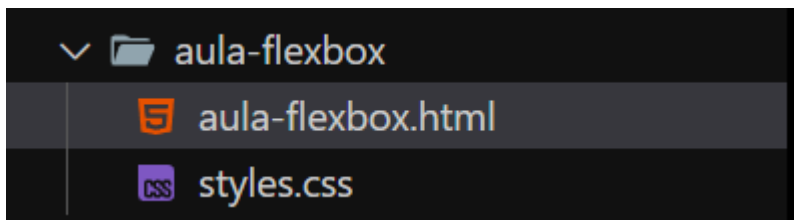


Exemplo Prático FlexBox

Prof.: Caio Malheiros

1 - Crie uma pasta chamada “aula-flexbox” dentro da pasta “HTML e CSS”, em seguida crie a seguinte estrutura abaixo:



2 - Abra o arquivo **aula-flexbox.html** e codifique conforme abaixo:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Layout Flexbox Completo</title>
</head>
<body>
  <div class="container">
    <div class="item">Item 01</div>
    <div class="item">Item 03</div>
    <div class="item">Item 04</div>
  </div>

</body>
</html>
```

3 - Abra o arquivo styles.css e crie as seguinte propriedades:

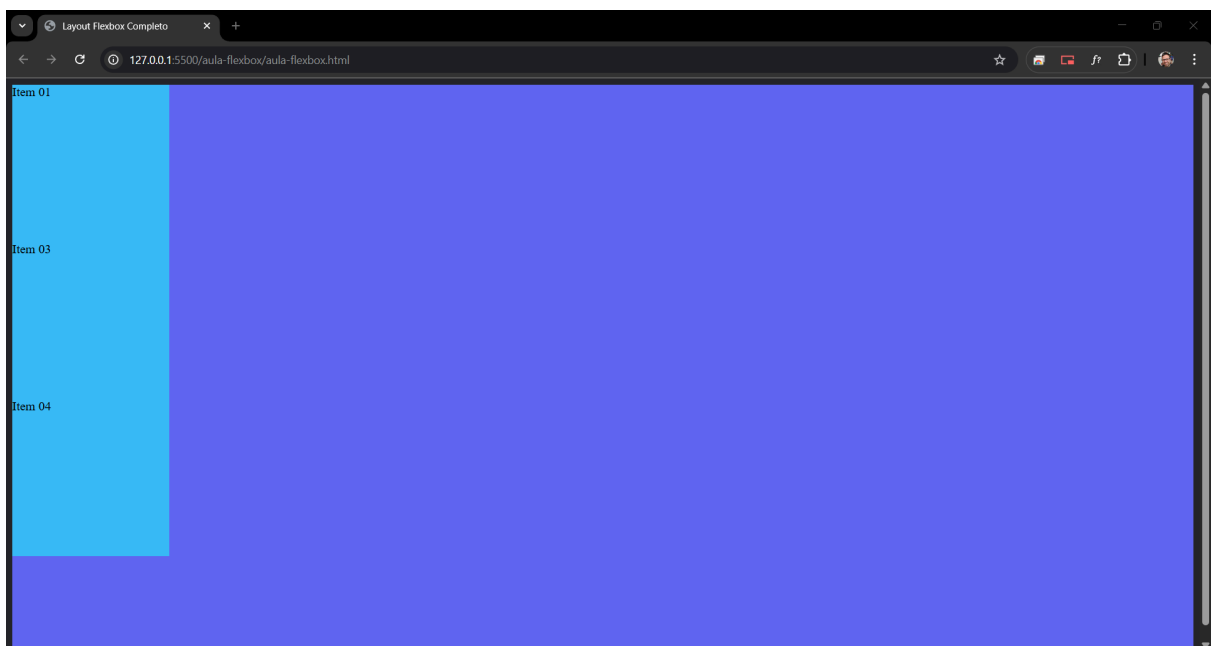
```
body{
  background-color: #27272a;
}

.container {
  height: 100vh;
  background-color: #6366f1;
}

.item {
  width: 200px;
  height: 200px;
  background-color: #38bdf8;
}
```

4 - Salve todos os arquivos e abra o arquivo aula-flexbox.html com o live server:

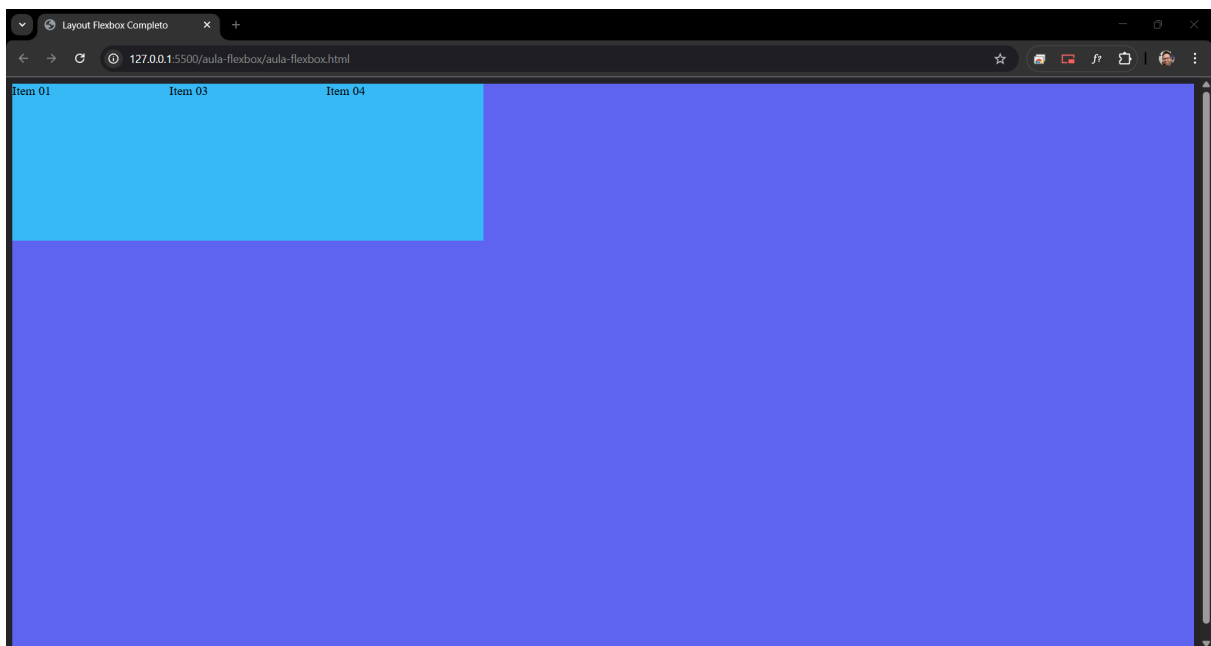
Note que os elementos apareceram um abaixo do outro.



5 - Na classe do container, adicione a propriedade **display:flex** para tornar a div flexível:

```
.container {  
  height: 100vh;  
  background-color: #6366f1;  
  
  /* Adicionando a propriedade flexbox */  
  display: flex;  
}
```

Observe que automaticamente os elementos ficaram um ao lado do outro na horizontal:



6 - Agora vamos alterar os tipos de alinhamentos dos elementos filhos do container, primeiramente vamos alterar o alinhamento dos itens na horizontal. Para isso vamos utilizar a propriedade **justify-content**.

Essa propriedade pode ter um dos seguintes valores:

center: Centraliza os itens no contêiner.

flex-start: Alinha os itens no início do contêiner.

flex-end: Alinha os itens no final do contêiner.

space-around: Distribui os itens com espaço igual ao redor de cada um.

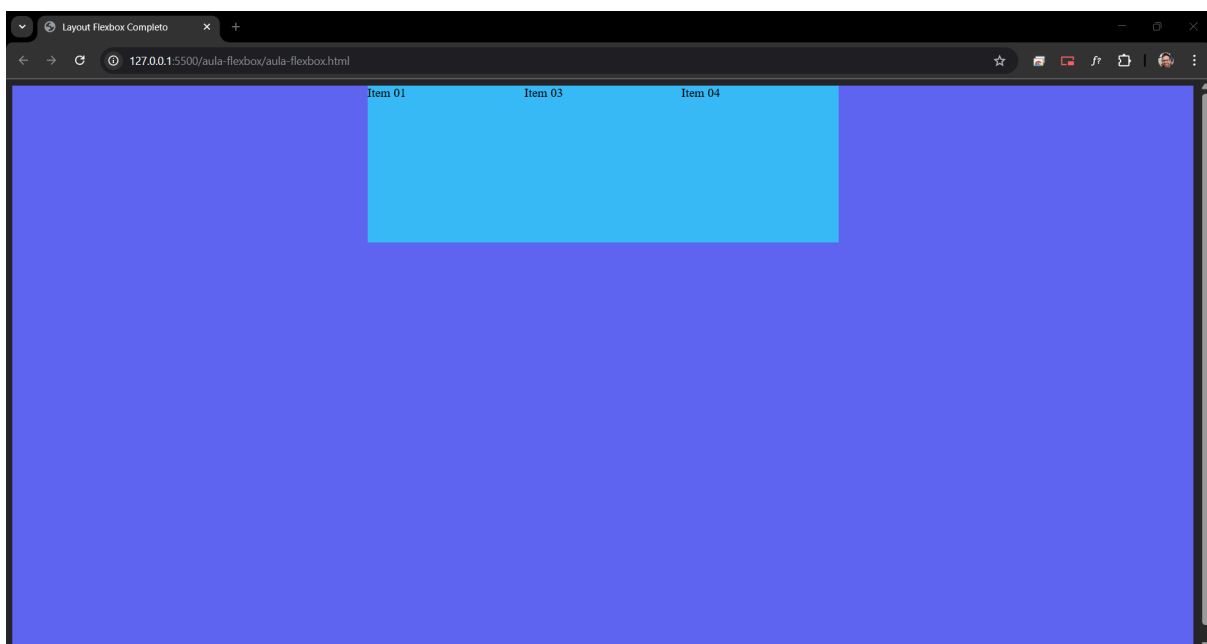
space-between: Distribui os itens de forma que o primeiro e o último fiquem nas extremidades e o espaço restante seja dividido igualmente entre eles.

space-evenly: Distribui os itens com espaçamento uniforme entre eles e as bordas do

Teste a propriedade center, space-around e por fim space-between:

```
.container {  
  height: 100vh;  
  background-color: #6366f1;  
  
  /* Adicionando a propriedade flexbox */  
  display: flex;  
  justify-content: center; /* Alinhamento horizontal */  
}
```

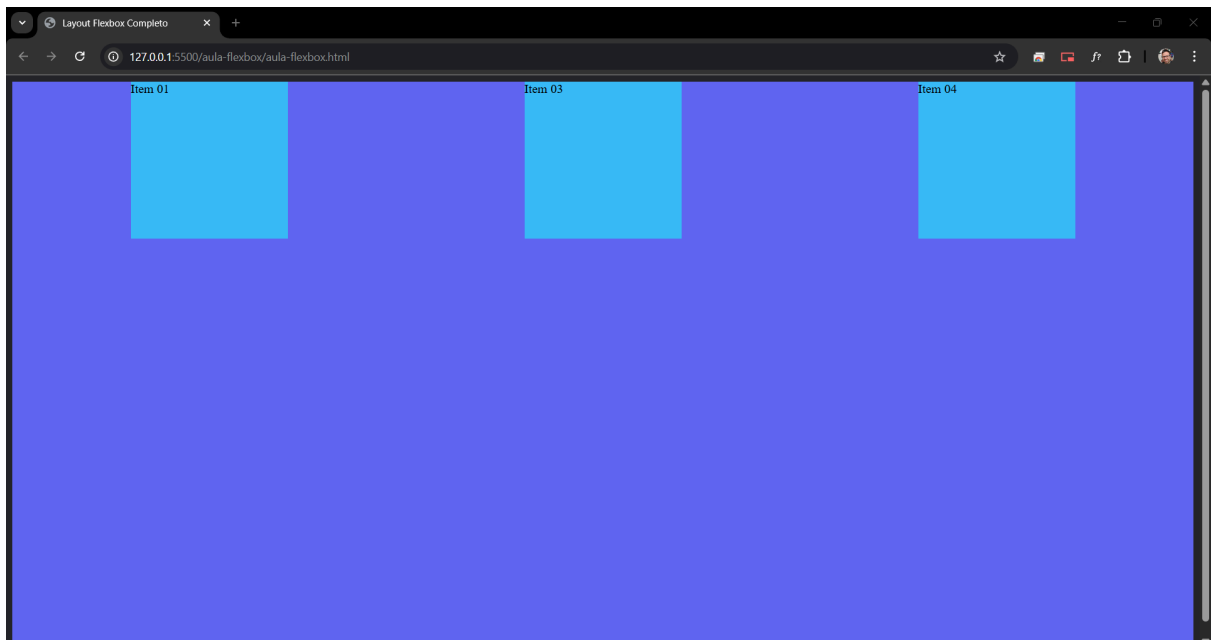
Resultado:



`justify-content: space-around;`

```
.container {  
  height: 100vh;  
  background-color: #6366f1;  
  
  /* Adicionando a propriedade flexbox */  
  display: flex;  
  /* justify-content: center; Alinhamento horizontal */  
  justify-content: space-around; /* Espaçamento entre os elementos */  
}
```

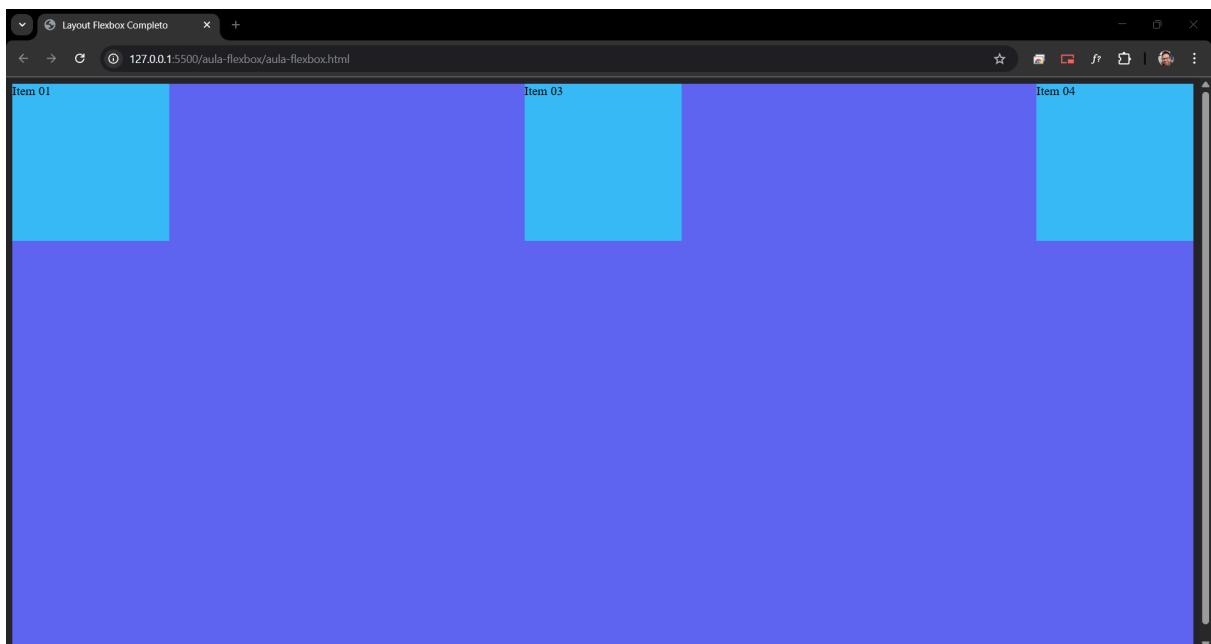
Resultado:



justify-content: space-between

```
.container {  
  height: 100vh;  
  background-color: #6366f1;  
  
  /* Adicionando a propriedade flexbox */  
  display: flex;  
  /* justify-content: center; Alinhamento horizontal */  
  /* justify-content: space-around; Espaçamento entre os elementos */  
  justify-content: space-between; /* Espaçamento entre os elementos, mas sem espaço nas extremidades */  
}
```

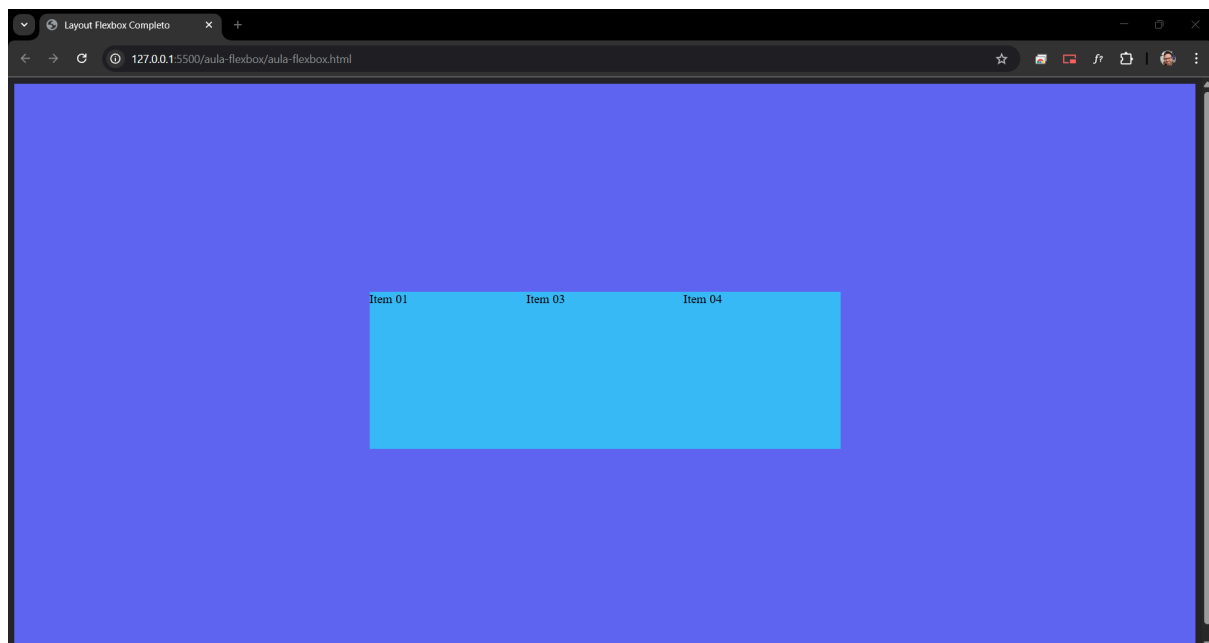
Resultado:



7 - Agora vamos alterar os tipos de alinhamentos dos elementos filhos do container, primeiramente vamos alterar o alinhamento dos itens na **vertical**. Para isso vamos utilizar a propriedade **align-items**.

```
.container {  
  height: 100vh;  
  background-color: #6366f1;  
  
  /* Adicionando a propriedade flexbox */  
  display: flex;  
  /* justify-content: center; Alinhamento horizontal */  
  /* justify-content: space-around; Espaçamento entre os elementos */  
  justify-content: center; /* Espaçamento entre os elementos, mas sem espaço nas extremidades */  
  
  align-items: center; /* Alinhamento vertical */  
}
```

Resultado esperado:



Teste as outras opções abaixo:

center: Centraliza os itens verticalmente, posicionando-os no meio do contêiner.

flex-start: Alinha os itens no início do contêiner (no topo, se a direção for de esquerda para direita).

flex-end: Alinha os itens no final do contêiner (na parte inferior, se a direção for de esquerda para direita).

stretch: Estica os itens para que ocupem toda a altura do contêiner, preenchendo o espaço disponível.

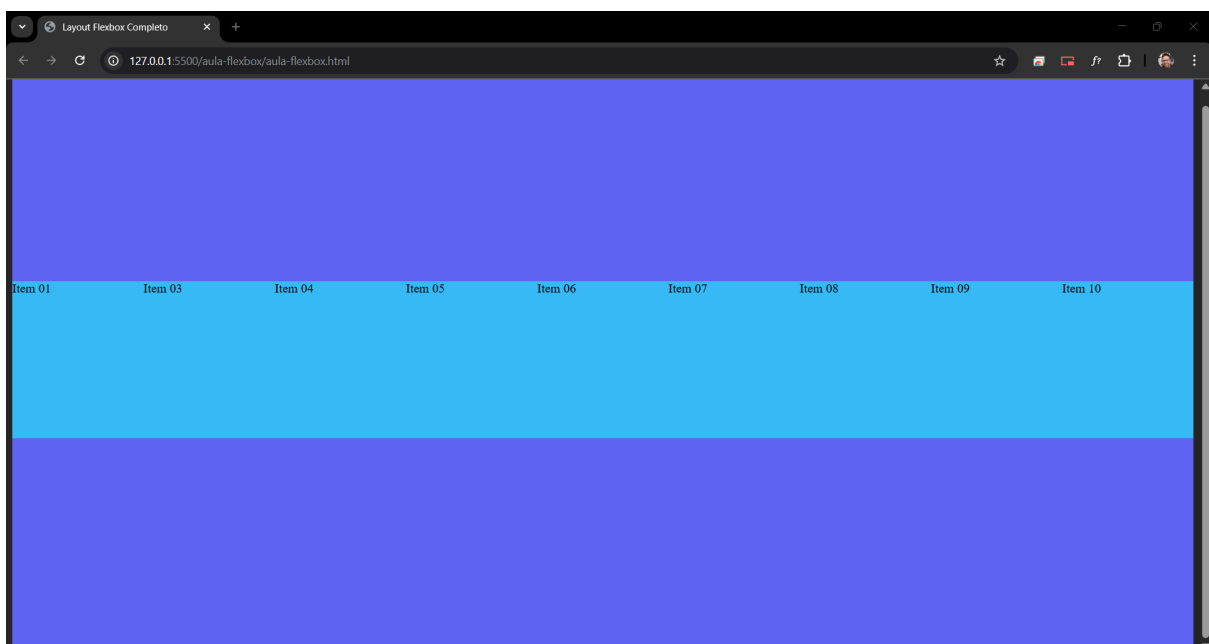
baseline: Alinha os itens com base na linha de base de seu conteúdo textual, garantindo que o texto em diferentes itens fique perfeitamente alinhado.

normal: Reverte o alinhamento para o seu estado padrão, que geralmente é **stretch**.

8 - Adicione mais alguns itens dentro do container para testarmos a propriedade **flex-wrap**:

```
<div class="container">
  <div class="item">Item 01</div>
  <div class="item">Item 03</div>
  <div class="item">Item 04</div>
  <div class="item">Item 05</div>
  <div class="item">Item 06</div>
  <div class="item">Item 07</div>
  <div class="item">Item 08</div>
  <div class="item">Item 09</div>
  <div class="item">Item 10</div>
</div>
```

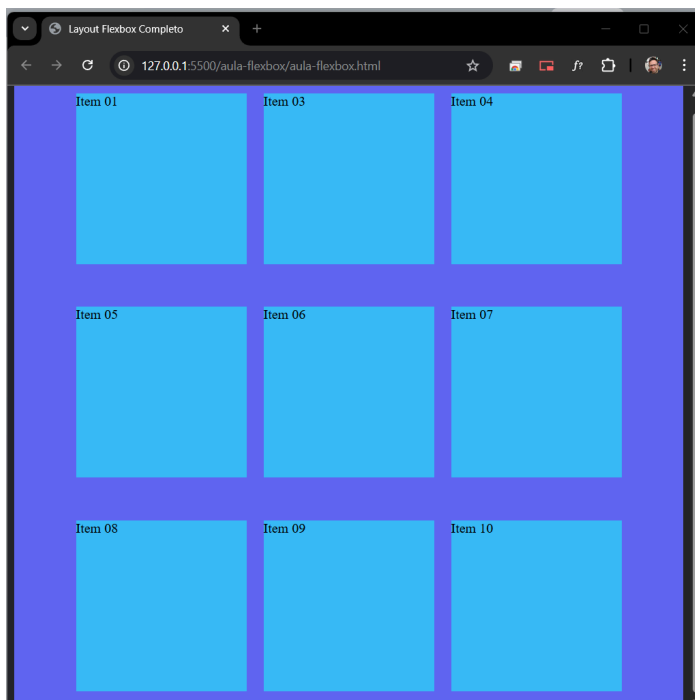
Resultado esperado antes de aplicar o flex-wrap:



Agora aplique a propriedade flex-wrap no container, e utilize o gap para determinar o espaçamento entre os itens (filhos)

```
.container {  
  height: 100vh;  
  background-color: #6366f1;  
  
  /* Adicionando a propriedade flexbox */  
  display: flex;  
  /* justify-content: center; Alinhamento horizontal */  
  /* justify-content: space-around; Espaçamento entre os elementos */  
  justify-content: center; /* Espaçamento entre os elementos, mas sem espaço nas extremidades */  
  
  align-items: center; /* Alinhamento vertical */  
  flex-wrap: wrap; /* Permite que os itens quebrem a linha */  
  gap: 20px; /* Espaçamento entre os itens */  
}
```

Salve todos os arquivos, e verifique o resultado no navegador com a janela reduzida:



9 - Abra o arquivo aula-flexbox.html e adicione as tags abaixo:

```
<body>
<div class="container">
  <div class="item">Item 01</div>
  <div class="item">Item 03</div>
  <div class="item">Item 04</div>
  <div class="item">Item 05</div>
  <div class="item">Item 06</div>
  <div class="item">Item 07</div>
  <div class="item">Item 08</div>
  <div class="item">Item 09</div>
  <div class="item">Item 10</div>
</div>

<div class="container">
  <div class="box" id="item-1">Item 08</div>
  <div class="box" id="item-2">Item 09</div>
  <div class="box" id="item-3">Item 10</div>
</div>
```

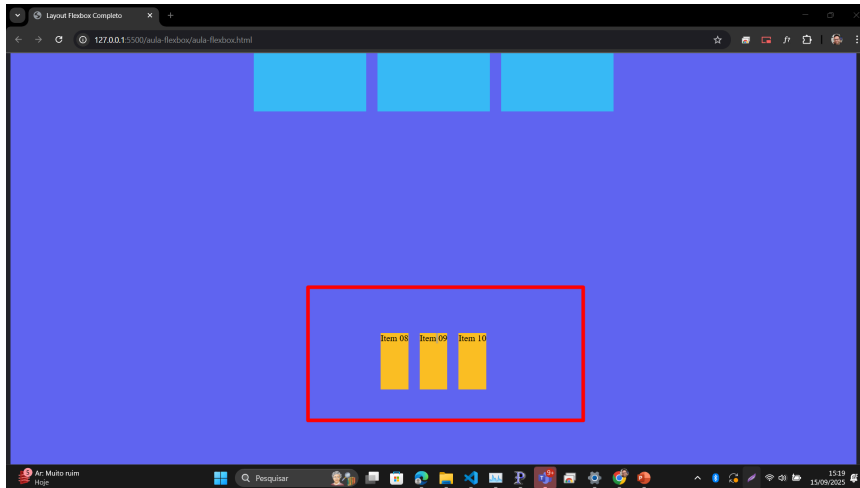
10 - Agora abra o arquivo styles.css e adicione as seguintes regras de css:

```
.item {
  width: 200px;
  height: 200px;
  background-color: #38bdf8;
}

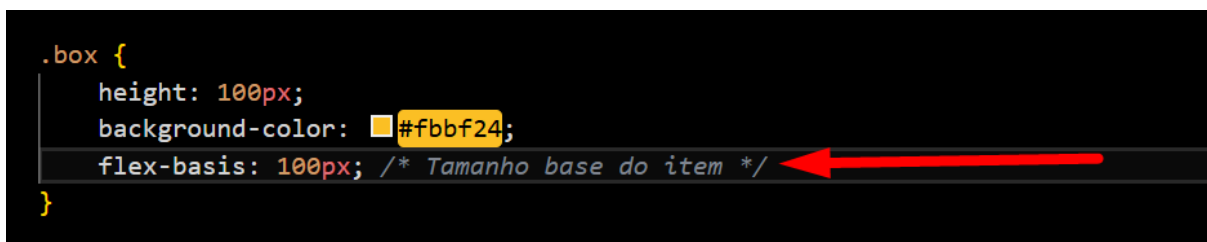
.box {
  height: 100px;
  background-color: #fbbf24;
}
```

Salve todos os arquivos e verifique o resultado esperado:

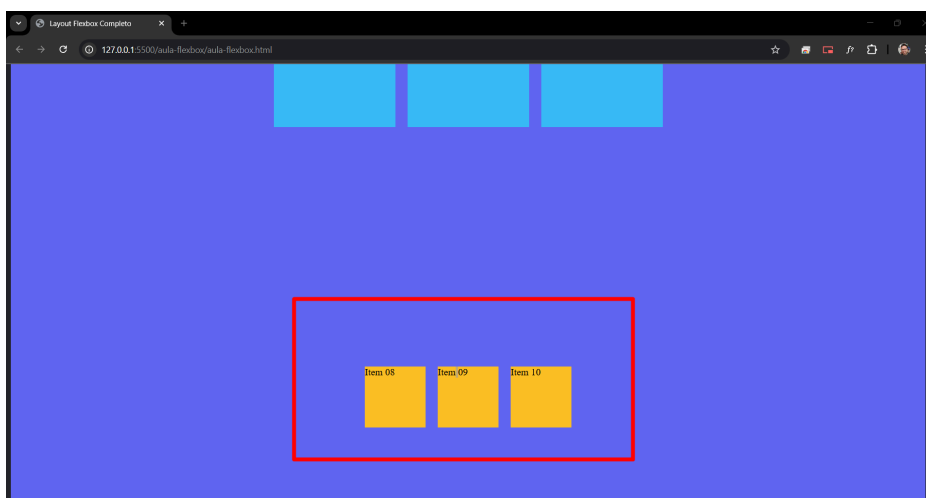
Perceba que como não definimos o tamanho da largura, como padrão o tamanho será o suficiente para exibir o conteúdo de dentro da div.



11 - Agora vamos atribuir a propriedade flex-basis, para definir um tamanho padrão para as div. Abra o arquivo styles.css e adicione a propriedade abaixo:



Salve e verifique o resultado abaixo:



12 - Atividade prática - Pesquise sobre a propriedade flex-grow e crie um exemplo exemplificando seu uso :)