

CSS Grid Layout

Grid Layout

- O CSS Grid Layout Module oferece um sistema de layout baseado em grid, com linhas e colunas, facilitando o design de páginas web sem a necessidade de usar floats e posicionamento.
- Um layout de grade consiste em um elemento pai, com um ou mais elementos filho.

1	2	3
4	5	6
7	8	9

Propriedade Display

- Um elemento HTML se torna um grid container quando sua propriedade **display** é definida como **grid** ou **inline-grid**.

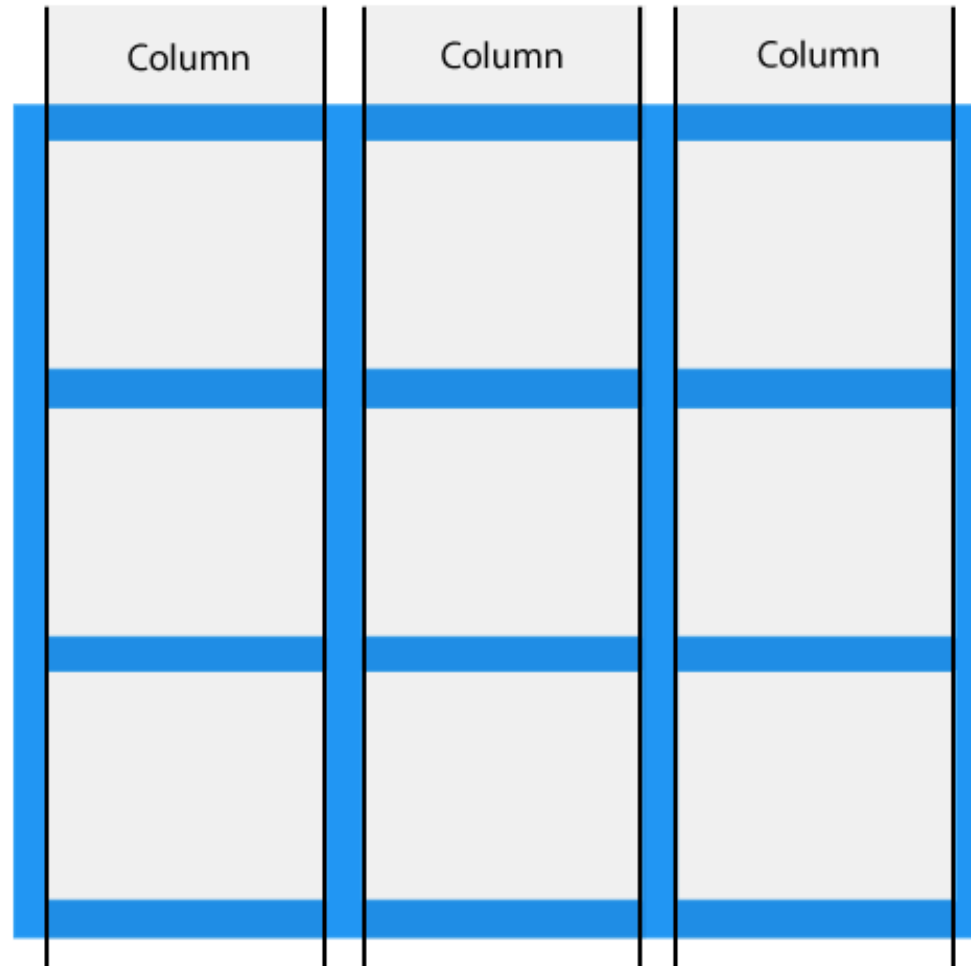
```
.grid-container {  
  display: grid;  
}
```

```
.grid-container {  
  display: inline-grid;  
}
```

- Todos os filhos diretos do grid container automaticamente se tornam *grid itens*(itens de grade).

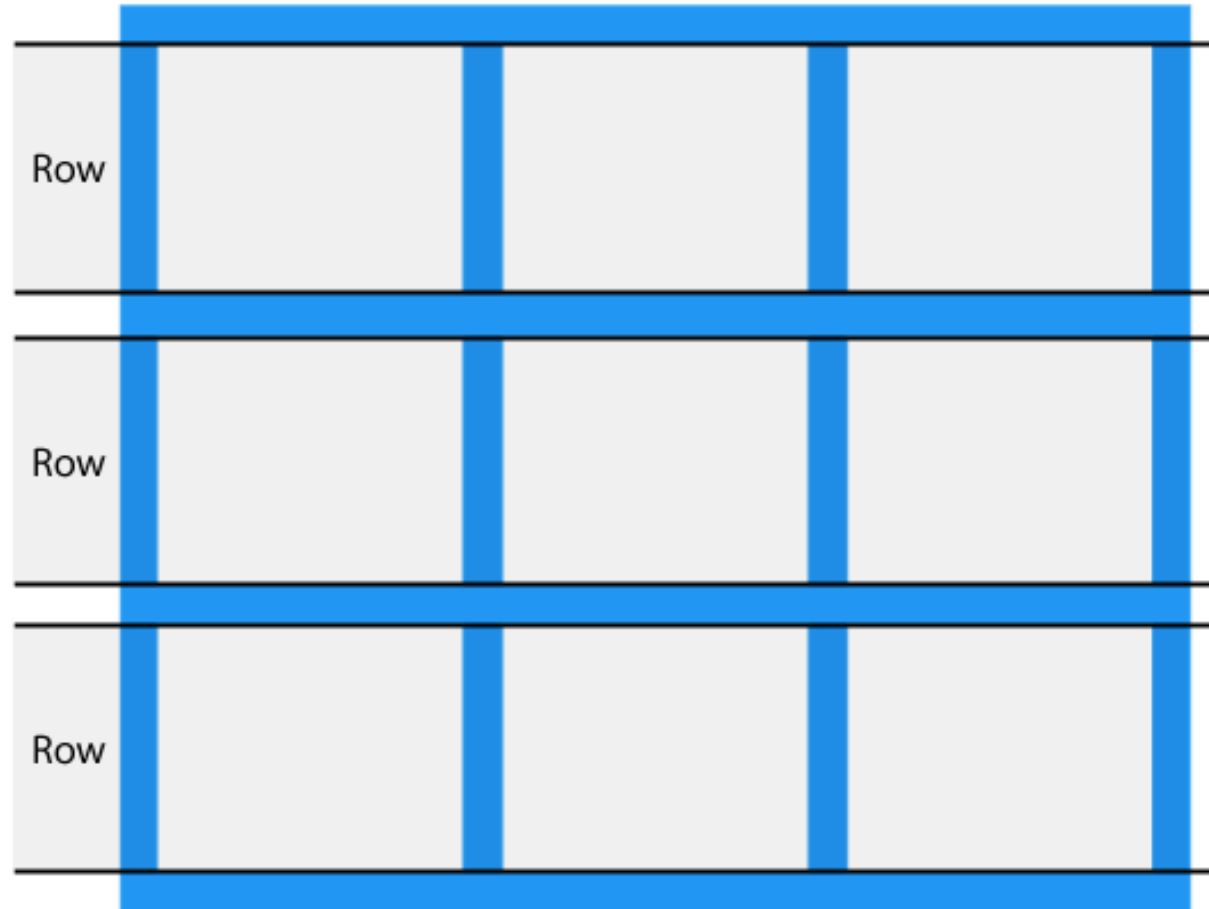
Grid Columns

- As linhas verticais dos grid itens são chamadas de *columns*(colunas).



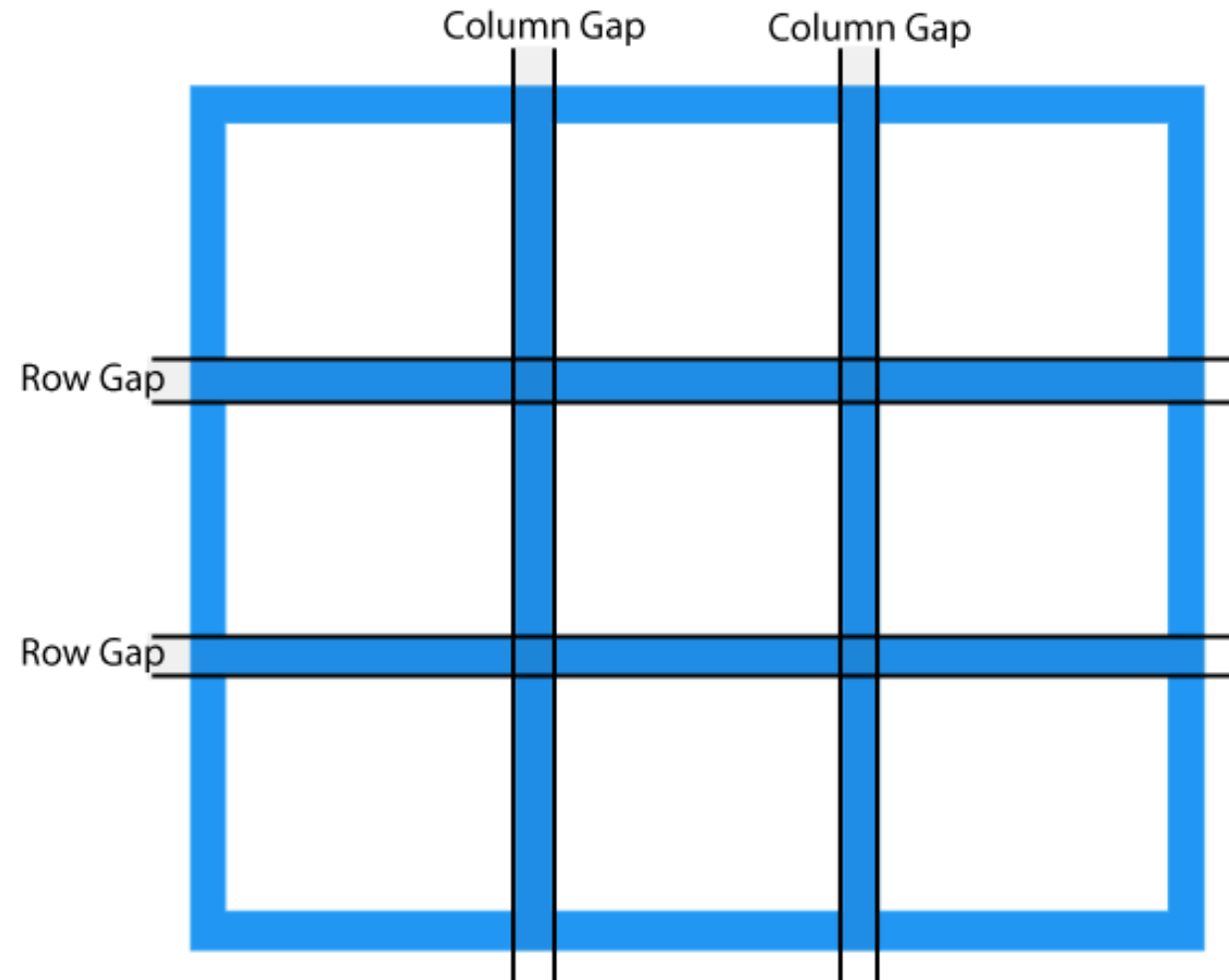
Grid rows

- As linhas horizontais dos grid itens são chamadas de *rows*(linhas).



Grid gap

- Os espaços entre cada coluna/linha são chamados de *gaps(lacunhas)*.



- Você pode ajustar o tamanho da lacuna usando uma das seguintes propriedades:

- `column-gap`
- `row-gap`
- `Gap`

- Exemplos:

- A propriedade `column-gap` define o intervalo entre as colunas:
- A propriedade `row-gap` define o intervalo entre as linhas:

```
.grid-container {  
  display: grid;  
  column-gap: 50px;  
}
```

```
.grid-container {  
  display: grid;  
  row-gap: 50px;  
}
```

- A propriedade **gap** é uma propriedade atalho para as propriedades **row-gap** e **column-gap**:

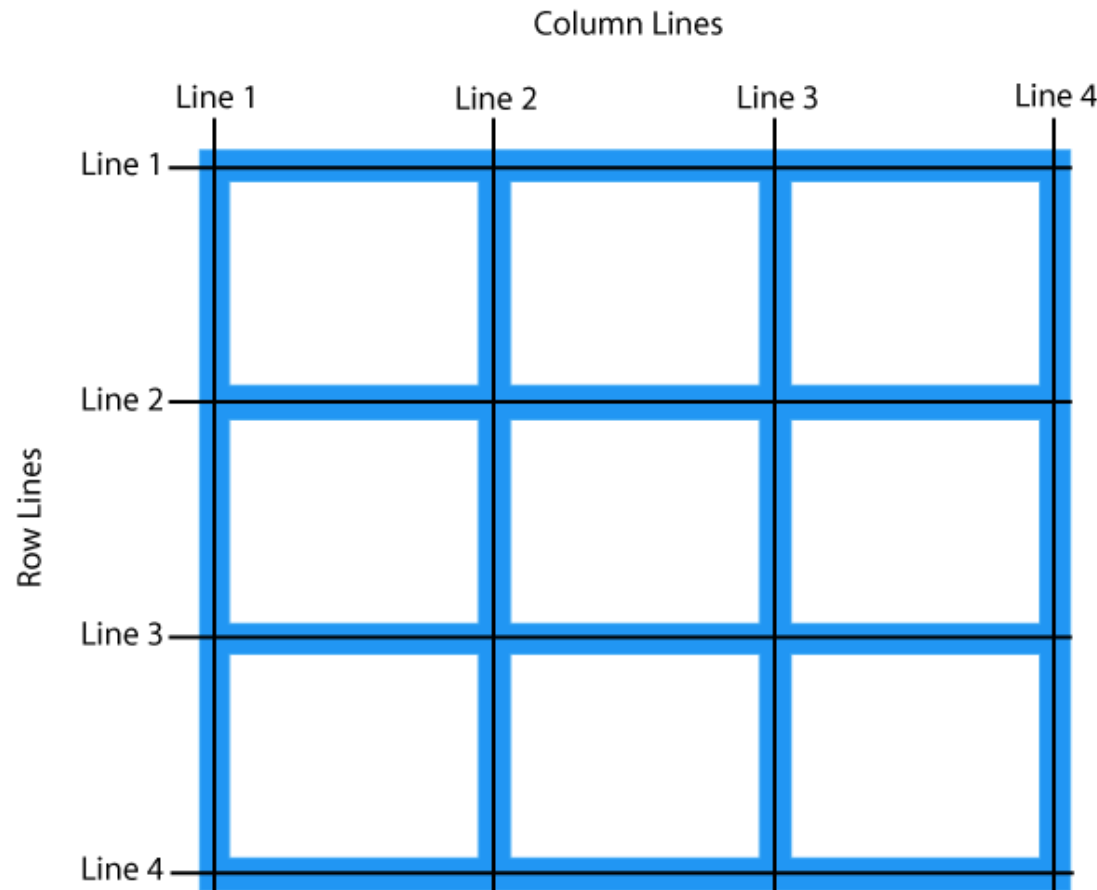
```
.grid-container {  
  display: grid;  
  gap: 50px 100px;  
}
```

- A propriedade **gap** também pode ser usada para definir o **row-gap** e o **column-gap** em um só valor:

```
.grid-container {  
  display: grid;  
  gap: 50px;  
}
```


Grid lines

- As linhas entre as colunas são chamadas de column lines .
- As linhas entre as linhas são chamadas de row lines .



- Consulte os números de linha ao colocar um grid item em um grid container

- Exemplos:

- Coloque um item de grade na linha de coluna 1 e deixe-o terminar na linha de coluna 3:

```
.item1 {  
  grid-column-start: 1;  
  grid-column-end: 3;  
}
```

1		2
3	4	5
6	7	8

- Coloque um item de grade na linha 1 da linha e deixe-o terminar na linha 3 da linha:

```
.item1 {  
  grid-row-start: 1;  
  grid-row-end: 3;  
}
```

1	2	3
	4	5
6	7	8

Propriedades CSS Grid

Propriedade	Descrição
column-gap	Especifica o espaço entre as colunas
gap	Um propriedade atalho para as propriedades <i>row-gap</i> e a <i>column-gap</i>
grid	Uma propriedade atalho para as propriedades <i>grid-template-rows</i> , <i>grid-template-columns</i> , <i>grid-template-areas</i> , <i>grid-auto-rows</i> , <i>grid-auto-columns</i> , e a <i>grid-auto-flow</i>
grid-area	Especifica um nome para um grid item, ou essa propriedade é um atalho para as propriedades <i>grid-row-start</i> , <i>grid-column-start</i> , <i>grid-row-end</i> , e <i>grid-column-end</i>
grid-auto-columns	Especifica um tamanho de coluna padrão
grid-auto-flow	Especifica como os itens colocados automaticamente são inseridos na grade
grid-auto-rows	Especifica um tamanho de linha padrão
grid-column	Uma propriedade atalho para as propriedades <i>grid-column-start</i> e <i>grid-column-end</i>
grid-column-end	Especifica onde o grid item termina
grid-column-gap	Especifica o tamanho do espaço entre as colunas

grid-column-start	Especifica onde iniciar o grid item
grid-gap	Uma propriedade abreviada para as propriedades grid-row-gap e grid-column-gap
grid-row	Uma propriedade atalho para as propriedades grid-row-start e grid-row-end
grid-row-end	Especifica onde terminar o grid item
grid-row-gap	Especifica o tamanho do espaço entre as linhas
grid-row-start	Especifica onde iniciar o grid item
grid-template	Uma propriedade atalho para as propriedades grid-template-rows, grid-template-columns e grid-area
grid-template-areas	Especifica como exibir colunas e linhas, usando grid itens nomeados
grid-template-columns	Especifica o tamanho das colunas e quantas colunas em um grid layout
grid-template-rows	Especifica o tamanho das linhas em um grid layout
row-gap	Especifica o tamanho entre as linhas da grade

Grid Container

- Para fazer um elemento HTML se comportar como um grid container, você deve definir a propriedade `display` como `grid` ou `inline-grid`.
- Os grid containers consistem em grid items, colocados dentro de colunas e linhas.

A propriedade grid-template-columns

- A propriedade **grid-template-columns** define o número de colunas em seu layout de grade e pode definir a largura de cada coluna.
- O valor é uma lista separada por espaços, onde cada valor define a largura da respectiva coluna.
- Se você quiser que seu layout de grade contenha 4 colunas, especifique a largura das 4 colunas ou "auto" se todas as colunas tiverem a mesma largura.

- Exemplo:

- Um grid com 4 colunas:

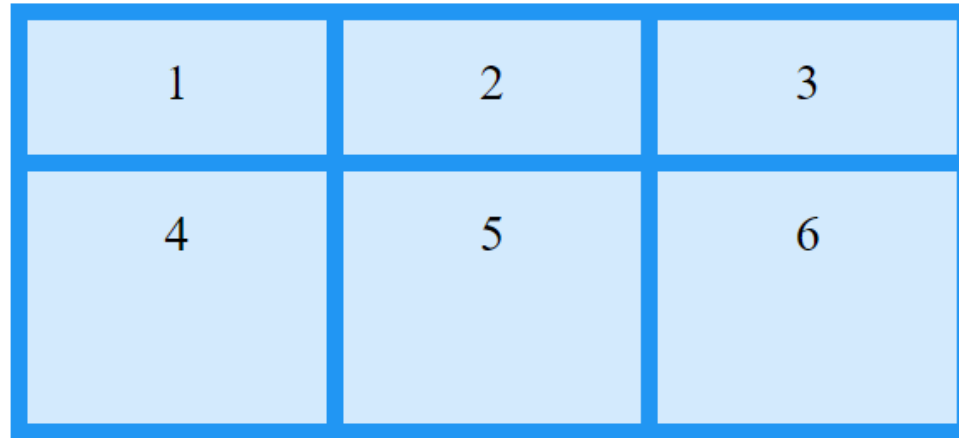
```
.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto auto;  
}
```

- **Nota:** Se você tiver mais de 4 itens em uma grade de 4 colunas, a grade adicionará automaticamente uma nova linha para colocar os itens.
- A propriedade `grid-template-columns` também pode ser usada para especificar o tamanho (largura) das colunas.
 - Um grid com tamanho para as 4 colunas:

```
.grid-container {  
  display: grid;  
  grid-template-columns: 80px 200px auto 40px;  
}
```

A propriedade grid-template-rows

- A propriedade **grid-template-rows** define a altura de cada linha.
- O valor é uma lista separada por espaços, onde cada valor define a altura da respectiva linha:



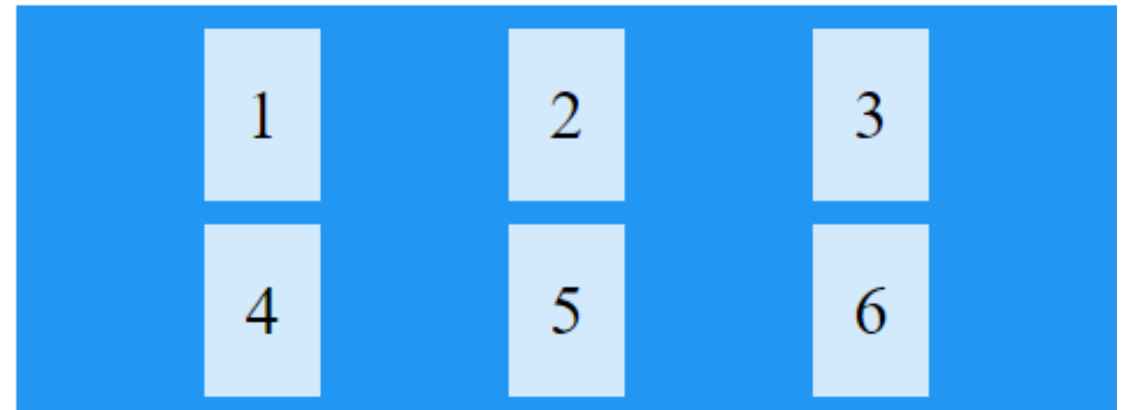
1	2	3
4	5	6

```
.grid-container {  
  display: grid;  
  grid-template-rows: 80px 200px;  
}
```

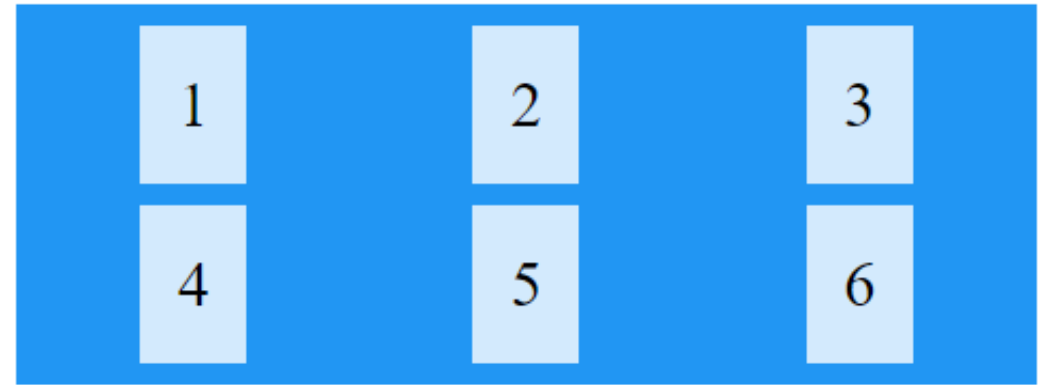

A propriedade justify-content

- A propriedade **justify-content** é usada para alinhar toda a grade dentro do container.
- **Observação:** a largura total da grade deve ser menor que a largura do contêiner para que a propriedade **justify-content** tenha algum efeito.
- Exemplo:

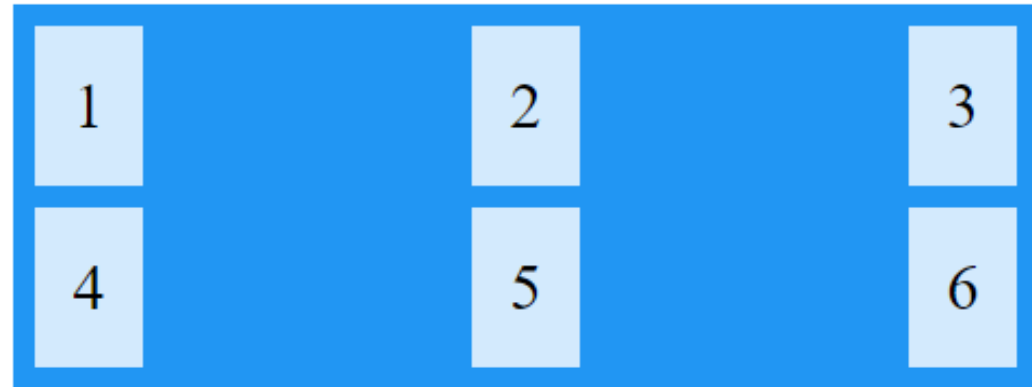
```
.grid-container {  
  display: grid;  
  justify-content: space-evenly;  
}
```



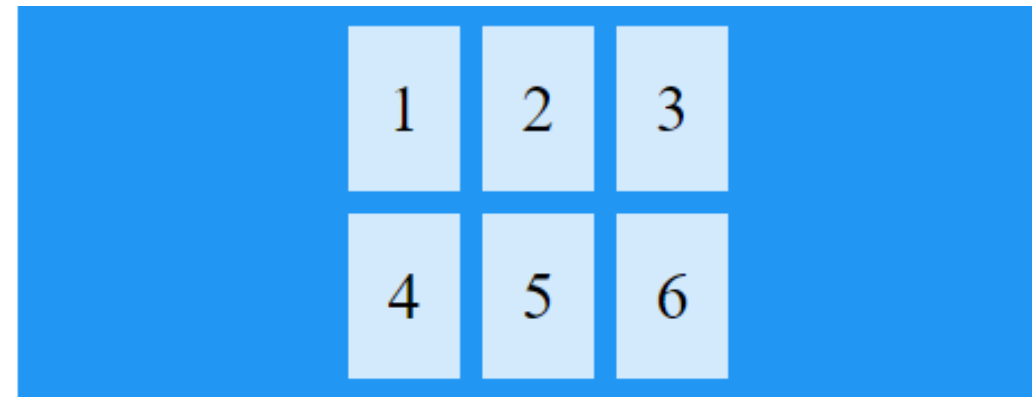
```
.grid-container {  
  display: grid;  
  justify-content: space-around;  
}
```



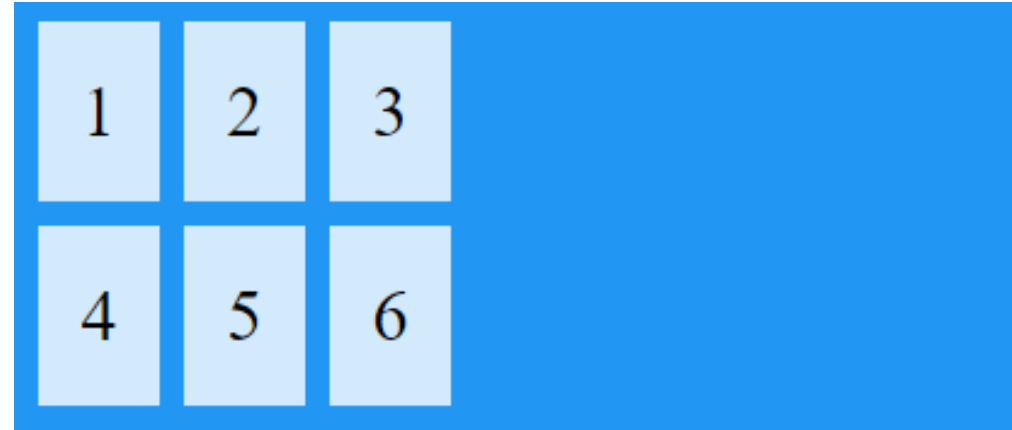
```
.grid-container {  
  display: grid;  
  justify-content: space-between;  
}
```



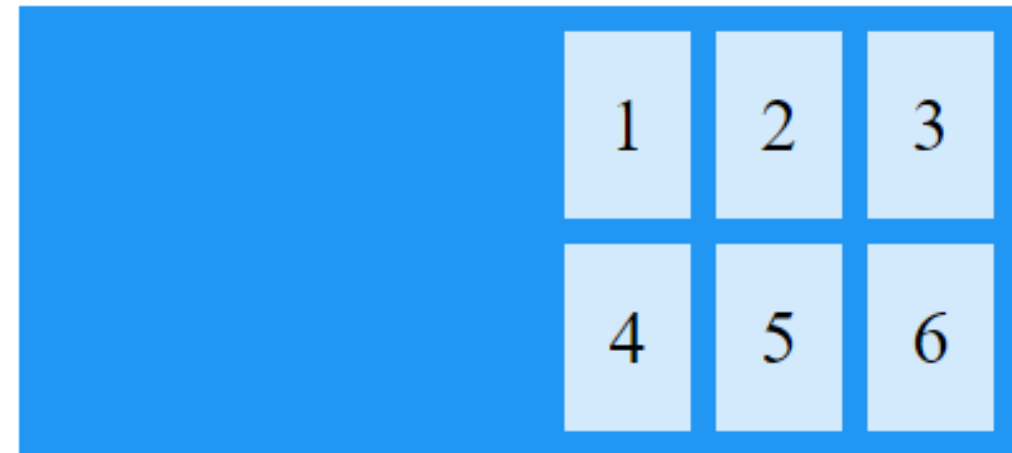
```
.grid-container {  
  display: grid;  
  justify-content: center;  
}
```



```
.grid-container {  
  display: grid;  
  justify-content: start;  
}
```



```
.grid-container {  
  display: grid;  
  justify-content: end;  
}
```

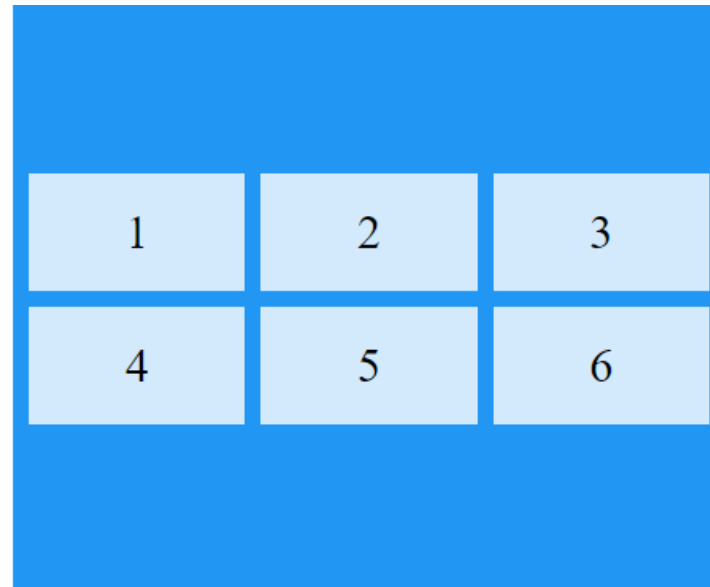


A propriedade align-content

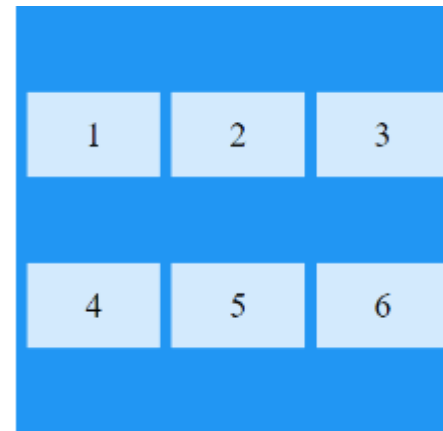
- A propriedade **align-content** é usada para alinhar verticalmente toda a grade dentro do container.
- **Nota:** A altura total da grade deve ser menor que a altura do contêiner para que a align-content propriedade tenha algum efeito.

- Exemplo:

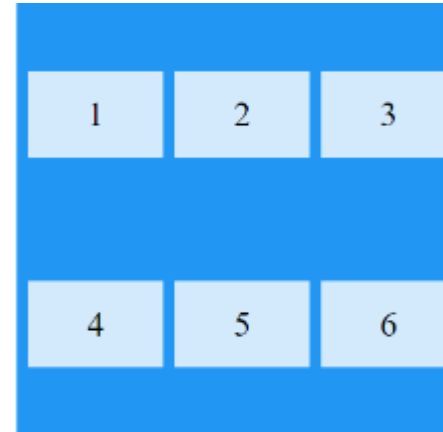
```
.grid-container {  
  display: grid;  
  height: 350px;  
  align-content: center;  
}
```



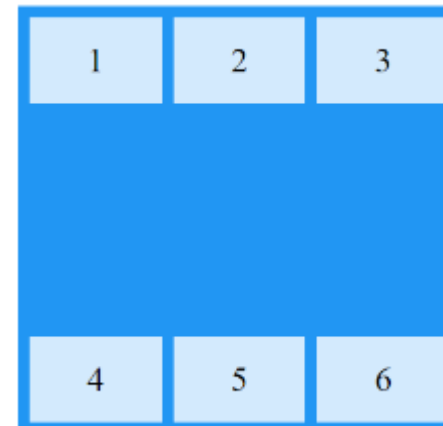
```
.grid-container {  
  display: grid;  
  height: 350px;  
  align-content: space-evenly;  
}
```



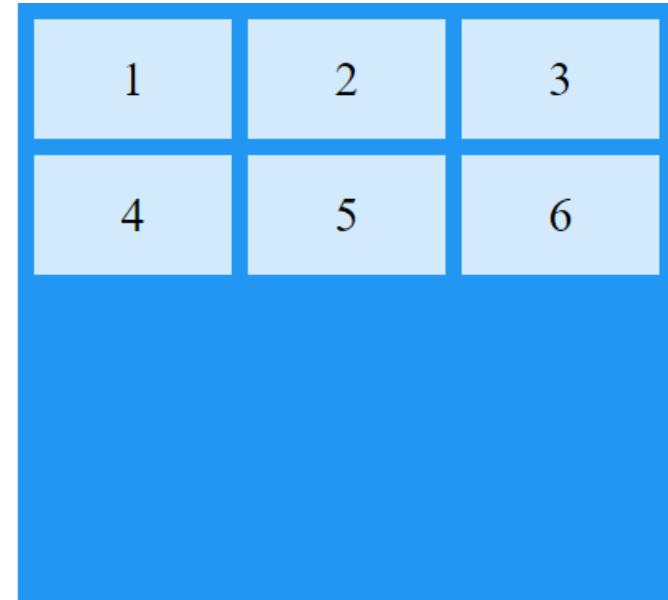
```
.grid-container {  
  display: grid;  
  height: 350px;  
  align-content: space-around;  
}
```



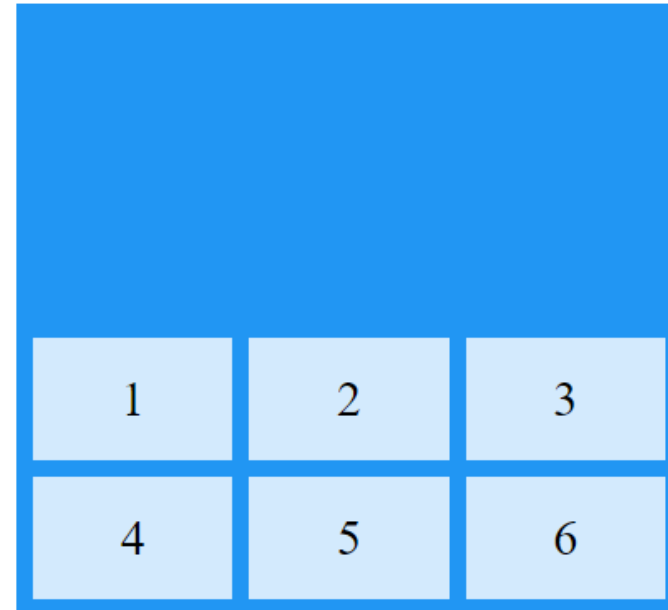
```
.grid-container {  
  display: grid;  
  height: 350px;  
  align-content: space-between;  
}
```



```
.grid-container {  
  display: grid;  
  height: 400px;  
  align-content: start;  
}
```



```
.grid-container {  
  display: grid;  
  height: 400px;  
  align-content: end;  
}
```



Elementos filhos (itens)

- Um *grid container*(container de grade) contém *grid items*(itens de grade).
- Por padrão, um contêiner tem um item de grade para cada coluna, em cada linha, mas você pode estilizar os itens de grade para que eles abranjam várias colunas e/ou linhas.

A propriedade grid-column

- A propriedade grid-column define em qual(is) coluna(s) colocar um item.
- Você define onde o item começará e onde o item terminará.
- **Nota:** A propriedade **grid-column** é uma propriedade abreviada para as propriedades **grid-column-start** e **grid-column-end**.
- Para colocar um item, você pode consultar os números de linha ou usar a palavra-chave "span" para definir quantas colunas o item abrangerá.

- Exemplos:

- Faça com que "item1" comece na coluna 1 e abranja 3 colunas:

```
.item1 {  
  grid-column: 1 / span 3;  
}
```

1			2	3	4
5	6	7	8	9	10
11	12	13	14	15	16

- Faça com que "item2" comece na coluna 2 e abranja 3 colunas:

```
.item2 {  
  grid-column: 2 / span 3;  
}
```

1	2			3	4
5	6	7	8	9	10
11	12	13	14	15	16

A propriedade grid-row:

- A propriedade **grid-row** define em qual linha colocar um item.
- Você define onde o item começará e onde o item terminará.
- **Nota:** A propriedade **grid-row** é uma propriedade abreviada para as propriedades **grid-row-start** e **grid-row-end**.
- Para colocar um item, você pode consultar os números de linha ou usar a palavra-chave "span" para definir quantas linhas o item abrangerá:

- Exemplos:

- Faça "item1" começar na linha 1 e terminar na linha 4:

```
.item1 {  
  grid-row: 1 / 4;  
}
```

1	2	3	4	5	6
	7	8	9	10	11
	12	13	14	15	16

- Faça "item1" começar na linha 1 e abranger 2 linhas:

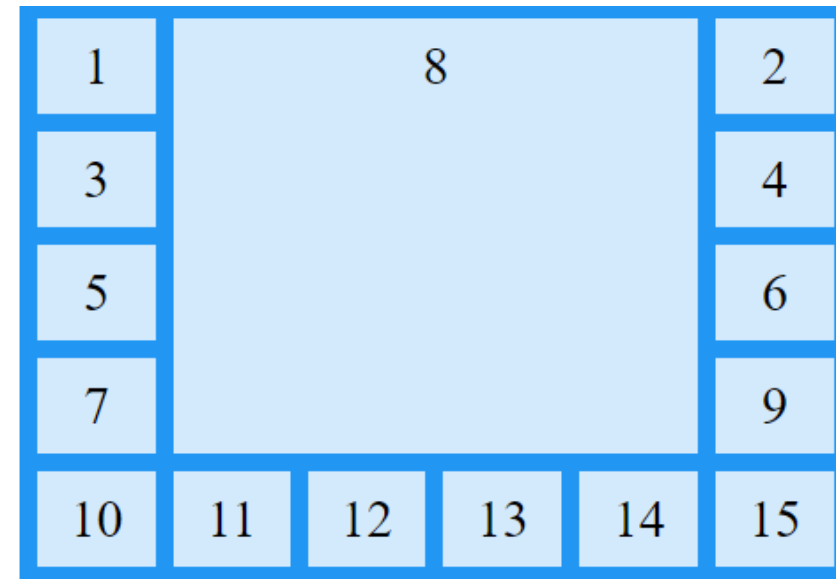
```
.item1 {  
  grid-row: 1 / span 2;  
}
```

1	2	3	4	5	6
	7	8	9	10	11
12	13	14	15	16	17

A propriedade grid-area

- A propriedade **grid-area** pode ser usada como uma propriedade atalho para as propriedades **grid-row-start**, **grid-column-start**, **grid-row-end** e **grid-column-end**.
- Exemplo:
 - Faça o "item8" começar na linha 1 da linha e na linha 2 da coluna e terminar na linha 5 da linha e na linha 6 da coluna:

```
.item8 {  
  grid-area: 1 / 2 / 5 / 6;  
}
```



- Faça o "item8" começar na linha 2 da linha e na linha 1 da coluna e abranger 2 linhas e 3 colunas:

```
.item8 {  
  grid-area: 2 / 1 / span 2 / span 3;  
}
```

1	2	3	4	5	6
8			7	9	10
			11	12	13

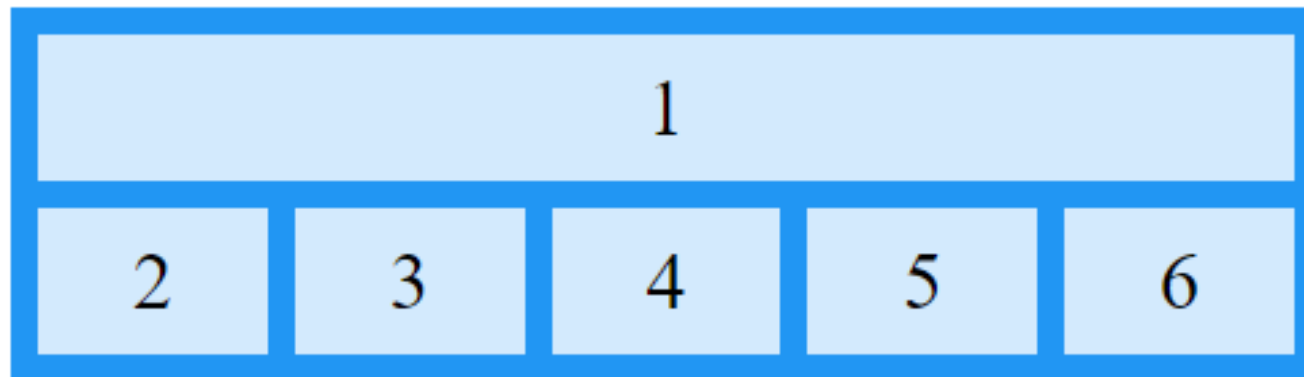
Nomeando grid itens

- A propriedade **grid-area** também pode ser usada para atribuir nomes a itens de grade.
- Itens de grade nomeados podem ser referenciados pela propriedade **grid-template-areas** do contêiner de grade.
- Cada linha é definida por apóstrofos (' ').
- As colunas em cada linha são definidas dentro dos apóstrofos, separadas por um espaço.
- **Nota:** Um sinal de ponto final representa um item de grade sem nome.

- Exemplo:

- Item1 recebe o nome "myArea" e abrange todas as cinco colunas em um grid layout de cinco colunas:

```
.item1 {  
  grid-area: area1;  
}  
.grid-container {  
  grid-template-areas: ' area1 area1 area1 area1 area1 ';  
}
```



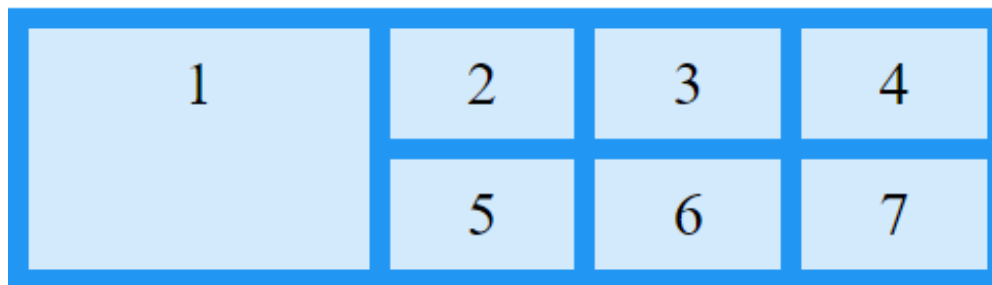
- Deixe "myArea" abranger duas colunas em um grid layout de cinco colunas (sinais de ponto representam itens sem nome):

```
.item1 {  
  grid-area: area1;  
}  
.grid-container {  
  grid-template-areas: area1 area1 ...';  
}
```



- Para definir duas linhas, defina a coluna da segunda linha dentro de outro conjunto de apóstrofos:
- Exemplo:
 - Faça com que "item1" abranja duas colunas e duas linhas:

```
.grid-container {  
  grid-template-areas: ' area1 area1 ...' ' area1 area1 ...';  
}
```



- Nomeie todos os itens e crie um modelo de página da Web pronto para uso:

```
.item1 { grid-area: cabecalho; }  
.item2 { grid-area: menu; }  
.item3 { grid-area: conteudo; }  
.item4 { grid-area: lateral; }  
.item5 { grid-area: rodape; }
```

```
.grid-container {  
  grid-template-areas:  
    ' cabecalho cabecalho cabecalho cabecalho cabecalho cabecalho '  
    'menu conteudo conteudo conteudo lateral lateral '  
    'menu rodape rodape rodape rodape rodape '  
}
```

