



Bom dia
Descodificadas!

Grid

Hoje em dia usamos CSS para estilizar layouts muito diferentes do que imaginávamos na época que a web foi criada. De tabelas para divs, de floats para Flexbox, as ferramentas disponíveis para criação de layouts na web evoluíram muito desde os anos 90.

Agora, CSS Grid já é suportado em todos os navegadores mais atuais e podemos dizer com segurança que esse é o método de layout mais poderoso que já foi adicionado ao CSS.

O que é grid?

Grid é uma malha formada pela interseção de um conjunto de linhas horizontais e um conjunto de linhas verticais – um dos conjuntos define colunas e outro linhas.

Dentro de um grid, respeitando-se a configuração criada pelas suas linhas, pode-se inserir elementos da marcação.

O grid é constituído por 12 colunas.

O que é grid?

Em palavras muito simples Grid System (sistema de grade, em português) é usado para dividir uma página em pequenos e espaçados segmentos.



Dimensões fixas ou flexíveis?

Podemos criar grids com dimensões fixas – por exemplo: definindo dimensões em pixels.

Ou criar grids com dimensões flexíveis definindo-as com uso de porcentagem ou da nova unidade CSS fr criada para esse propósito.

Posicionamento de itens

Podemos posicionar com precisão itens de uma página usando o número que define uma linha do grid, usando nomes ou ainda fazendo referência a uma determinada região do grid.

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 2fr;  
}
```

```
.container {  
  display: grid;  
  grid-template-columns: [sidebar] 1fr [content] 2fr;  
}
```

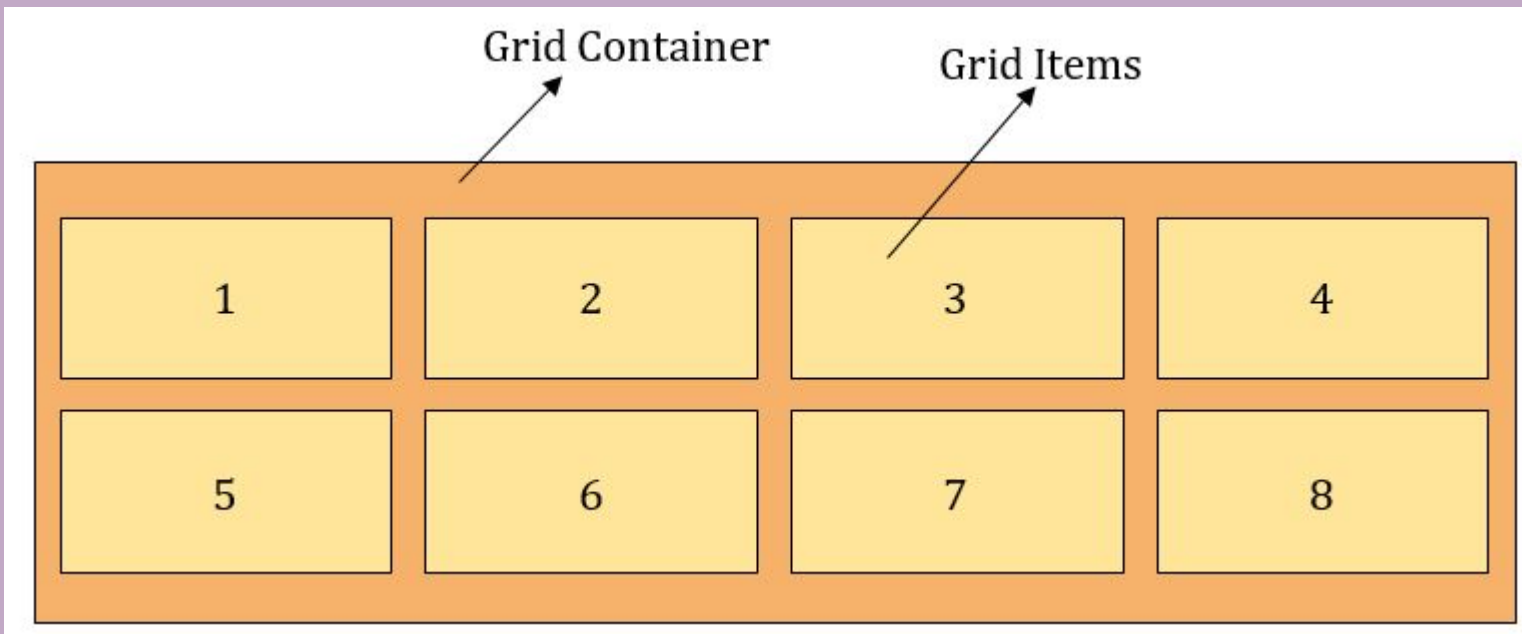

Posicionamento de itens

CSS Grid Layout é uma poderosa especificação que se for combinada com outras partes do CSS, tal como flexbox, possibilita a criação de layouts que até então eram impossíveis de serem criados com CSS.

Grid - Estrutura

Grid container é o elemento que envolve sua estrutura.

Em um contêiner grid, todos os seus filhos diretos se tornam itens grid. Com um contêiner grid em mãos podemos começar a criar layouts usando colunas e linhas.



Container

Grid Container

```
display: grid;  
grid-template-columns: 2fr 5fr 2fr;  
grid-gap: 1rem;
```

Grid Items 👉 📌 🖐️ 👁️

```
background: lightsalmon;
```



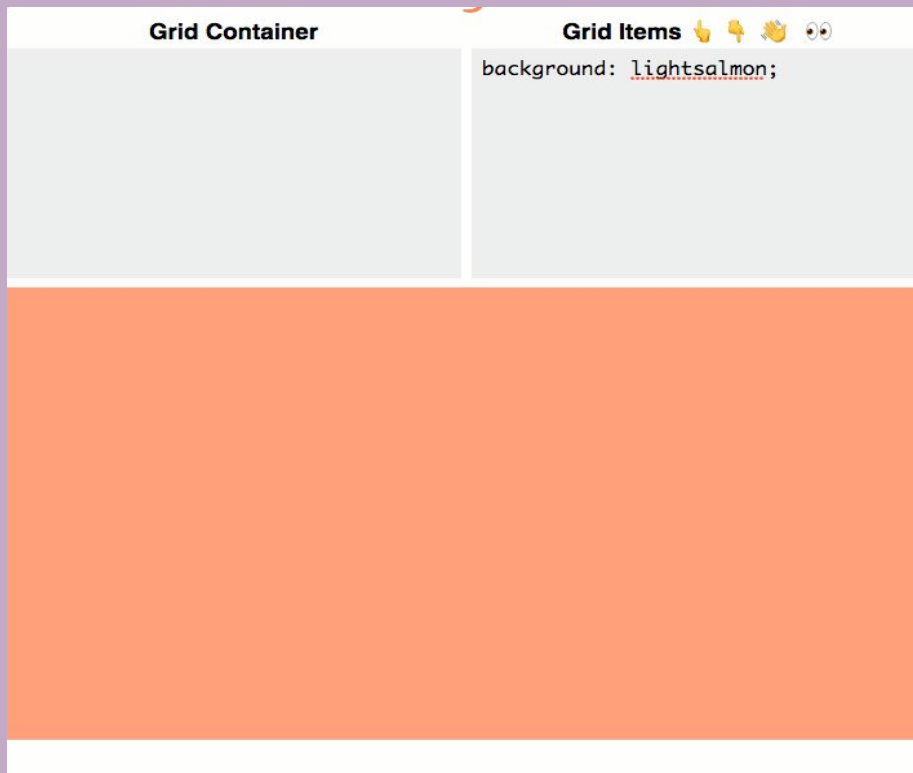
Container

As grades CSS se comportam de maneira semelhante ao Flexbox, pois há um relacionamento pai-filho assim:

```
<div class="grid-container"> | Parent  
  <div class="grid-item"></div>  
  <div class="grid-item"></div>  
  <div class="grid-item"></div>  
  <div class="grid-item"></div> | Children  
</div>
```

Container

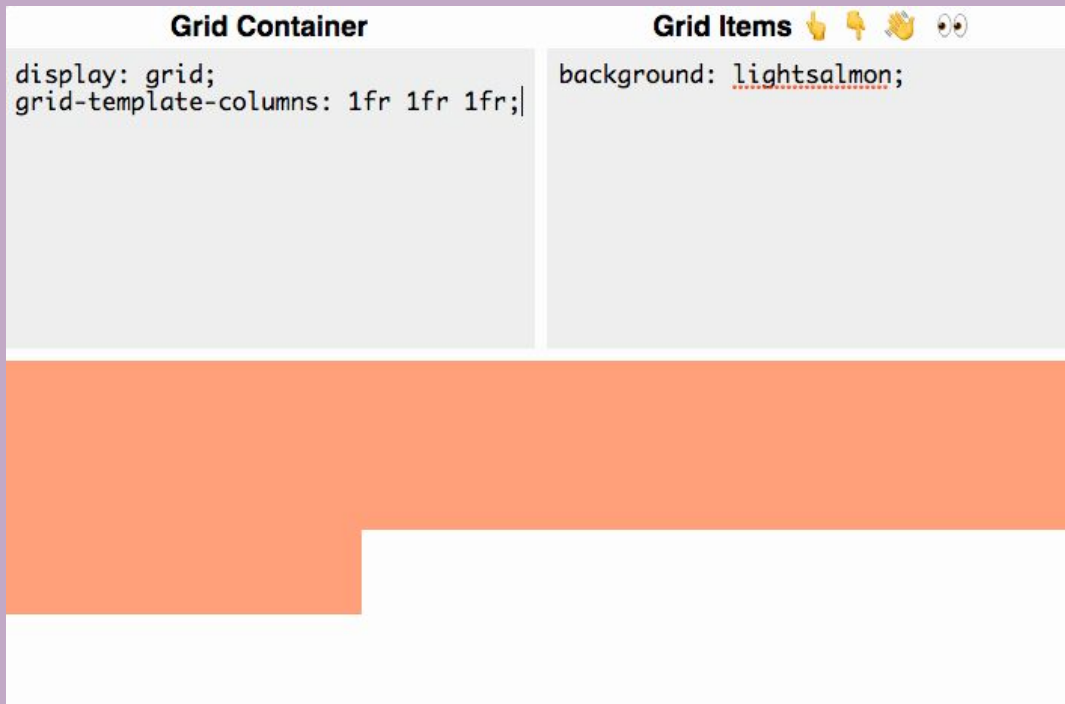
Portanto, o primeiro passo que precisamos fazer é definir a propriedade `display:grid;` do nosso elemento pai.



Nos bastidores nosso div pai agora é um [Grid Container](#), o que significa que todos os seus filhos diretos são [Grid Items](#).

Container

Portanto, o primeiro passo que precisamos fazer é definir a propriedade `display:grid;` do nosso elemento pai.



A seguir, vamos definir uma grade de 3 colunas com :

grid-template-columns: 1fr 1fr 1fr;
grid-gap: 5px;(tornará as divisões mais proeminentes).

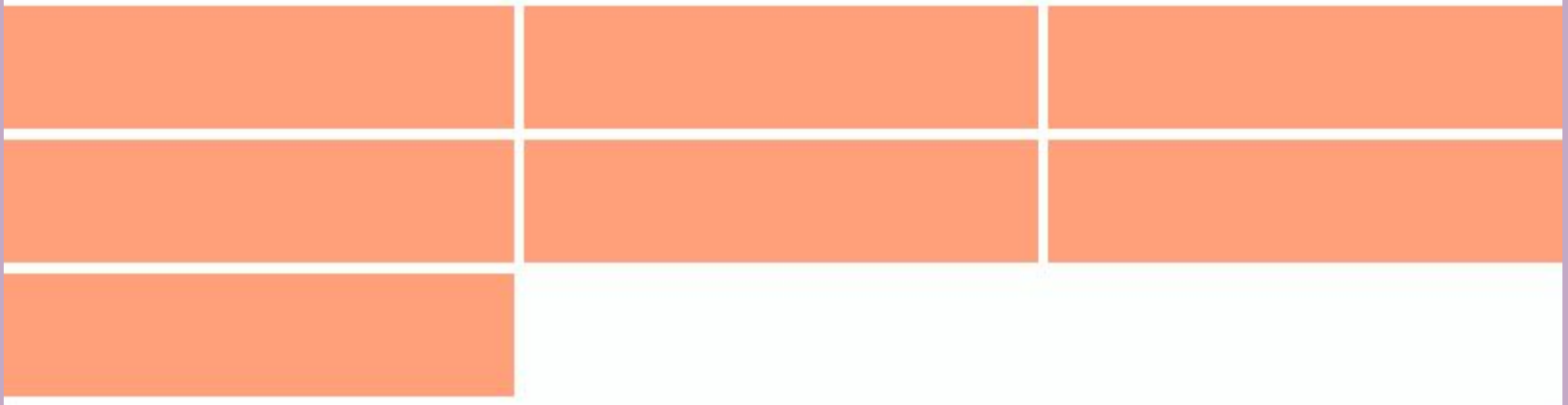
Distribuição de colunas

Grid Container

```
display: grid;  
grid-template-columns: 1fr 1fr 1fr;  
grid-gap: 5px;
```

Grid Items 🙌 🙌 🙌 👀

```
background: lightsalmon;
```



Mesclando columnas

Grid Container

```
display: grid;  
grid-template-columns: 1fr 2fr 1fr;  
grid-gap: 5px;
```

Grid Items 👉 📌 🙌 👁

```
background: lightsalmon;
```



Mesclando/Alargando Linhas

Grid Container

```
display: grid;  
grid-template-columns: 1fr 2fr 1fr;  
grid-gap: 5px;
```

Grid Items 👉 📌 🙌 👀

```
background: lightsalmon;
```

```
grid-column: 1 / 4;
```

|



Mesclando/Alargando Linhas

Grid Container

```
display: grid;  
grid-template-columns: 1fr 2fr 1fr;  
grid-gap: 5px;
```

Grid Items 👉 📌 🤝 👁

```
background: lightsalmon;
```

```
grid-column: 1 / 4;
```

```
height: 300px;  
grid-row: 2 / 4;
```

|

Criando um rodapé

Grid Container

```
display: grid;  
grid-template-columns: 1fr 2fr 1fr;  
grid-gap: 5px;
```

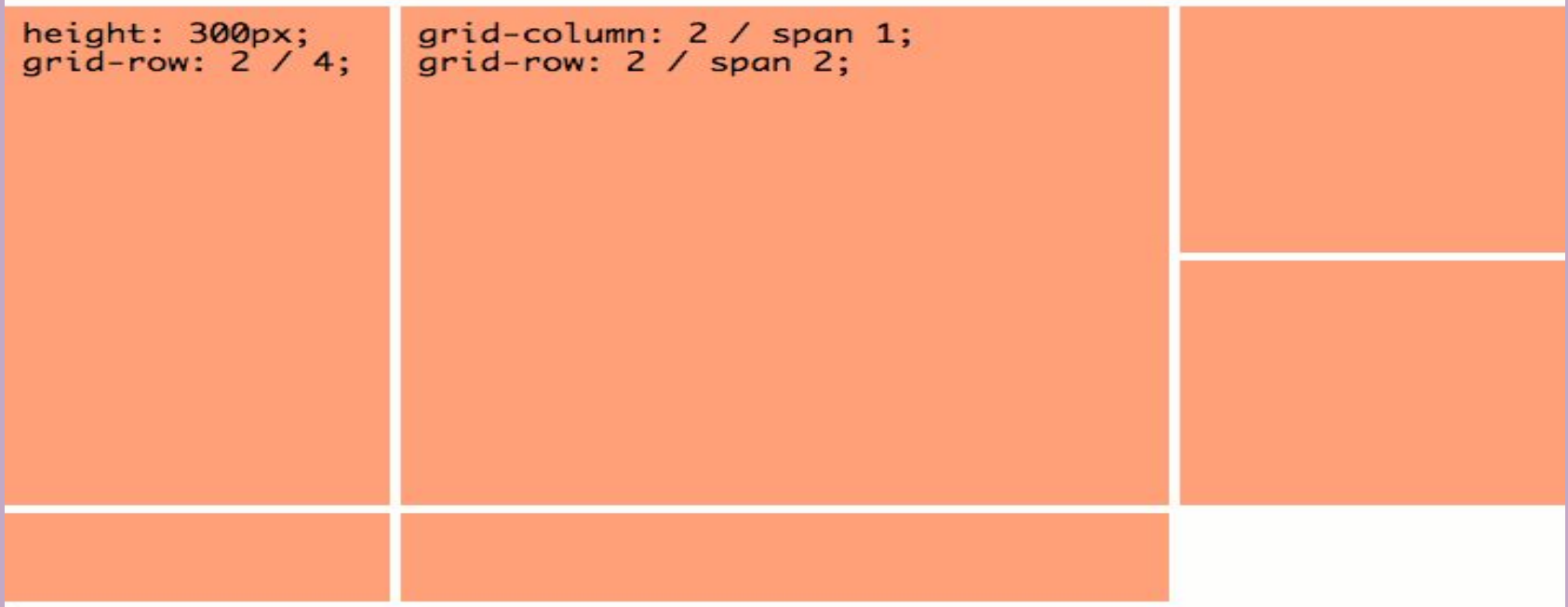
Grid Items 👉 📌 🖱️ 👁️

```
background: lightsalmon;
```

```
grid-column: 1 / 4;
```

```
height: 300px;  
grid-row: 2 / 4;
```

```
grid-column: 2 / span 1;  
grid-row: 2 / span 2;
```



Alinhamento item inferior

Grid Container

```
display: grid;  
grid-template-columns: 1fr 2fr 1fr;  
grid-gap: 5px;
```

Grid Items



```
background: lightsalmon;
```

```
grid-column: 1 / 4;
```

```
height: 300px;  
grid-row: 2 / 4;
```

```
grid-column: 2 / span 1;  
grid-row: 2 / span 2;
```

```
|
```

```
grid-column: 1 / 4;
```

Alinhamento item centro

Grid Container

```
display: grid;  
grid-template-columns: 1fr 2fr 1fr;  
grid-gap: 5px;
```

Grid Items



```
background: lightsalmon;
```

```
grid-column: 1 / 4;
```

```
height: 300px;  
grid-row: 2 / 4;
```

```
grid-column: 2 / span 1;  
grid-row: 2 / span 2;
```

```
grid-row: 2 / 4;  
height: 150px;  
align-self: end;
```

```
grid-column: 1 / 4;
```



Você encontrará
todas as
orientações e
conteúdos
respectivos a
cada semana na
nossa
plataforma!

Bons estudos!!

< descodificadas />

Sigam nossas redes
sociais!!!

