalmente ou verticalmente, ou para criar layouts em que o tamanho dos elementos deve ser ajustado automaticamente para se adaptar a diferentes tamanhos de tela ou dispositivos.

No Flexbox você trabalha com um contêiner flex (um elemento com `display: flex`) que contém um ou mais itens flex (elementos filhos diretos do contêiner flex). O contêiner flex define as propriedades do layout, enquanto os itens flex se ajustam para se adaptar ao contêiner.

EXEMPLOS DE USO

Aqui estão alguns exemplos de comandos de uso do Flexbox.

Exemplo 1 - Alinhamento horizontal de elementos

```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>

.container {
  display: flex;
  justify-content: center; /* alinhamento horizontal no centro */
}

.item {
  margin: 10px;
  padding: 10px;
  background-color: #ccc;
}
```

DISCIPLINA: PROGRAMAÇÃO DE SISTEMAS WEB

PROFESSOR: CLERIVALDO

ROTEIRO DA ATIVIDADE: FLEBOX CSS

Neste exemplo simples, criamos um contêiner flex com três itens flex. Usamos a propriedade **justify-content** para alinhar os itens horizontalmente no centro.

Exemplo 2 - Alinhamento vertical de elementos

```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
.container {
  display: flex;
  align-items: center; /* alinhamento vertical no centro */
  height: 300px; /* definimos uma altura para o contêiner para que o
alinhamento vertical seja visível */
}

.item {
  margin: 10px;
  padding: 10px;
  background-color: #ccc;
}
```

Neste exemplo, usamos a propriedade **align-items** para alinhar os itens verticalmente no centro. Também definimos uma altura para o contêiner para que o alinhamento vertical seja visível.

Exemplo 3 - Layouts responsivos

```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>

.container {
  display: flex;
  flex-wrap: wrap; /* permite que os itens "quebrem" para a linha
abaixo em telas menores */
  justify-content: space-between; /* alinhamento horizontal dos itens
*/
}
```

DISCIPLINA: PROGRAMAÇÃO DE SISTEMAS WEB

PROFESSOR: CLERIVALDO

ROTEIRO DA ATIVIDADE: FLEBOX CSS

```
.item {  
  flex-basis: 30%; /* define a largura base dos itens */  
  margin: 10px;  
  padding: 10px;  
  background-color: #ccc;  
}
```

Neste exemplo, usamos a propriedade **flex-wrap** para permitir que os itens quebrem para a linha abaixo em telas menores. Usamos a propriedade **justify-content** para alinhar os itens horizontalmente. Também definimos uma largura base

EXERCÍCIOS

Exercício 1 - Alinhar os itens à direita. Alinhe os três itens do contêiner à direita.

```
<div class="container">  
  <div class="item">Item 1</div>  
  <div class="item">Item 2</div>  
  <div class="item">Item 3</div>  
</div>
```

Solução:

```
.container {  
  display: flex;  
  justify-content: flex-end;  
}  
  
.item {  
  margin: 10px;  
  padding: 10px;  
  background-color: #ccc;  
}
```

Neste exemplo, usamos a propriedade **justify-content** no contêiner para alinhar os itens à direita.

DISCIPLINA: PROGRAMAÇÃO DE SISTEMAS WEB

PROFESSOR: CLERIVALDO

ROTEIRO DA ATIVIDADE: FLEBOX CSS

Exercício 2 - Alinhar os itens verticalmente no centro. Alinhe os três itens verticalmente no centro do contêiner.

```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
```

Solução:

```
.container {
  display: flex;
  align-items: center;
  height: 300px;
}

.item {
  margin: 10px;
  padding: 10px;
  background-color: #ccc;
}
```

Neste exemplo, usamos a propriedade `align-items` no contêiner para alinhar os itens verticalmente no centro.

Exercício 3 - Layout responsivo. Crie um layout responsivo com três itens em que cada item tenha uma largura de 33% no desktop e uma largura de 100% em dispositivos móveis.

```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
```

Solução:

```
.container {
  display: flex;
  flex-wrap: wrap;
}

.item {
  flex-basis: 33%;
}
```

DISCIPLINA: PROGRAMAÇÃO DE SISTEMAS WEB

PROFESSOR: CLERIVALDO

ROTEIRO DA ATIVIDADE: FLEBOX CSS

```
margin: 10px;
padding: 10px;
background-color: #ccc;
}

@media screen and (max-width: 480px) {
  .item {
    flex-basis: 100%;
  }
}
```

Neste exemplo, usamos a propriedade flex-wrap no contêiner para permitir que os itens quebrem para a linha abaixo em dispositivos móveis. Usamos a propriedade flex-basis nos itens para definir a largura base. Usamos uma media query para aplicar uma largura de 100% nos itens em dispositivos móveis.

Exercício 4 - Alinhamento horizontal e vertical. Crie um layout com um item centralizado horizontalmente e verticalmente no contêiner e dois itens alinhados verticalmente na parte inferior.

```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
```

Solução:

```
.container {
  display: flex;
  justify-content: center;
  align-items: center;
  flex-direction: column;
  height: 300px;
}

.item {
  margin: 10px;
  padding: 10px;
  background-color: #ccc;
}

.item:nth-child(2) {
  margin-top: auto;
}
```