

IOS – Instituto de
Oportunidade Social

Aula CSS 10 - Grid CSS Layout - Parte 01



Objetivos



- > O que é Grid.
- > Termos utilizados pelo Grid
- > Propriedades do elemento pai (Grid Container).

IOS – Instituto de
Oportunidade Social

O que é Grid



O que é Grid

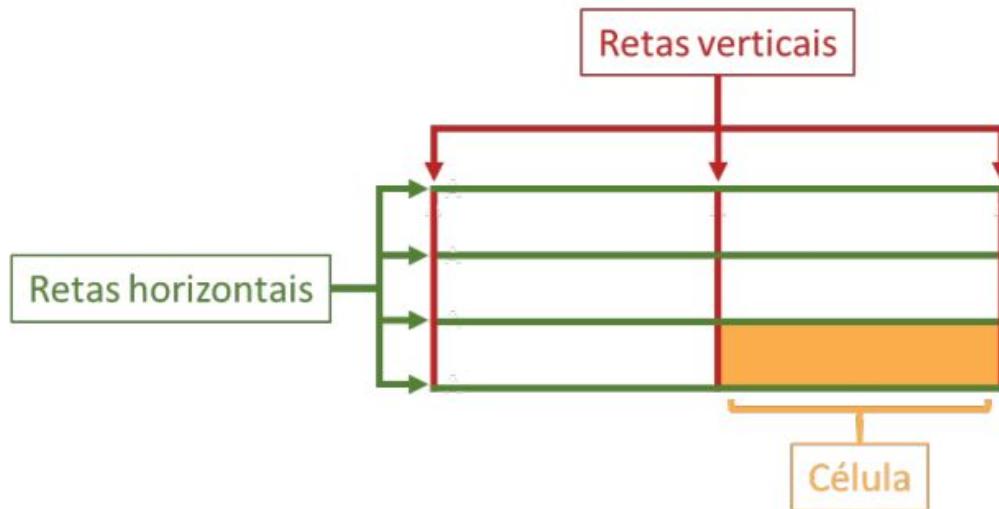


Você tem usado a propriedade **CSS Float** e o **CSS Flexible Box Layout** para criar a aparência de páginas web com várias colunas, porém existe uma técnica bastante eficiente para isso: o **CSS Grid Layout**.

O CSS Grid Layout é uma excelente forma de **dividir uma página em regiões principais ou definir o relacionamento em termos de medidas, posicionamento e camadas entre os diferentes componentes da marcação HTML**.

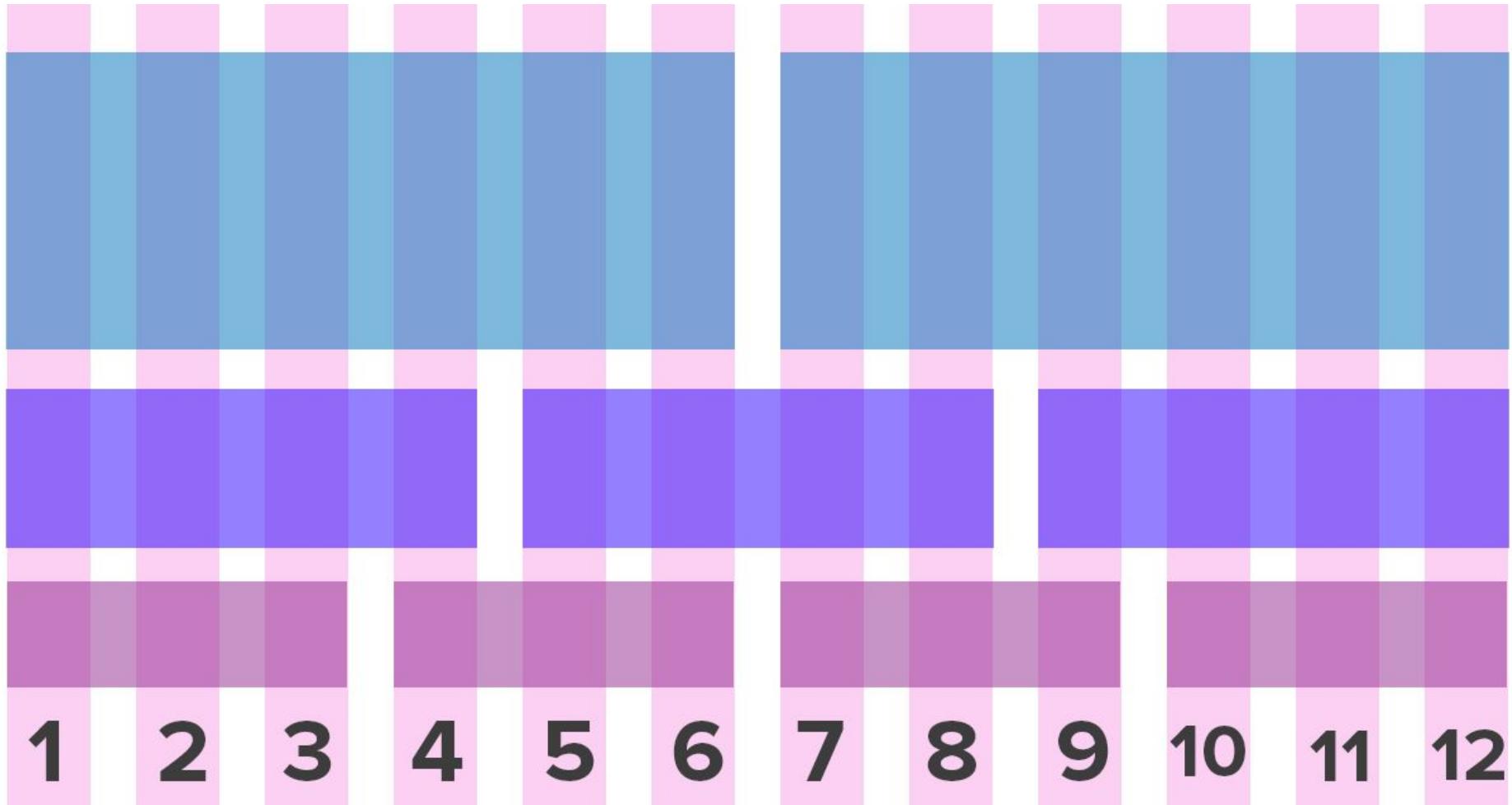
O que é Grid

Você deve entender o grid como um layout formado por **retas verticais e horizontais**, que **formam as células do grid**. Por exemplo, a imagem abaixo mostra um grid **3x2 (3 linhas e 2 colunas)** formado por três retas verticais e quatro retas horizontais, formando assim seis células.



- Tabelas bidimensionais
- Vantagens sobre tabelas

Elementos filhos (Grid Items)



IOS – Instituto de
Oportunidade Social

Termos utilizados pelo Grid



Termos utilizados pelo Grid



A proposta do Grid é definir o layout bidimensional de uma página web criando itens fixos ou flexíveis que podem ser configurados individualmente. Como tabelas, o layout Grid permite alinhar elementos em linha e colunas por meio de um método bidimensional para criação de layout utilizando CSS.

O **Grid tem vantagem** sobre tabelas por **não se basear em estruturação de conteúdo**. Possibilita uma **enorme variedade de layouts**. O Grid layout é o primeiro módulo no CSS criado especificamente para resolver problemas de layout.

Termos utilizados pelo Grid



Grid Container: é o elemento ou classe que definimos a propriedade `display: grid`. Então, esse elemento será o pai de todos os itens do Grid. Por exemplo, podemos configurar uma classe `.container` em um elemento `<section>` e vários itens dentro dessa seção.

Termos

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

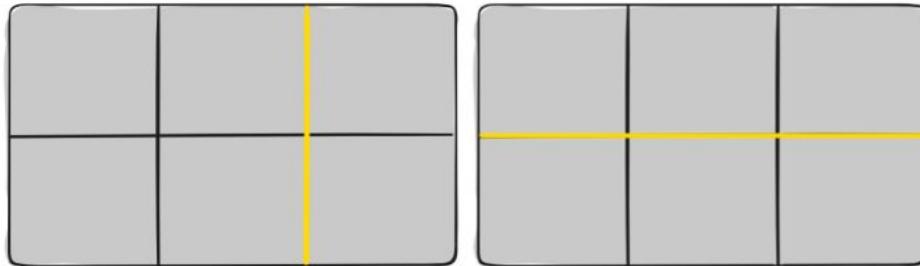
- **display: grid**
- Você sempre precisa definir um container para trabalhar com **Grid layout**

Termos utilizados pelo Grid

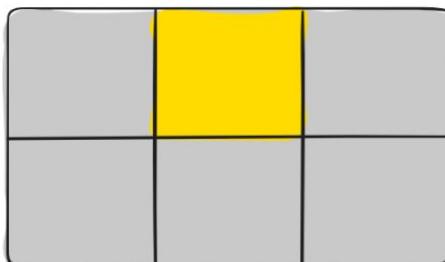


Grid Item: são os filhos do grid container.

Grid Line: são as linhas divisórias que formam a estrutura do grid. Elas podem ser as retas verticais ou horizontais e estar em qualquer lado da linha ou coluna.



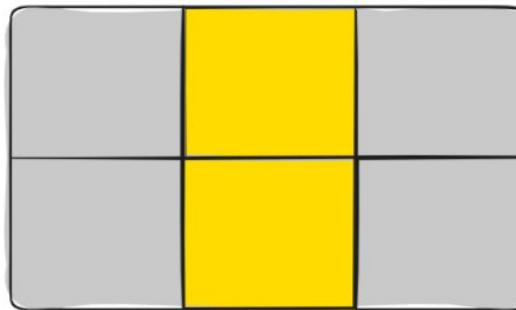
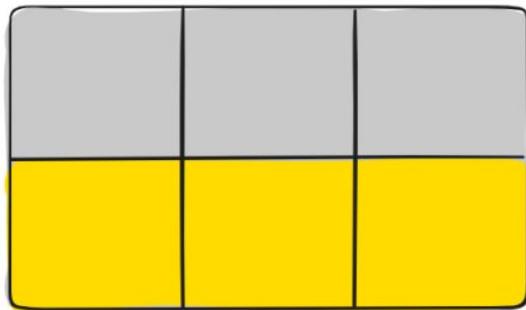
Grid Cell: é a célula do grid (unidade única do grid), ou seja, o espaço entre duas retas horizontais ou verticais adjacentes. Abaixo é mostrado o Grid Cell entre as retas 1 e 2 horizontais e as retas 2 e 3 verticais.



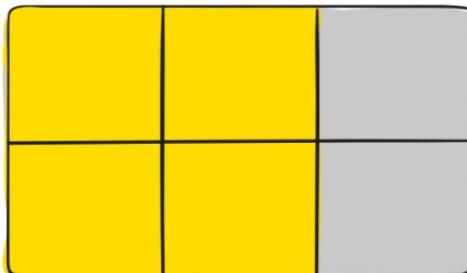
Termos utilizados pelo Grid



Grid Track: é uma trilha do grid, que pode ser uma linha ou uma coluna inteira.



Grid Area: O espaço total rodeado por quatro retas do grid. Grid Area pode ser composta de qualquer número de células do grid. Por exemplo, abaixo a Grid Area está entre as retas horizontais 1 e 3 e as retas verticais 1 e 3.



IOS – Instituto de
Oportunidade Social

Elemento pai (Grid Container)



Elemento pai (Grid Container)

- display
- grid-template-columns
- grid-template-rows
- grid-template-areas
- column-gap
- row-gap
- justify-items
- align-items
- justify-content
- align-content
- grid

Elemento pai (Grid Container)



> Propriedade display

- grid: gera um block grid
- inline-grid: gera um grid alinhado

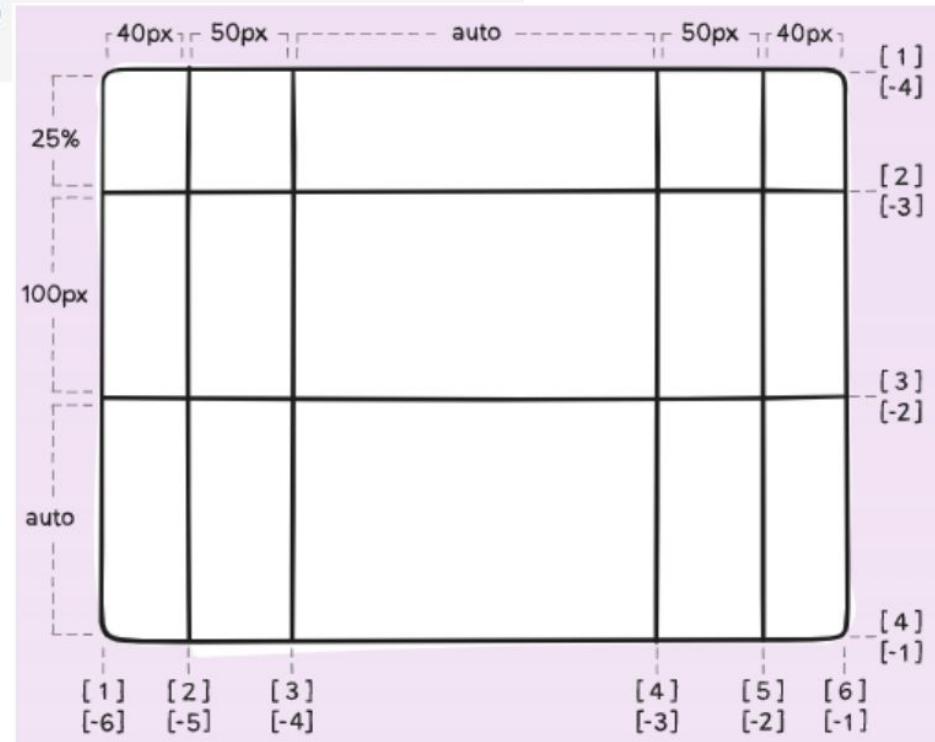
```
.container {  
    display: grid;  
}
```

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

Elemento pai (Grid Container)

> grid-template-columns / grid-template-rows

```
.container {  
    grid-template-columns: 40px 50px auto 50px 40px;  
    grid-template-rows: 25% 100px auto;  
}
```



Elemento pai (Grid Container)



Caso você queira dividir em partes iguais, pode usar a notação **repeat()**. Por exemplo:

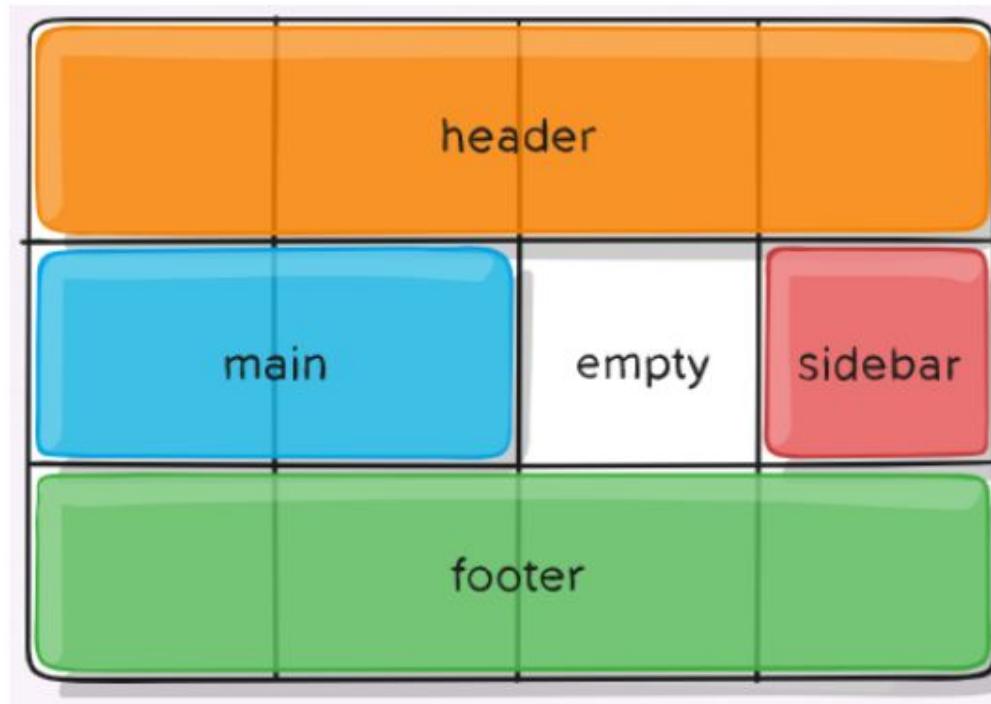
```
.container {  
    grid-template-columns: repeat(4, 1fr);  
}
```

> **grid-template-areas**

A propriedade **grid-template-areas** define um modelo de grid referenciando os nomes das áreas do grid que são especificadas pela propriedade **grid-area**. Repetir o nome de uma área do grid faz com que o conteúdo se espalhe por essas células com o nome repetido. Um ponto significa uma célula vazia

Elemento pai (Grid Container)

> grid-template-areas

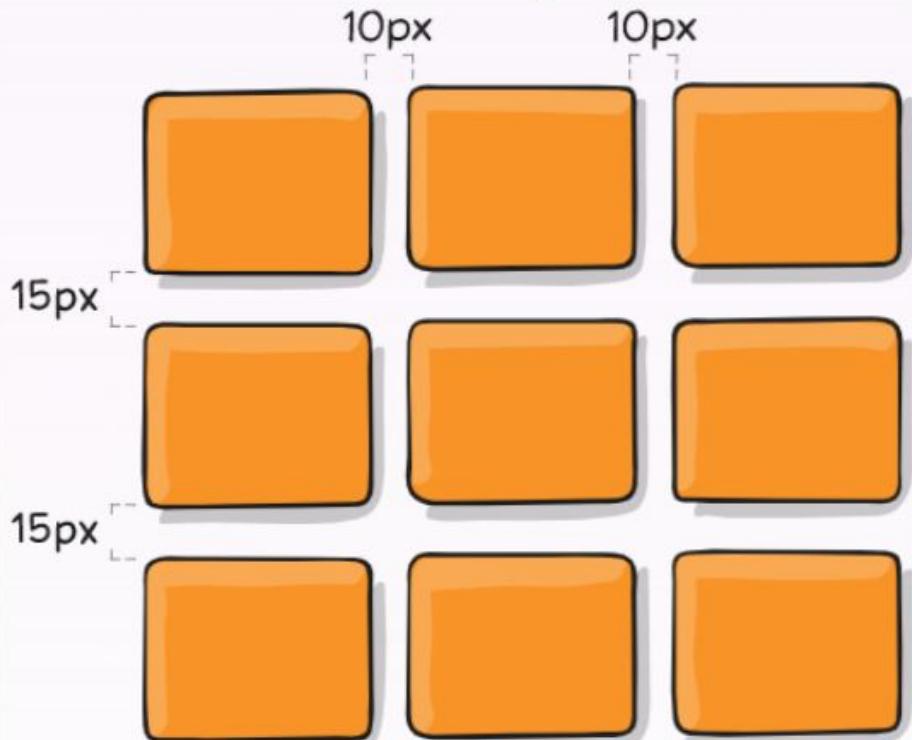


Elemento pai (Grid Container)

> column-gap / row-gap

```
.container {  
    grid-template-columns: repeat(3, 1fr);  
    grid-template-rows: 80px auto 80px;  
    column-gap: 10px;  
    row-gap: 15px;  
}
```

Elas especificam o espaçoamento (**gap**) entre as linhas e as colunas do grid, ou seja, a espessura das retas verticais ou horizontais



IOS – Instituto de
Oportunidade Social

Propriedade justify-items



Propriedade justify-items



A propriedade `justify-items` alinha horizontalmente os itens do grid. Os valores possíveis para configurar essa propriedade são:

start: os itens ficam alinhados na reta esquerda da célula do grid.

```
.container {  
    justify-items: start;  
}
```

Resultado:

Propriedade justify-items

end: os itens ficam alinhados na reta direita da célula do grid.

```
.container {  
    justify-items: end;  
}
```

Resultado:

Propriedade justify-items



center: os itens ficam centralizados horizontalmente na célula do grid.

```
.container {  
    justify-items: center;  
}
```

Resultado:

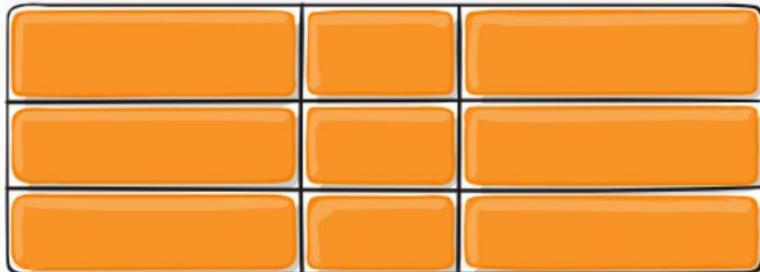
Propriedade justify-items



stretch: os itens preenchem todo o espaço da célula do grid.

```
.container {  
  justify-items: stretch;  
}
```

Resultado:



Esse mesmo comportamento para o alinhamento horizontal dos itens pode ser definido individualmente para cada item do grid com a propriedade **justify-self**.

IOS – Instituto de
Oportunidade Social

Propriedade align-items



Propriedade align-items

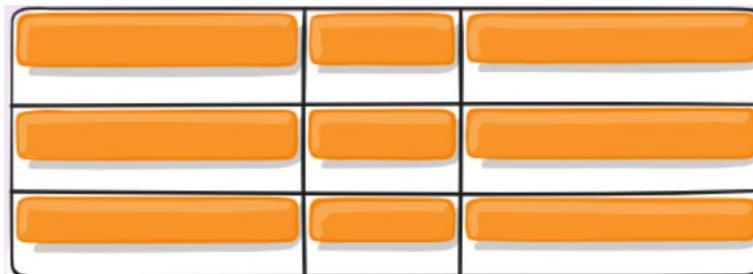


A propriedade align-items alinha verticalmente os itens do grid. Os valores possíveis para configurar essa propriedade são:

start: os itens ficam alinhados na reta superior da célula do grid.

```
.container {  
    align-items: start;  
}
```

Resultado:



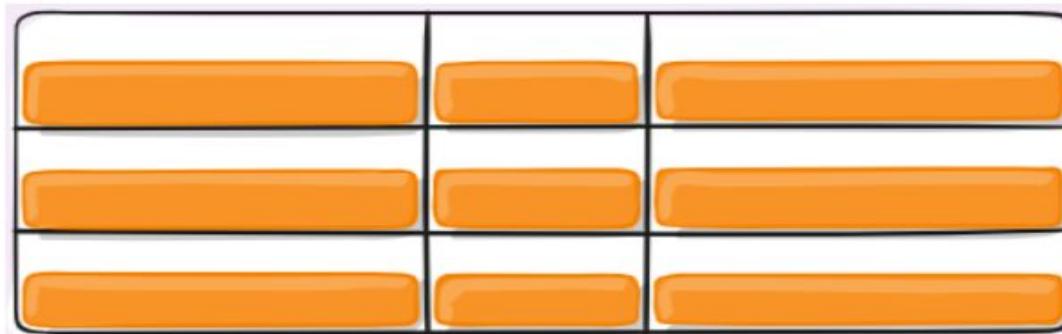
Propriedade align-items



end: os itens ficam alinhados na reta inferior da célula do grid.

```
.container {  
    align-items: end;  
}
```

Resultado:



Propriedade align-items



center: os itens ficam centralizados verticalmente na célula do grid.

```
.container {  
    align-items: center;  
}
```

Resultado:

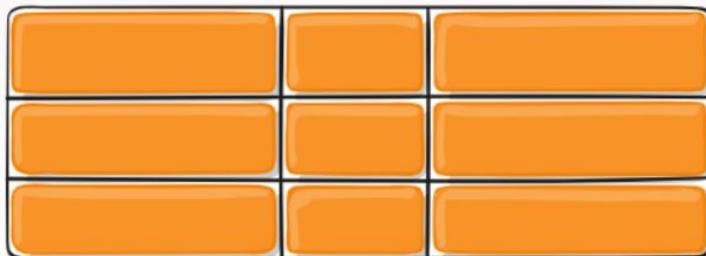
Propriedade align-items



stretch: os itens preenchem todo o espaço da célula do grid.

```
.container {  
    align-items: stretch;  
}
```

Resultado:



Esse mesmo comportamento para o alinhamento vertical dos itens pode ser definido individualmente para cada item do grid com a propriedade **align-self**.

IOS – Instituto de
Oportunidade Social

Propriedade justify-content



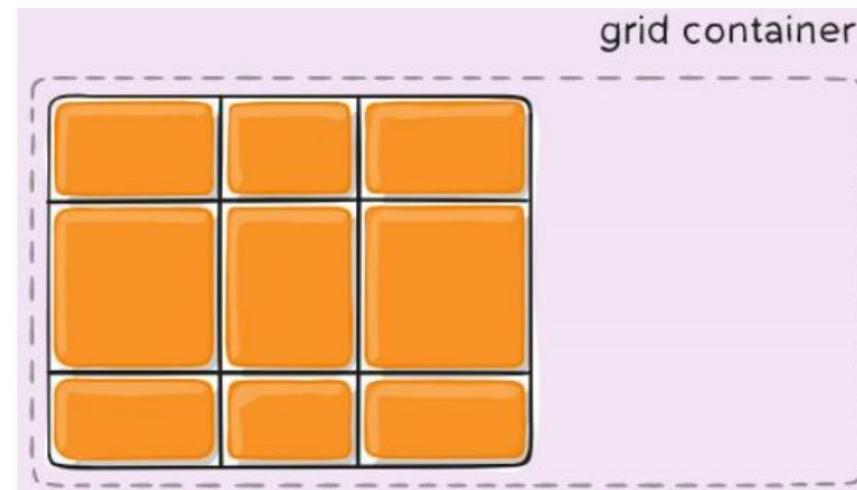
Propriedade justify-content

A propriedade **justify-content** alinha o grid completo ao longo do eixo horizontal do container do grid. Os valores possíveis para configurar essa propriedade são:

start: o grid fica alinhado na reta esquerda do container.

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

Resultado:

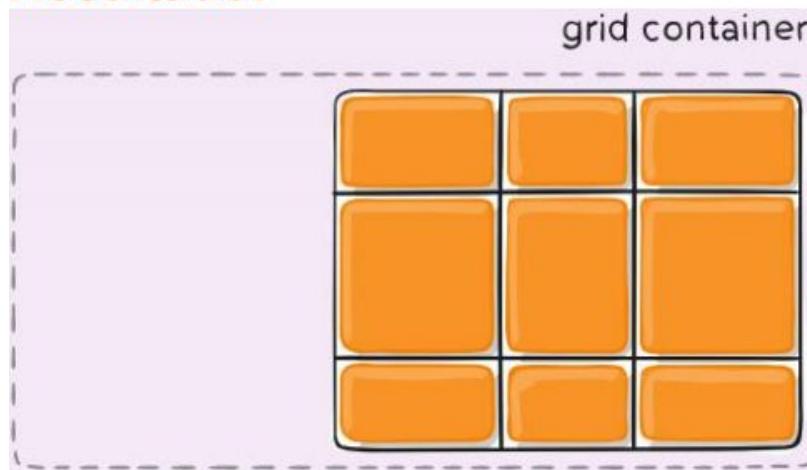


Propriedade justify-content

- **end**: o grid fica alinhado na reta direita do container.

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

Resultado:

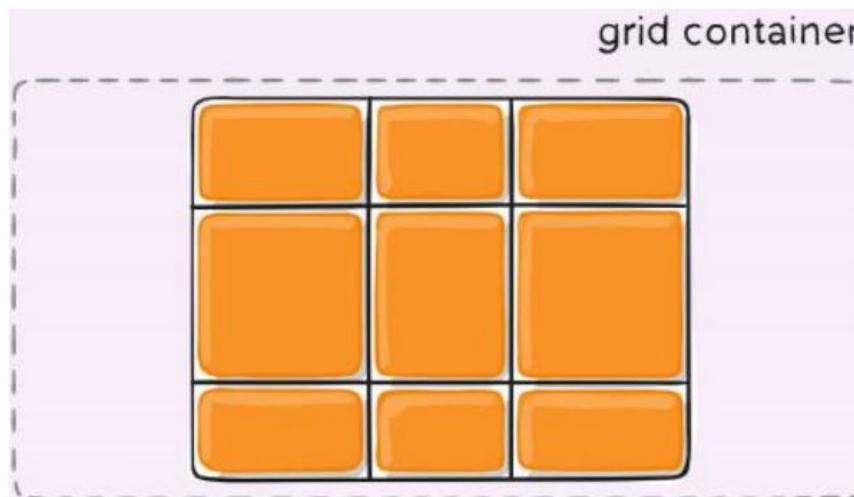


Propriedade justify-content

center: o grid fica centralizado horizontalmente no container.

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

Resultado:



Propriedade justify-content

stretch: o grid preenche todo o espaço do container.

```
.container {  
    justify-content: stretch;  
}
```

Resultado:



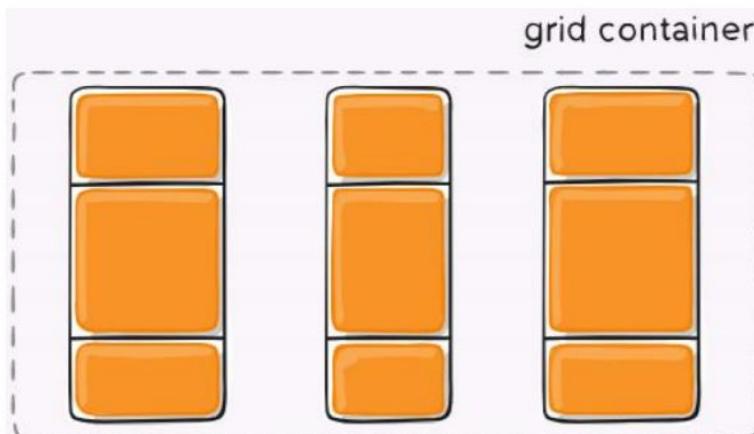
Propriedade justify-content



space-around: coloca uma quantidade uniforme de espaço entre cada item do grid e metade do tamanho do espaço nas extremidades da esquerda e da direita.

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

Resultado:

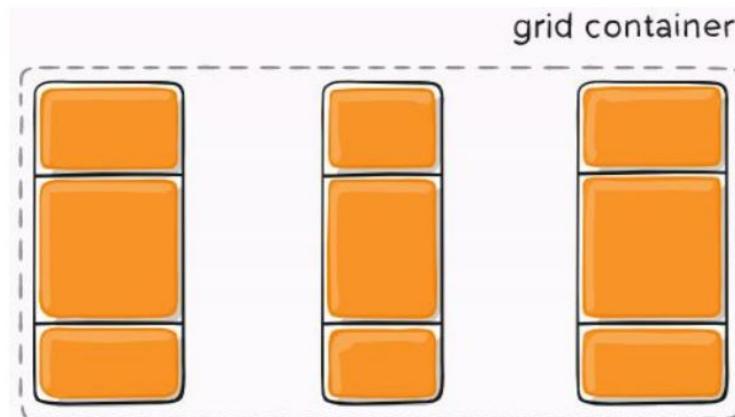


Propriedade justify-content

space-between: coloca uma quantidade uniforme de espaço entre cada item do grid e sem espaço nas extremidades da esquerda e da direita.

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

Resultado:

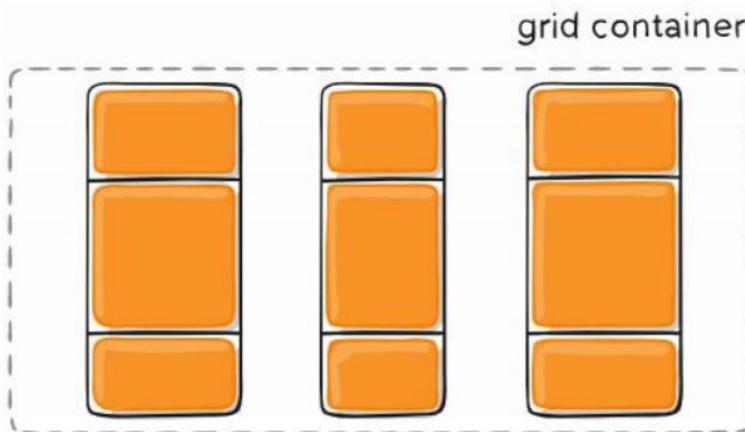


Propriedade justify-content



space-evenly: coloca uma quantidade uniforme de espaço entre cada item do grid, incluindo as extremidades da esquerda e da direita com mesmo tamanho de espaço.

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



IOS – Instituto de
Oportunidade Social

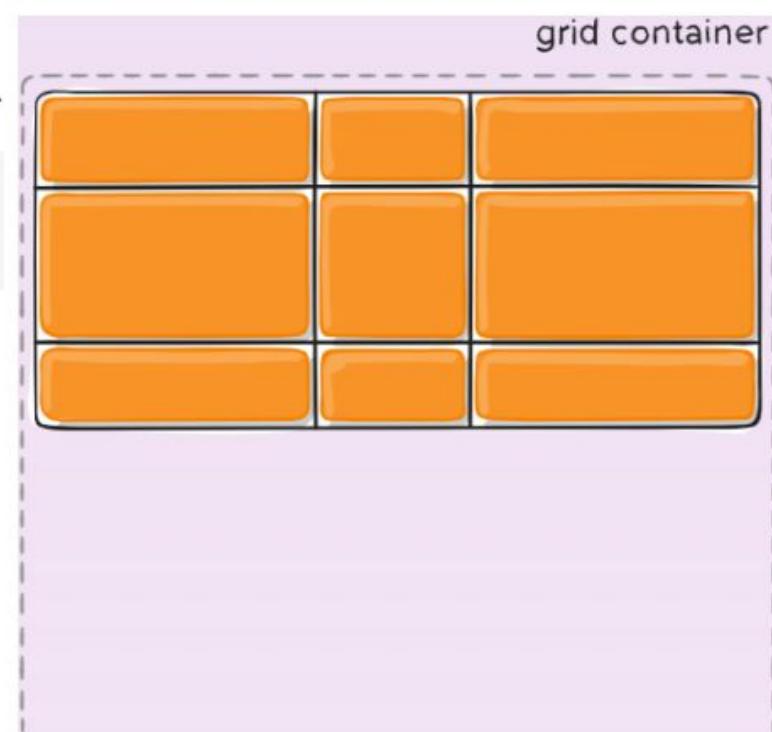
Propriedade align-content



Propriedade align-content

A propriedade align-content alinha o grid completo ao longo do eixo vertical do container do grid. Os valores possíveis para configurar essa propriedade são:

Resultado:



start: o grid fica alinhado na borda superior do container.

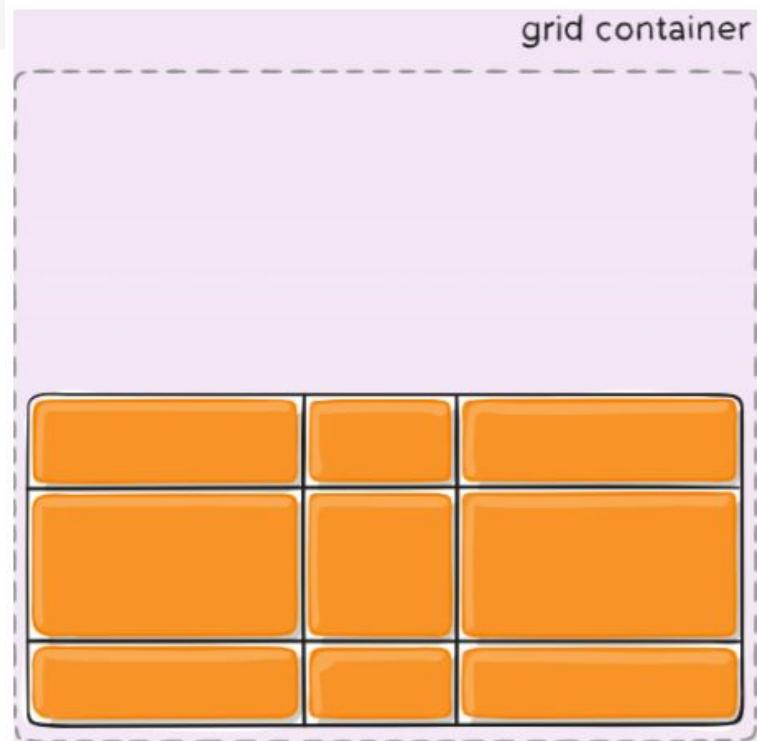
```
.container {  
  align-content: start;  
}
```

Propriedade align-content

end: o grid fica alinhado na borda inferior do container.

```
.container {  
    align-content: end;  
}
```

Resultado:

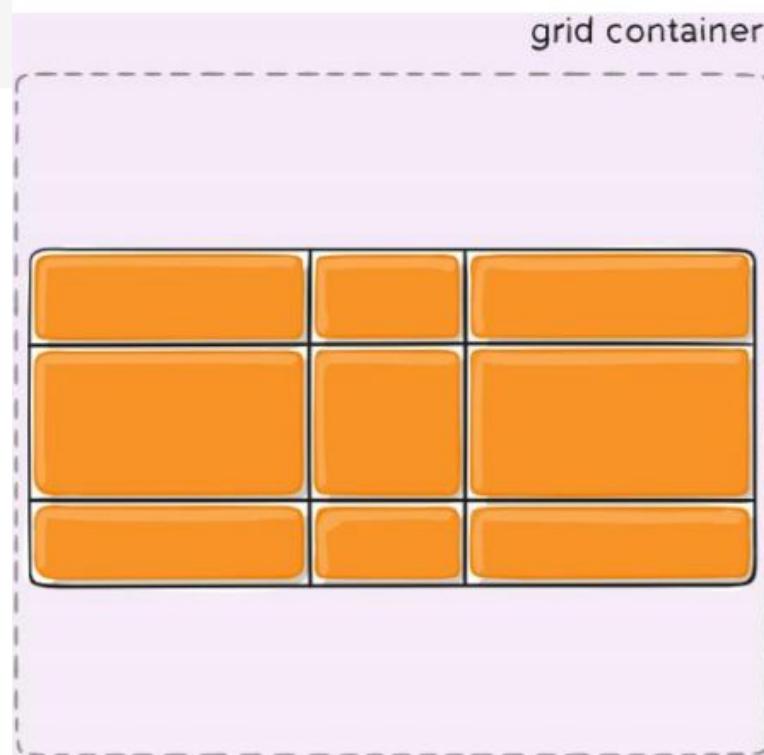


Propriedade align-content

center: o grid fica centralizado verticalmente no container.

```
.container {  
    align-content: center;  
}
```

Resultado:



Propriedade align-content

stretch: o grid preenche todo o espaço do container.

```
.container {  
    align-content: stretch;  
}
```

Resultado:



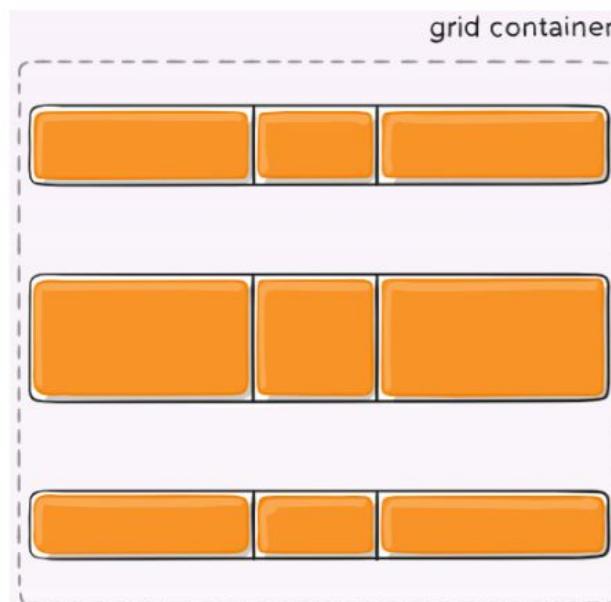
Propriedade align-content



space-around: coloca uma quantidade uniforme de espaço entre cada item do grid e metade do tamanho do espaço nas extremidades superior e inferior.

```
.container {  
    align-content: space-around;  
}
```

Resultado:



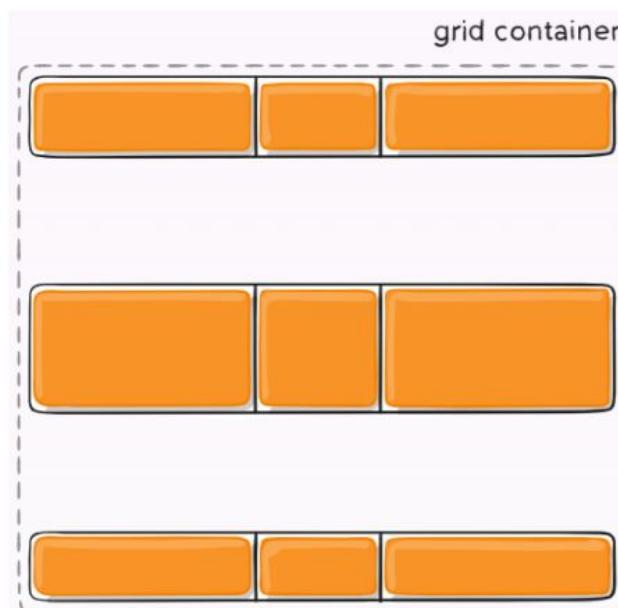
Propriedade align-content



space-between: coloca uma quantidade uniforme de espaço entre cada item do grid e sem espaço nas extremidades superior e inferior.

```
.container {  
    align-content: space-between;  
}
```

Resultados:



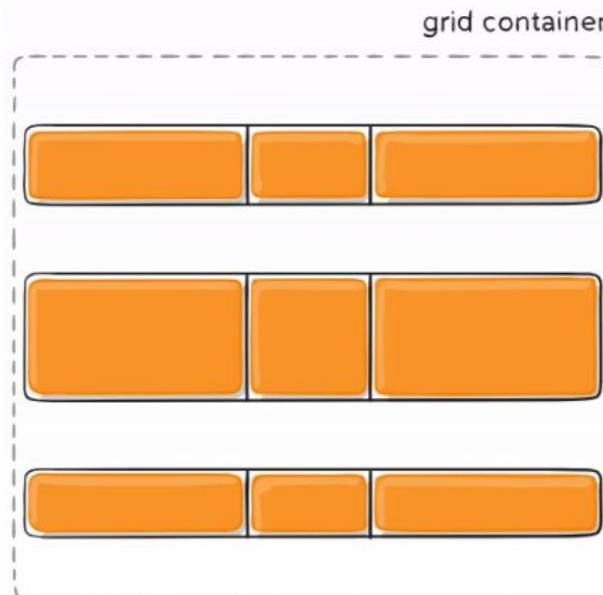
Propriedade align-content



space-evenly: coloca uma quantidade uniforme de espaço entre cada item do grid, incluindo as extremidades superior e inferior com mesmo tamanho de espaço.

```
.container {  
    align-content: space-evenly;  
}
```

Resultado:



IOS – Instituto de
Oportunidade Social

Exercício



Você deve criar um container e nove filhos(itens).

O seu grid deverá conter 4 colunas e 6 linhas sendo divididas da seguinte forma:

- > Na primeira linha coloque dois itens e cada um deve ocupar a mesma quantidade de colunas.
- > Na segunda linha coloque apenas um único item de forma que ele ocupe todas as colunas.
- > Na terceira linha coloque apenas um único item de forma que ele ocupe todas as colunas.

- > Na quarta linha coloque um item que ocupe apenas a segunda e terceira coluna sendo assim a coluna um e quatro devem ficar vazias.
- > Na quinta linha coloque um item que ocupe todas as colunas.
- > Na sexta linha coloque três itens sendo que o primeiro item deve ocupar uma coluna a mais que os outros dois itens.