

Curso Técnico em

INFORMÁTICA PARA INTERNET



INSTITUTO FEDERAL
Ceará

Centro de Referência
em Educação a Distância

Guia de atividades para a Unidade 6



Foto: dusanpetkovic1 / Adobe Stock.

Agenda da Semana

Aula síncrona –

Aula expositiva sobre CSS - Flex e Grid.

Atividades assíncronas –

- Tarefa: CSS - Posicionamento e variáveis - 5h
- Quiz - 1h

Encontro presencial –

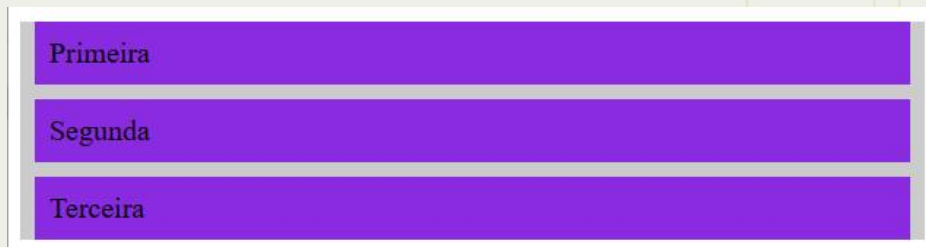
- Discutir os assuntos estudados ao longo da semana abordando as principais dúvidas;
- Acompanhar a execução da atividade prática da semana.

Roteiro

- Display
 - Flex
 - Grid

Sem Flexbox

```
<style>
  .container{
    background-color: #ccc;
  }
  .container div{
    background-color: blueviolet;
    margin: 10px;
    padding: 10px;
    font-size: 20px;
  }
</style>
</head>
<body>
  <div class="container">
    <div>Primeira</div>
    <div>Segunda</div>
    <div>Terceira</div>
  </div>
```



Com FlexBox

```
<style>
  .container{
    display: flex;
    background-color: #ccc;
  }
  .container div{
    background-color: blueviolet;
    margin: 10px;
    padding: 10px;
    font-size: 20px;
  }
</style>
</head>
<body>
  <div class="container">
    <div>Primeira</div>
    <div>Segunda</div>
    <div>Terceira</div>
  </div>
```

Primeira

Segunda

Terceira

flex-direction - padrão

```
<style>
.container{
  display: flex;
  flex-direction: row;
  background-color: #ccc;
}
.container div{
  background-color: blueviolet;
  margin: 10px;
  padding: 10px;
  font-size: 20px;
}
</style>
```

Primeira

Segunda

Terceira

flex-direction: column

```
<style>
  .container{
    display: flex;
    flex-direction: column;
    background-color: #ccc;
  }
  .container div{
    background-color: blueviolet;
    margin: 10px;
    padding: 10px;
    font-size: 20px;
  }
</style>
```

Primeira

Segunda

Terceira

row-reverse

```
.container{  
  display: flex;  
  flex-direction: row-reverse;  
  background-color: #ccc;  
}
```



column-reverse

```
.container{  
  display: flex;  
  flex-direction: column-reverse;  
  background-color: #ccc;  
}
```

Terceira

Segunda

Primeira

flex:1

```
<style>
.container{
  display: flex;
  background-color: #ccc;
}
.container div{
  flex:1;
  background-color: blueviolet;
  margin: 10px;
  padding: 10px;
  font-size: 20px;
}
</style>
```

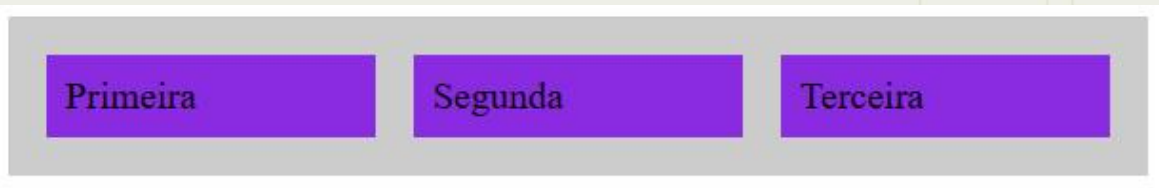
Primeira

Segunda

O flex 1 fará com que os dois elementos se dividam na tela de forma igual.

flex:1

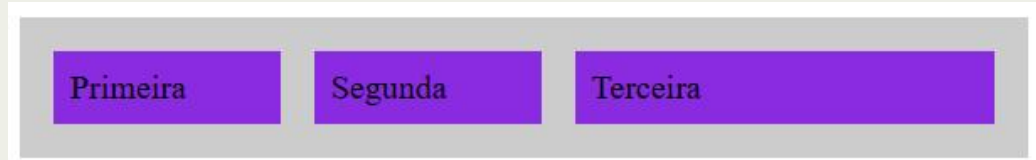
```
<style>
  .container{
    display: flex;
    background-color: #ccc;
  }
  div{
    background-color: blueviolet;
    margin: 10px;
    padding: 10px;
    font-size: 20px;
  }
  .div1{
    flex:1;
  }
  .div2{
    flex:1;
  }
  .div3{
    flex:1;
  }
</style>
</head>
<body>
  <div class="container">
    <div class="div1">Primeira</div>
    <div class="div2">Segunda</div>
    <div class="div3">Terceira</div>
  </div>
</body>
</html>
```



Definição do flex de forma individual

flex:2

```
<style>
  .container{
    display: flex;
    background-color: #ccc;
  }
  div{
    background-color: blueviolet;
    margin: 10px;
    padding: 10px;
    font-size: 20px;
  }
  .div1{
    flex:1;
  }
  .div2{
    flex:1;
  }
  .div3{
    flex:2;
  }
</style>
</head>
<body>
  <div class="container">
    <div class="div1">Primeira</div>
    <div class="div2">Segunda</div>
    <div class="div3">Terceira</div>
  </div>
```



A terceira div terá o dobro de tamanho das outras divs.

Width -barra de rolagem

Primeira

Segunda

Terceira

```
<style>
.container{
  display: flex;
  background-color: #ccc;
}
div{
  flex:1;
  width: 1500px;
  background-color: blueviolet;
  margin: 10px;
  padding: 10px;
  font-size: 20px;
}
</style>
```

Perceba que como a largura das divs é grande, vai gerar uma barra de rolagem e não vai quebrar em duas linhas.

flex-wrap

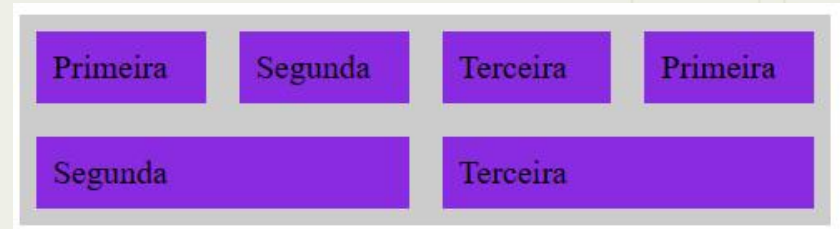
```
.container{  
  display: flex;  
  background-color: #ccc;  
  flex-wrap: wrap;  
}
```



Por padrão, todos os itens flexíveis tentarão caber em uma linha. Você pode alterar isso fazer uma quebra de linha. Isso acontece quando não for possível mostrar todas na mesma linha.

flex-flow: row wrap/ column wrap

```
<style>
.container{
  display: flex;
  background-color: #ccc;
  flex-flow: row wrap;
}
```

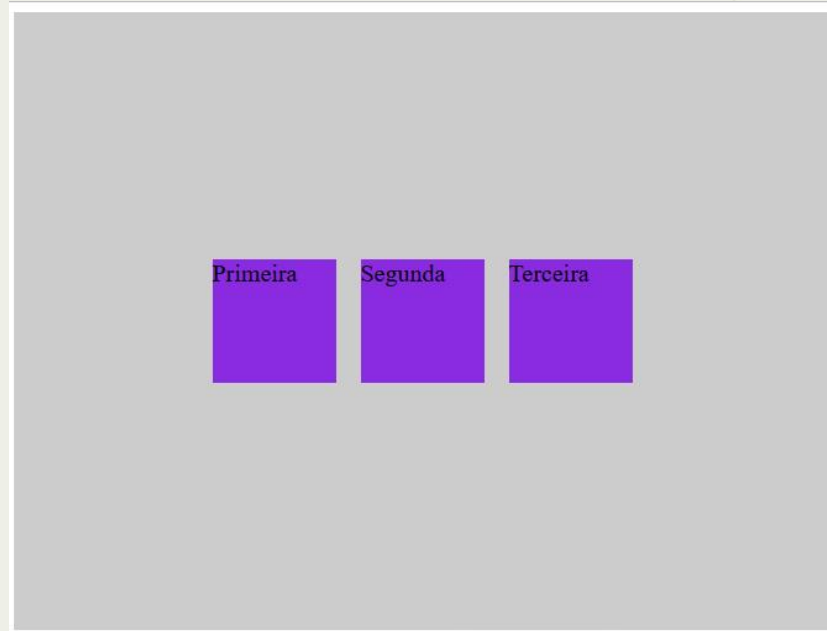


Especifica a direção e a quebra

Alinhamento

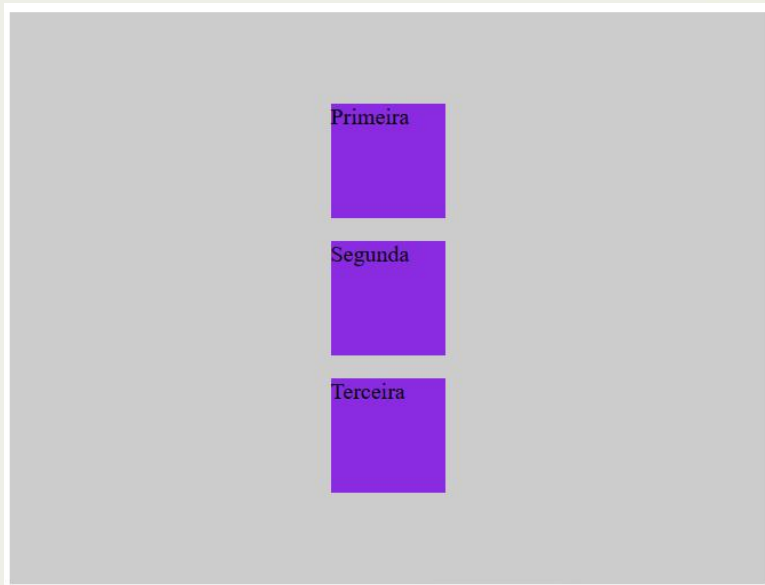
```
.container{
  display: flex;
  background-color: #ccc;
  /*alinha na horizontal*/
  justify-content: center;
  height: 500px;
  /*alinha na vertical*/
  align-items: center ;
}

.container div{
  flex:1;
  background-color: blueviolet;
  margin: 10px;
  padding: 10px;
  font-size: 20px;
  width: 20px;
  height: 20px;
}
```



Alinhamento - Column

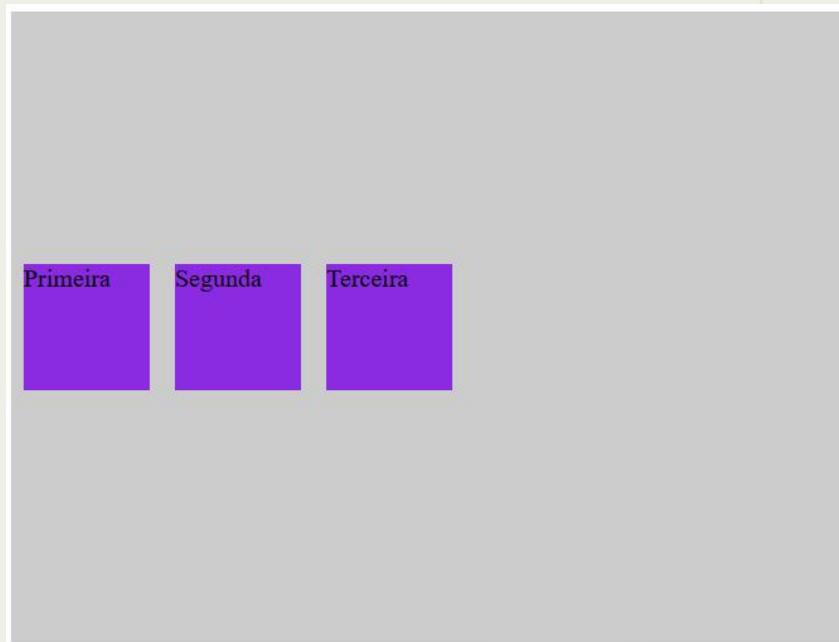
```
<style>
.container{
  display: flex;
  flex-direction: column;
  background-color: #ccc;
  /*alinha na horizontal*/
  justify-content: center;
  height: 500px;
  /*alinha na vertical*/
  align-items: center ;
}
.container div{
  flex:1;
  background-color: blueviolet;
  margin: 10px;
  font-size: 20px;
  width: 100px;
  height: 100px;
}
</style>
```



justify-content: flex-start

```
<style>
.container{
  display: flex;
  flex-direction: row;
  background-color: #ccc;
  /*alinha na horizontal*/
  justify-content: flex-start;
  height: 500px;
  /*alinha na vertical*/
  align-items: center ;
}
.container div{

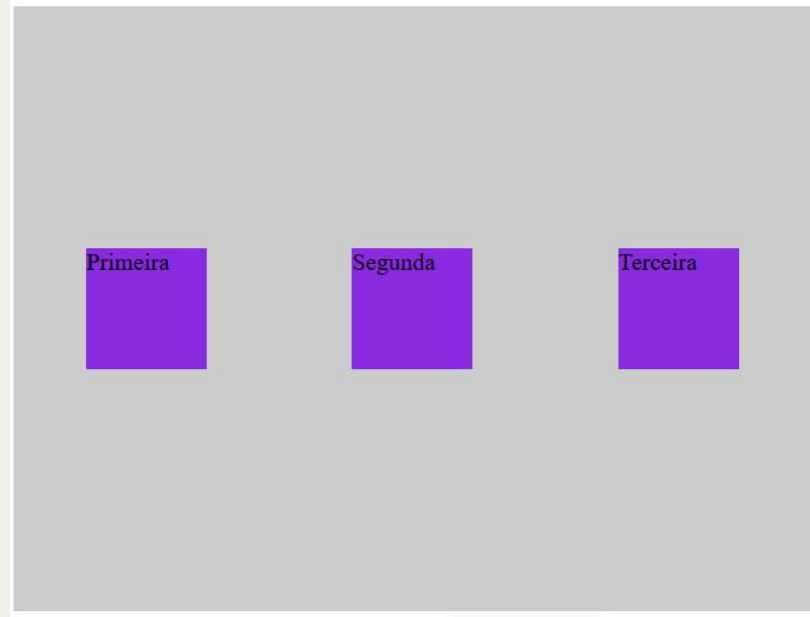
  background-color: blueviolet;
  margin: 10px;
  font-size: 20px;
  width: 100px;
  height: 100px;
}
```



space-around

```
<style>
.container{
  display: flex;
  flex-direction: row;
  background-color: #ccc;
  /*alinha na horizontal*/
  justify-content: space-around;
  height: 500px;
  /*alinha na vertical*/
  align-items: center ;
}
.container div{
  background-color: blueviolet;
  margin: 10px;
  font-size: 20px;
  width: 100px;
  height: 100px;
}
</style>
```

O espaçamento ao redor deles fica o mesmo.



space-between

```
<style>
.container{
  display: flex;
  flex-direction: row;
  background-color: #ccc;
  /*alinha na horizontal*/
  justify-content: space-between;
  height: 500px;
  /*alinha na vertical*/
  align-items: center ;
}
.container div{

  background-color: blueviolet;
  margin: 10px;
  font-size: 20px;
  width: 100px;
  height: 100px;
}
```

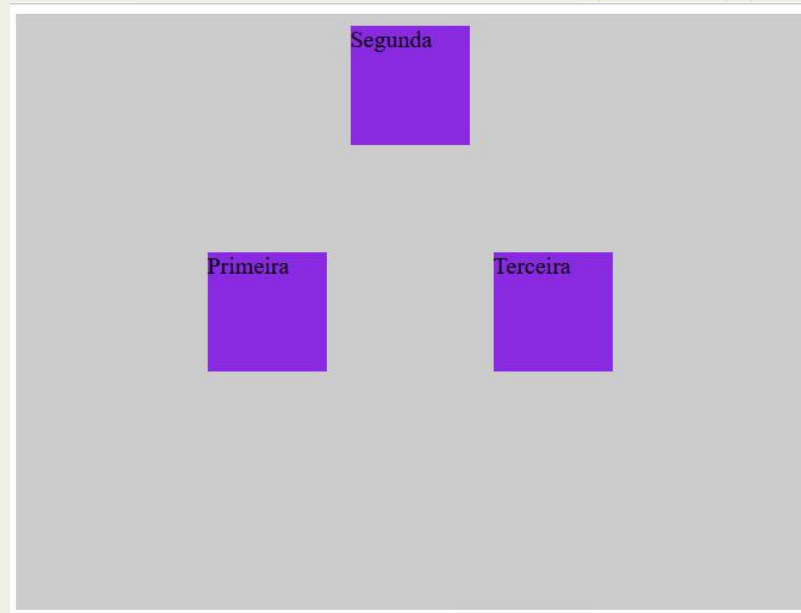
O espaçamento entre eles fica o mesmo.



align-self: flex-start/flex-end/center

Um alinhamento específico para uma div

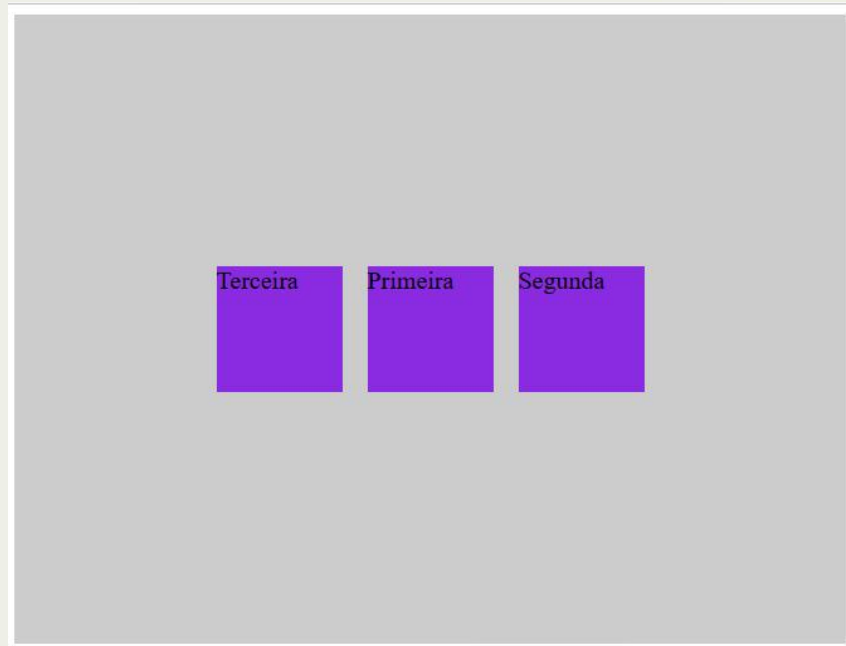
```
<style>
  .container{
    display: flex;
    flex-direction: row;
    background-color: #ccc;
    justify-content: center;
    height: 500px;
    align-items: center ;
  }
  .container div{
    background-color: blueviolet;
    margin: 10px;
    font-size: 20px;
    width: 100px;
    height: 100px;
  }
  .div2{
    align-self: flex-start
  }
</style>
</head>
<body>
  <div class="container">
    <div>Primeira</div>
    <div class="div2">Segunda</div>
    <div>Terceira</div>
  </div>
</body>
```



order

As que não foram especificadas são order 0, por isso a div 3 aparece primeiro

```
    }  
    .div2{  
        order:2;  
    }  
    .div1{  
        order:1;  
    }  
    </style>  
</head>  
<body>  
    <div class="container">  
        <div class="div1">Primeira</div>  
        <div class="div2">Segunda</div>  
        <div>Terceira</div>  
    </div>
```

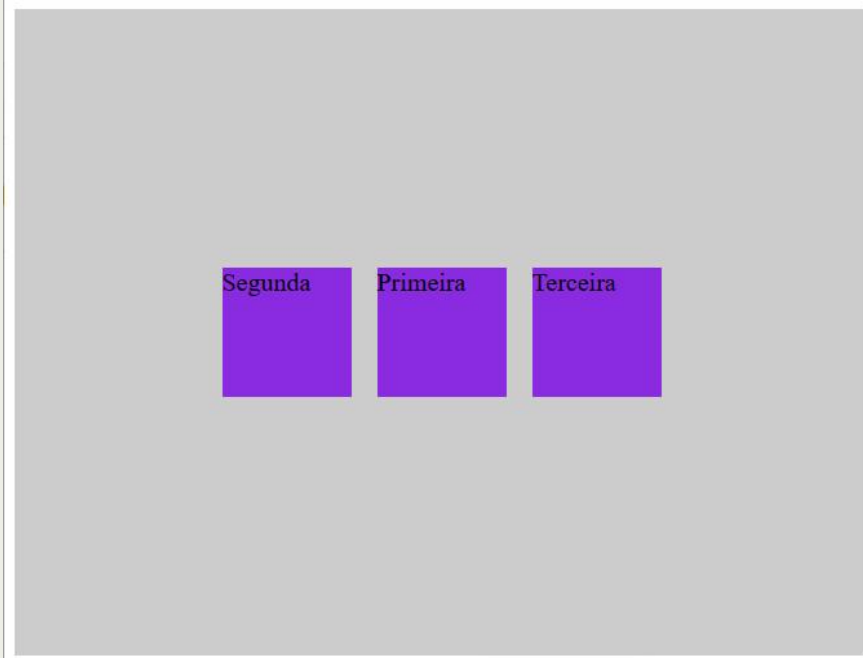


order

Caso queira forçar a posição antes da 0

```
    .div2{
        order:-1;
    }

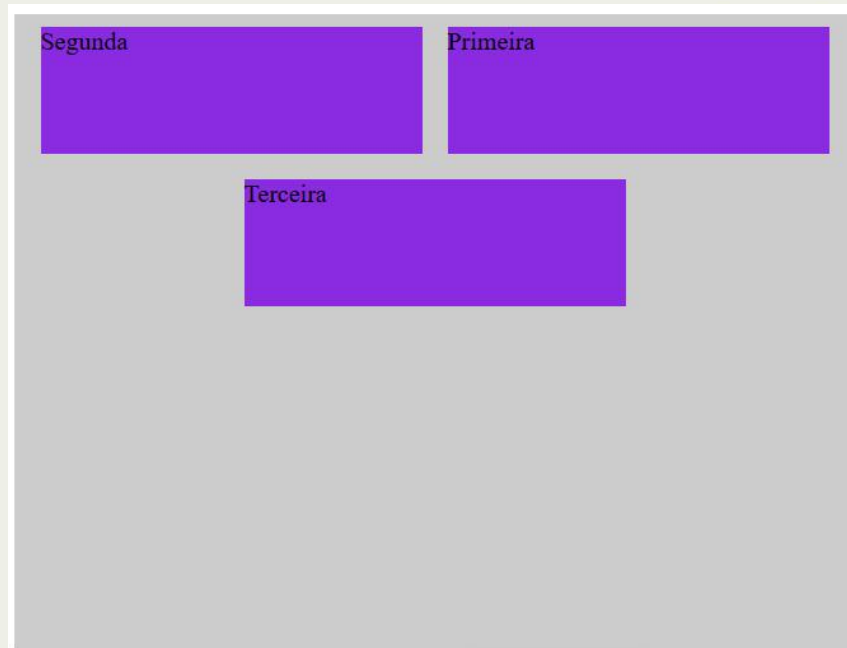
</style>
</head>
<body>
    <div class="container">
        <div>Primeira</div>
        <div class="div2">Segunda</div>
        <div>Terceira</div>
    </div>
</body>
```



align-content: flex-start/flex-end/space-around/space-between

Funciona quando tem um wrap

```
.container{
  display: flex;
  flex-direction: row;
  background-color: #ccc;
  justify-content: center;
  height: 500px;
  align-items: center;
  flex-wrap: wrap;
  align-content: flex-start;
}
.container div{
  background-color: blueviolet;
  margin: 10px;
  font-size: 20px;
  width: 300px;
  height: 100px;
}
```



Grid

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
    <div class="item">5</div>
    <div class="item">6</div>
  </div>
</body>
</html>
```

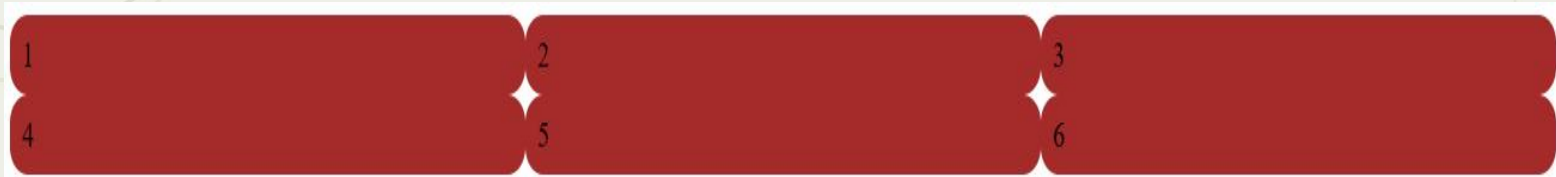
1
2
3
4
5
6

Formatação de todos os itens

```
<style>
  .item{
    background-color: ■ brown;
    padding: 10px;
    border-radius: 15px;
    font-size: 20px;
  }
</style>
</head>
<body>
  <div class="container">
    <div class="item">1</div>
    <div class="item">2</div>
    <div class="item">3</div>
    <div class="item">4</div>
    <div class="item">5</div>
    <div class="item">6</div>
  </div>
</body>
```



grid-template-columns: auto



```
<style>
.container{
  display: grid;
  grid-template-columns: auto auto auto;
}
.item{
  background-color: #brown;
  padding: 10px;
  border-radius: 15px;
  font-size: 20px;
}
</style>
```

Distribui de forma
igual todos os itens
na linha

grid-template-columns



```
<style>
  .container{
    display: grid;
    /*grid-template-columns: auto auto auto;*/
    grid-template-columns: 200px 500px 100px;
  }
  .item{
    background-color: brown;
    padding: 10px;
    border-radius: 15px;
    font-size: 20px;
  }
</style>
```

Especifica tamanho
para cada coluna.

grid-template-columns

1

2

3

4

5

6

```
<style>
  .container{
    display: grid;
    /*grid-template-columns: auto auto auto;*/
    /*grid-template-columns: 200px 500px 100px;*/
    /*grid-template-columns: 1fr 2fr 3fr;*/
    grid-template-columns: repeat(3, 1fr) 200px;
  }
  .item{
    background-color: #brown;
    padding: 10px;
    border-radius: 15px;
    font-size: 20px;
  }
</style>
```

grid-gap

1

2

3

4

5

6

```
<style>
.container{
  display: grid;
  /*grid-template-columns: auto auto auto;*/
  /*grid-template-columns: 200px 500px 100px;*/
  /*grid-template-columns: 1fr 2fr 3fr;*/
  grid-template-columns: repeat(3, 1fr) 200px;
  grid-gap: 10px;
}
.item{
  background-color: #brown;
  padding: 10px;
  border-radius: 15px;
  font-size: 20px;
}
</style>
```

Espaçamento de
linha e coluna entre
as divs.

grid-gap



```
<style>
  .container{
    display: grid;
    grid-template-columns: repeat(3, 1fr) 200px;
    grid-gap: 20px 10px;
  }
  .item{
    background-color: #brown;
    padding: 10px;
    border-radius: 15px;
    font-size: 20px;
  }
</style>
```

Primeiro ele define a coluna de 200px, depois separa o restante do espaço da linha entre três colunas.

grid-column-gap

1

2

3

4

5

6

```
<style>
  .container{
    display: grid;
    grid-template-columns: repeat(3, 1fr) 200px;
    /*grid-gap: 20px 10px;*/
    grid-column-gap: 50px;
  }
  .item{
    background-color: brown;
    padding: 10px;
    border-radius: 15px;
    font-size: 20px;
  }
</style>
```

Gap só das colunas

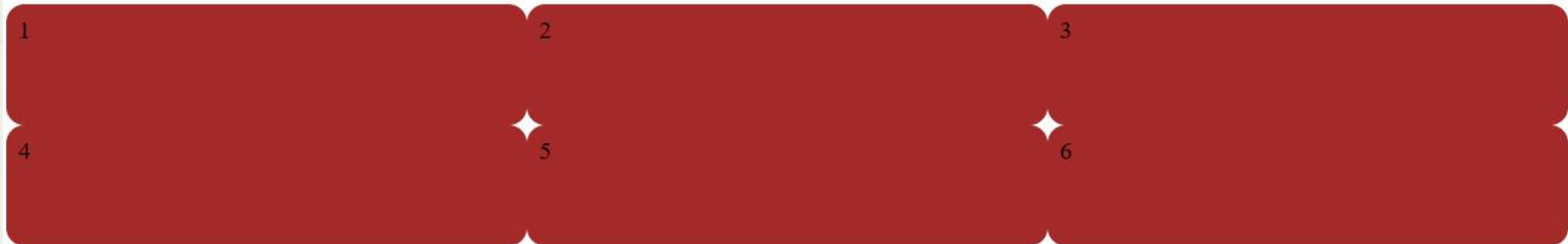
grid-row-gap



```
<style>
.container{
  display: grid;
  grid-template-columns: repeat(3, 1fr) 200px;
  /*grid-gap: 20px 10px;*/
  /*grid-column-gap: 50px;*/
  grid-row-gap: 50px;
}
.item{
  background-color: #brown;
  padding: 10px;
  border-radius: 15px;
  font-size: 20px;
}
</style>
```

Gap só das linhas

grid-auto-rows



```
<style>
.container{
  display: grid;
  grid-template-columns: auto auto auto;
  grid-auto-rows: 100px;
}
.item{
  background-color: #8B4513;
  padding: 10px;
  border-radius: 15px;
  font-size: 20px;
}
</style>
```

Altura das linhas

justify-content:center/start/end



```
<style>
.container{
  grid-gap:10px;
  background-color: cornflowerblue;
  display: grid;
  grid-template-columns: repeat(3, 200px);
  justify-content: center;
}
.item{
  background-color: brown;
  padding: 10px;
  border-radius: 15px;
  font-size: 20px;
}
</style>
```

Alinhamento horizontal
dos itens do container

justify-content:space-around



```
<style>
  .container{
    grid-gap:10px;
    background-color: #cornflowerblue;
    display: grid;
    grid-template-columns: repeat(3, 200px);
    justify-content: space-around;
  }
  .item{
    background-color: #brown;
    padding: 10px;
    border-radius: 15px;
    font-size: 20px;
  }
</style>
```

Alinhamento com o mesmo espaço entre e ao redor das colunas.

justify-content:space-between



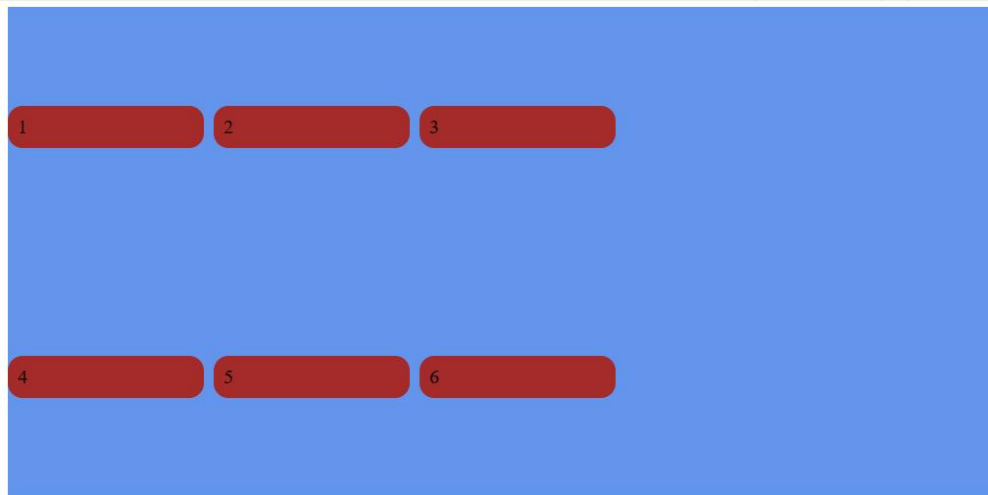
```
<style>
.container{
  grid-gap:10px;
  background-color: cornflowerblue;
  display: grid;
  grid-template-columns: repeat(3, 200px);
  justify-content: space-between;
}
.item{
  background-color: brown;
  padding: 10px;
  border-radius: 15px;
  font-size: 20px;
}
</style>
```

Alinhamento com o mesmo espaço entre as colunas.

align-items:center/start/end/space-between/space-around

```
<style>
.container{
  height: 500px;
  grid-gap:10px;
  background-color: cornflowerblue;
  display: grid;
  grid-template-columns: repeat(3, 200px);
  align-items: center;
}
.item{
  background-color: brown;
  padding: 10px;
  border-radius: 15px;
  font-size: 20px;
}
</style>
```

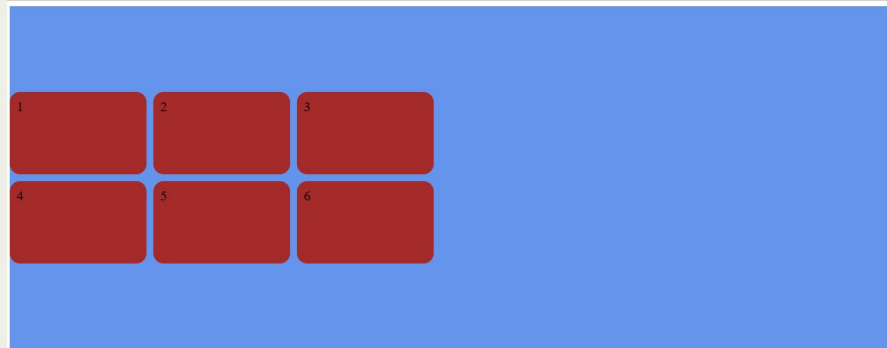
Alinhamento vertical dos itens em cada linha do container



align-content: center/start/end

Alinhamento vertical de todos os itens do container

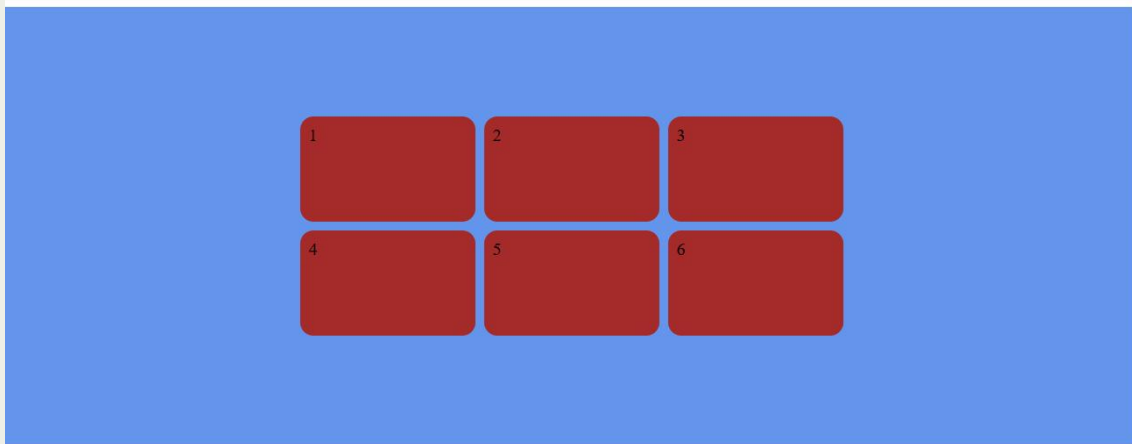
```
<style>
.container{
  height: 500px;
  grid-gap:10px;
  background-color: cornflowerblue;
  display: grid;
  grid-template-columns: repeat(3, 200px);
  align-content: center;
}
.item{
  height: 100px;
  background-color: brown;
  padding: 10px;
  border-radius: 15px;
  font-size: 20px;
}
</style>
```



align-content e justify-content

Alinhamento horizontal e vertical de todos os itens

```
<style>
.container{
  height: 500px;
  grid-gap:10px;
  background-color: cornflowerblue;
  display: grid;
  grid-template-columns: repeat(3, 200px);
  align-content: center;
  justify-content:center;
}
.item{
  height: 100px;
  background-color: brown;
  padding: 10px;
  border-radius: 15px;
  font-size: 20px;
}
</style>
```



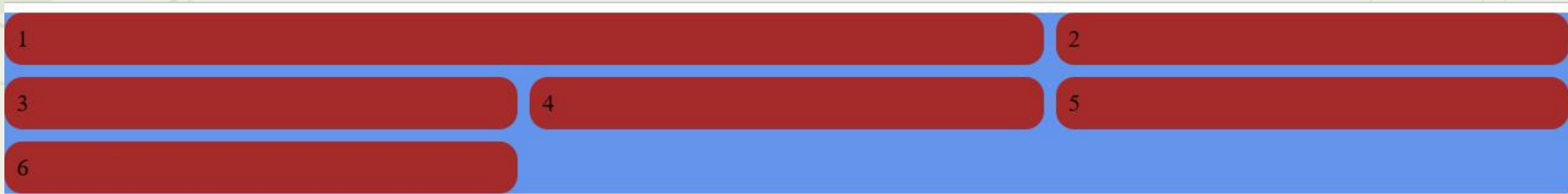
Colunas

1	2	3
4	5	6

Linhas

1	1	2	3
2	4	5	6
3			

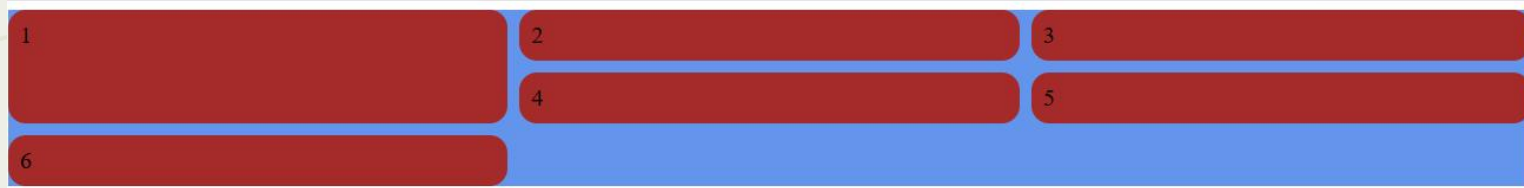
Mesclando colunas: grid-column-start/end



```
<style>
  .container{
    grid-gap:10px;
    background-color: #cornflowerblue;
    display: grid;
    grid-template-columns: repeat(3, auto);
  }
  .item{
    background-color: #brown;
    padding: 10px;
    border-radius: 15px;
    font-size: 20px;
  }
  .item1{
    grid-column-start: 1;
    grid-column-end: 3;
  }
</style>
```

O item 1 inicia na coluna 1 e termina na coluna 3.

Mesclando colunas: grid-row-start/end



```
<style>
  .container{
    grid-gap:10px;
    background-color: cornflowerblue;
    display: grid;
    grid-template-columns: repeat(3, auto);
  }
  .item{
    background-color: brown;
    padding: 10px;
    border-radius: 15px;
    font-size: 20px;
  }
  .item1{
    grid-row-start: 1;
    grid-row-end: 3;
  }
</style>
```

O item 1 inicia na linha 1 e termina na coluna 3.

grid-column e grid-row

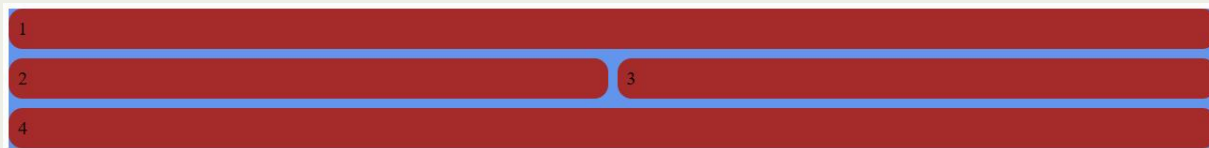


```
<style>
.container{
  grid-gap:10px;
  background-color: #6495ED;
  display: grid;
  grid-template-columns: repeat(3, auto);
}
.item{
  background-color: #8B4513;
  padding: 10px;
  border-radius: 15px;
  font-size: 20px;
}
.item1{
  grid-column: 1/3;
  grid-row: 1/3;
}
</style>
```

Outra forma de fazer

Template Area

```
<body>
  <div class="container">
    <div class="item item1">1</div>
    <div class="item item2">2</div>
    <div class="item item3">3</div>
    <div class="item item4">4</div>
  </div>
</body>
```



```
.container{
  background-color: cornflowerblue;
  display: grid;
  grid-gap:10px;
  grid-template-areas:
    "item1 item1"
    "item2 item3"
    "item4 item4"
  ;
}
.item{
  background-color: brown;
  padding: 10px;
  border-radius: 15px;
  font-size: 20px;
}
.item1{
  grid-area: item1;
}
.item2{
  grid-area: item2;
}
.item3{
  grid-area: item3;
}
.item4{
  grid-area: item4;
}
```

Template Area



```
.container{  
  background-color: cornflowerblue;  
  display: grid;  
  grid-gap:10px;  
  grid-template-areas:  
    "item1 item1 item1"  
    "item2 item3 item4"  
  ;  
}
```

Template Area



```
.container{  
  background-color: cornflowerblue;  
  display: grid;  
  grid-gap:10px;  
  grid-template-areas:  
    "item1 item1 item1"  
    "item2 item2 item4"  
    "item3 item3 ."  
  ;  
}
```

Exemplo Layout

```
<style>
.container {
  display:grid;
  grid-gap: 5px 10px;
  grid-template-areas:
    "header header header header header header"
    "sidenav content content content content content"
    "footer footer footer footer footer footer footer"
}
.item {
  background-color: #brown;
  padding: 10px;
  border-radius: 15px;
  font-size: 20px;
  color: #aliceblue;
}
.header {
  grid-area: header;
}
.sidenav {
  grid-area: sidenav;
}
.content {
  grid-area: content;
}
.footer {
  grid-area: footer;
}
</style>
```

```
<body>
  <div class="container">
    <div class="item header">Cabeçalho</div>
    <div class="item sidenav">Barra lateral</div>
    <div class="item content">Conteúdo</div>
    <div class="item footer">Rodapé</div>
  </div>
</body>
```

Cabeçalho

Barra lateral

Conteúdo

Rodapé

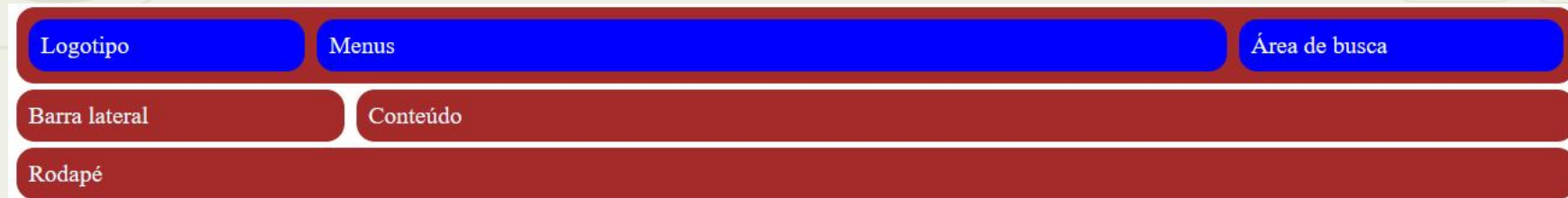
Grid dentro de Grid - Layout

```
<body>
  <div class="container">
    <div class="item header">
      <div class="itemmenu logo">Logo</div>
      <nav class="itemmenu menu">Menus</nav>
      <div class="itemmenu busca">Área de busca</div>
    </div>
    <div class="item sidenav">Barra lateral</div>
    <div class="item content">Conteúdo</div>
    <div class="item footer">Rodapé</div>
  </div>
</body>
```

```
<style>
  .container {
    display:grid;
    grid-gap: 5px 10px;
    grid-template-areas:
      "header header header header header header header"
      "sidenav content content content content content content"
      "footer footer footer footer footer footer footer"
  }
  .header {
    display:grid;
    grid-gap: 5px 10px;
    grid-template-areas:
      "logo menu menu menu menu menu busca"
  }
  .item {
    background-color: #brown;
    padding: 10px;
    border-radius: 15px;
    font-size: 20px;
    color: #aliceblue;
  }
  .itemmenu{
    background-color: #blue;
    padding: 10px;
    border-radius: 15px;
    font-size: 20px;
    color: #aliceblue;
  }
</style>
```

```
.header {
  grid-area: header;
}
.sidenav {
  grid-area: sidenav;
}
.content {
  grid-area: content;
}
.footer {
  grid-area: footer;
}
.menu{
  grid-area: menu;
}
.busca{
  grid-area: busca;
}
</style>
```

Resultado



Flex

- <https://origamid.com/projetos/flexbox-guia-completo/>
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- <https://www.joshwcomeau.com/css/interactive-guide-to-flexbox/>
- <https://www.alura.com.br/artigos/css-guia-do-flexbox>
- <https://www.devmedia.com.br/css3-flexbox-funcionamento-e-propriedades/29532>

Grid

- <https://www.origamid.com/projetos/css-grid-layout-guia-completo/>
- <https://www.treinaweb.com.br/blog/css-grid-um-guia-interativo-parte-1-containers>
- <https://learncssgrid.com/>
- <https://css-tricks.com/snippets/css/complete-guide-grid/>
- <https://developer.mozilla.org/en-US/docs/Web/CSS/grid>

Ambientes de desenvolvimento

Desktop

- [Notepad++](#)
- [Visual Studio Code](#) - Extensão Live Preview
- DevTools do navegador (Inspecionar)

Online

- [Live Environment for Agents - tiia.com.br](#)
- [Codepen](#)
- [JSFiddle](#)
- [JSBin](#)



Hora da Tarefa!

Vamos começar a customizar o visual do seu site!

Defina:

- Espaçamentos
- Esquema de cores
- Alinhamentos dos textos
- Tipografia

Até a próxima semana!
;)