



Co-funded by the  
Erasmus+ Programme  
of the European Union



## Resultado 3.1.: Elaboração dos Materiais de Formação Code4SP

WP3: Materiais de Formação Code4SP

Preparado por: Social Hackers Academy



CITIZENS  
IN POWER



Center for Social  
Innovation



Co-funded by the  
Erasmus+ Programme  
of the European Union





# Grelhas no CSS

Co-funded by the  
Erasmus+ Programme  
of the European Union



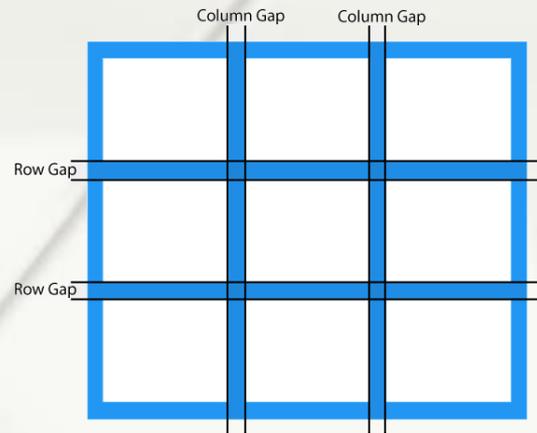
# Grelhas no CSS

- O **layout de grelhas no CSS** ou apenas **grelhas do CSS** cria *layouts* de design da Web reativos, complexos e com mais facilidade e consistência nos navegadores. Existem outras formas de controlar métodos de *layout* de página da Web, como tabelas, modelo de caixa e a *flexbox CSS*.
- Para começar, é importante definir um elemento de um *container* como grelha, utilizando a função **display: grid**, definir o tamanho das colunas e filas usando **grid-template-columns** e **grid-template-rows**, colocando os seus elementos “filhos” na coluna, através das propriedades **grid-column** e **grid-row**.



## O que são as grelhas no CSS?

- As grelhas do CSS são baseadas na perspetiva de filas e colunas. Veremos como funcionam e que ordens devem ser dadas.
- Tomemos o exemplo do próximo slide e construamos uma grelha, de forma a ficarmos a conhecer diferentes formas de o fazer.



## A perspetiva das filas e colunas

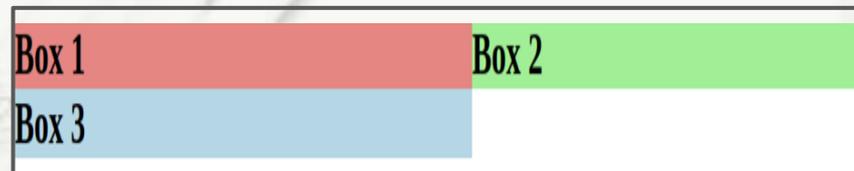
- No exemplo da imagem à direita, temos um design baseado em **colunas** e **filas**. Neste caso, temos apenas duas colunas e duas filas, e podemos alcançar este resultado ao colocar as caixas 1 e 2 no mesmo *div*, para, desta forma, dar-lhe uma largura absoluta. Para a caixa n.º 3, esta será traçada fora do *div* anterior, e vai responder aos mesmos comandos CSS. Ao usar o **CSS grid**, este processo torna-se mais simples.



## Perspetivas de filas e colunas

- Ao usar os comandos descritos na coluna da esquerda, conseguiremos o resultado presente na figura da direita. Em relação ao comando **grid-template –columns**, este pode suportar mais de um valor com base em quantas colunas são desejadas. Neste caso, apenas descrevemos dois valores, gerando o resultado de duas colunas com 500px de largura, mesmo havendo espaço extra à esquerda.

```
.container {  
  display: grid;  
  grid-template-columns: 500px 500px;  
}
```



## Perspetivas de filas e colunas

- Reparamos também na existência do comando **grid-template-rows**, e reparamos no resultado, no qual vemos que a altura é superior ao anterior, pelo facto de termos dado **300px** a cada fila.

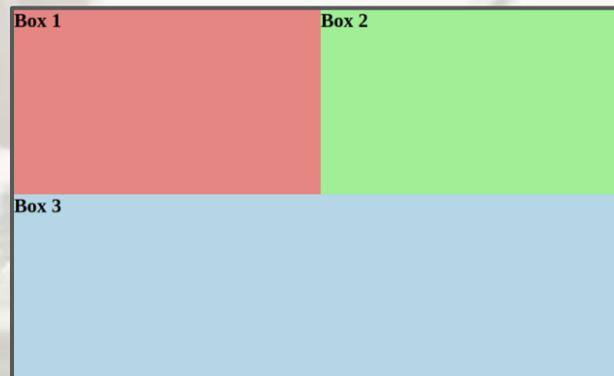
```
.container {  
  display: grid;  
  grid-template-columns: 500px 500px;  
  grid-template-rows: 300px 300px;  
}
```



## Perspetivas de filas e colunas

- Podemos agora desenhar a caixa 3 ao longo do espaço vazio que se encontra debaixo da caixa 2 ao utilizar apenas a propriedade **grid-column-start/end** . Ao usar este comando, escolhemos o ponto de partida e de chegada e... está feito!

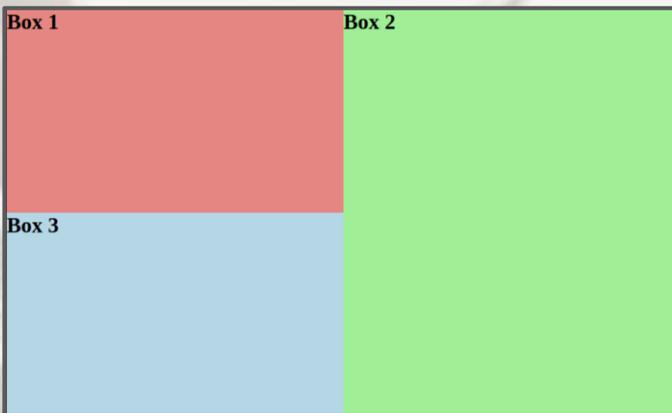
```
.container {  
  display: grid;  
  grid-template-columns: 500px 500px;  
  grid-template-rows: 300px 300px;  
}  
  
.box3 {  
  grid-column-start: 1;  
  grid-column-end: 3;  
}
```



## Perspetivas de filas e colunas

- Podemos aplicar o mesmo princípio para **filas**, usando o comando **grid-row**, seguindo-se a posição exata que pretendemos, como se vê no exemplo abaixo.

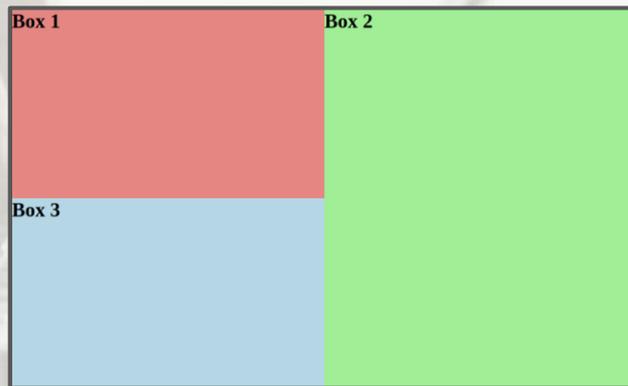
```
.container {  
  display: grid;  
  grid-template-columns: 500px 500px;  
  grid-template-rows: 300px 300px;  
}  
  
.box1 {  
  grid-row: 1/2;  
}  
  
.box2 {  
  grid-row: 1/3;  
}
```



## Perspetivas de filas e colunas

- Para tornar este processo ainda mais simples, podemos usar o **grid-template-areas**, que nos oferece o mesmo resultado ao declarar os nomes das áreas nas suas linhas de código “filhas”, aplicando-as de uma forma visual no *container*.

```
.container {  
  display: grid;  
  grid-template-columns: 500px 500px;  
  grid-template-rows: 300px 300px;  
  grid-template-areas:  
    "box1 box2"  
    "box3 box2";  
}  
  
.box1 {  
  grid-area: box1;  
}  
  
.box2 {  
  grid-area: box2;  
}  
  
.box3 {  
  grid-area: box3;  
}
```





# Perspetivas de filas e colunas

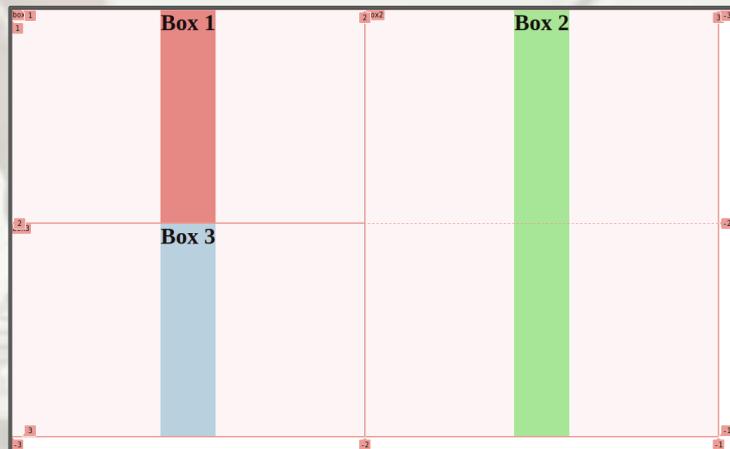
- Podemos também alterar a posição de todas as “filhas”, ao usar o **justify-items**. Depois, podemos escolher a sua posição: ao centro, no início, no fim, podendo também esticá-las (**center, start, end** e **stretch**).

```
.container {
  display: grid;
  grid-template-columns: 500px 500px;
  grid-template-rows: 300px 300px;
  grid-template-areas:
    "box1 box2"
    "box3 box2";
  justify-items: center;
}

.box1 {
  grid-area: box1;
}

.box2 {
  grid-area: box2;
}

.box3 {
  grid-area: box3;
}
```





## Perspetivas de filas e colunas

- Para um elemento específico, podemos usar o comando anterior, aplicando somente o **align-self** ou **justify-self** na “filha” desejada.

```
.container {  
  display: grid;  
  grid-template-columns: 500px 500px;  
  grid-template-rows: 300px 300px;  
  grid-template-areas:  
    "box1 box2"  
    "box3 box2";  
}  
  
.box1 {  
  grid-area: box1;  
  align-self: center;  
}  
  
.box2 {  
  grid-area: box2;  
  justify-self: center;  
}  
  
.box3 {  
  grid-area: box3;  
}
```

