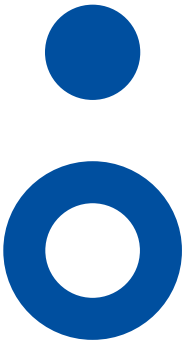
critec

creative agency

this is your full article

you got it from **critec.pt**

critec



we hope you find it useful and beneficial. enjoy.

critec

Porque é que ainda não estás a usar css grid?

escrito por
Filipe Nascimento

16 Maio 2019

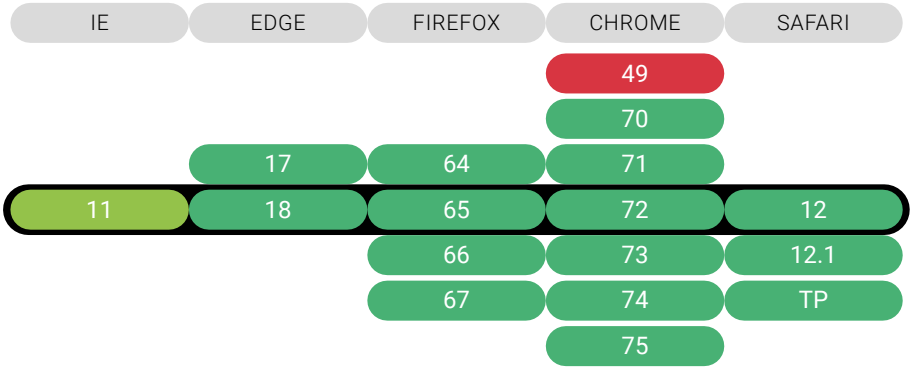
critec

Porque é que ainda não estás a usar css grid?

O sistema de grelha está disponível há séculos, dos primeiros jornais, cartazes, revistas até layouts de websites. Desde sempre replicar o que está no papel para website foi uma tarefa difícil, tabelas, floats, posições absolutas, alinhamentos e elementos dentro de elementos só para obter o resultado final.

Mesmo com frameworks e flex-box, as dificuldades foram surgindo em designs mais complexos, foi nesse momento que nós encontramos esta propriedade de css3 chamada css grid system.

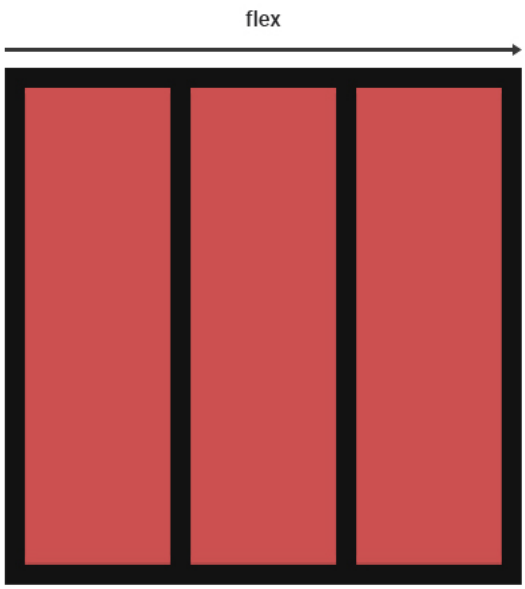
O css grid é suportado pelos browsers mais utilizados e começam a surgir nos websites mais modernos.



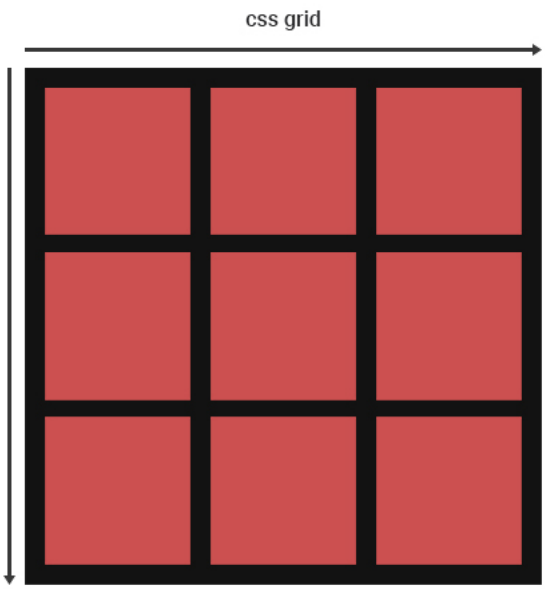
Uso Global dos browsers
85.08% + 2.7% = 87.78%

Fonte - caniuse.com

Então, de forma simples, o css grid é uma propriedade que permite definir um certo número de colunas e linhas, e posicionar cada child na célula que se pretende. Muito simples, certo? Como é um sistema de 2 dimensões que permite posicionar e alinhar os elementos, torna o código html mais simples e com menos "nesting".



Enquanto flex-box e a maior parte dos frameworks, com base em flex, são uni-dimensional.



O css grid é bi-dimensional, o que permite posicionar um elementos horizontalmente e verticalmente num lugar preciso.

características

- **Tamanho**

As colunas e as linhas podem ser definidas por tamanhos fixos (ex: px) ou por tamanhos flexíveis (ex: percentagem) de forma a criar o layout desejado. O css grid também vem introduzir uma nova unidade de medida que é o fr, que significa fração. Por exemplo se definir que uma linha tem 1fr e a seguinte tem 2fr, sabemos que a segunda linha terá sempre o dobro do tamanho da primeira.

Posicionamento

O css grid permite posicionar um elemento num lugar preciso. Pode-se usar número de linha, nomes ou alvejar uma área da grid. A grid também detecta os elementos child e posiciona-os automaticamente numa célula disponível da grid, se esse elemento não tiver uma posição especificada.

Autoconsciente

O elementos que tem display: grid é autoconsciente dos childs elements diretos, se existe uma grid de 1 coluna por 1 linha, mas existem 2 childs, irá automaticamente adicionar mais uma linha automaticamente para adicionar o 2º child. Esta característica torna-se muito útil se um site for alimentado por conteúdo dinâmico.

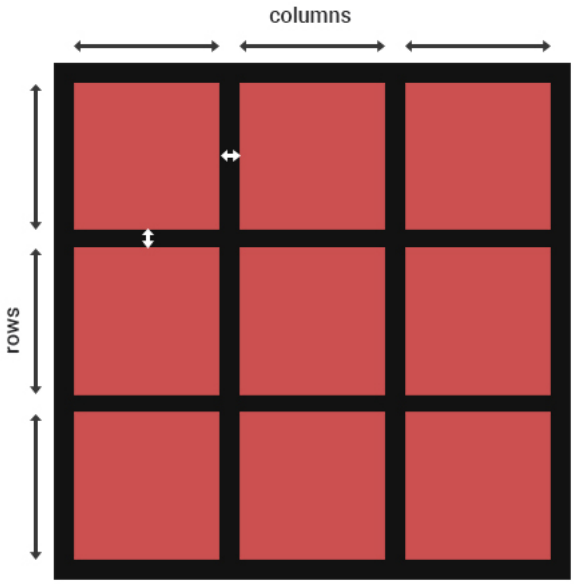
Alinhamento

O css grid herdou as propriedades de alinhamento do flex, tudo pode ser alinhado dentro da célula horizontalmente e verticalmente. É possível alinhar todos os elementos juntos ou cada um em separado.

Sobreposição

O facto da grid ser dividida por células não significa que o conteúdo não possa sobrepor. É possível posicionar mais que um elemento na mesma célula e usar a propriedade z-index para definir a ordem destes.

colunas, linhas e intervalos

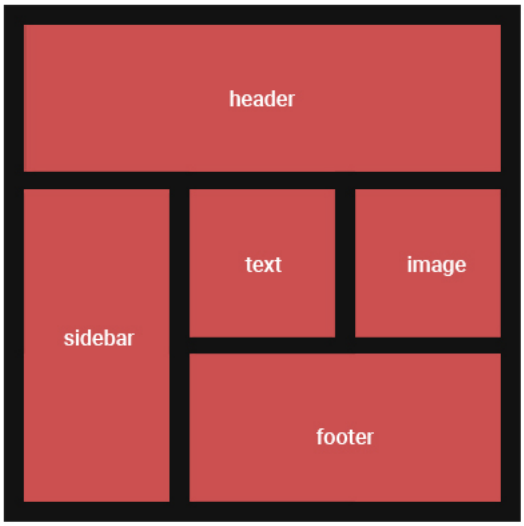


Pode-se definir o número de colunas e de linhas, assim como os intervalos entre colunas e linhas. Exemplo:

```
body{
      display: grid;
      grid-template-columns: repeat (3, 1fr);
      grid-template-rows: repeat (3, 1fr);
      grid-column-gap: 15px;
      grid-row-gap: 15px;
}
```

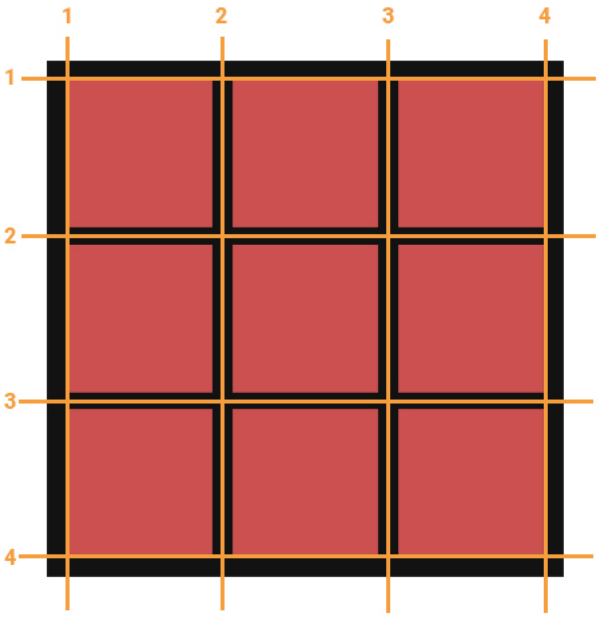
posicionar elementos na grid

Como referido anteriormente, pode-se inserir os elementos numa ou mais células. Considera o layout seguinte, usando a grid criada anteriormente:




```
body{
      display: grid;
      grid-template-columns: repeat (3, 1fr);
      grid-template-rows: repeat (3, 1fr);
      grid-column-gap: 15px;
      grid-row-gap: 15px;
}

header{
      grid-column: 1/4;
      grid-row: 1/2;
}
```

A grid-column: 1/4; Significa que o header começa no ponto 1 e termina no ponto 4.
As linhas seguem a mesma lógica.




```
sidebar{
      grid-column: 1/2;
      grid-row: 1/4;
}

text{
      grid-column: 2/3;
      grid-row: 2/3;
}

image{
      grid-column: 3/4;
      grid-row: 2/3;
}

footer{
      grid-column: 2/4;
      grid-row: 3/4;
}
```

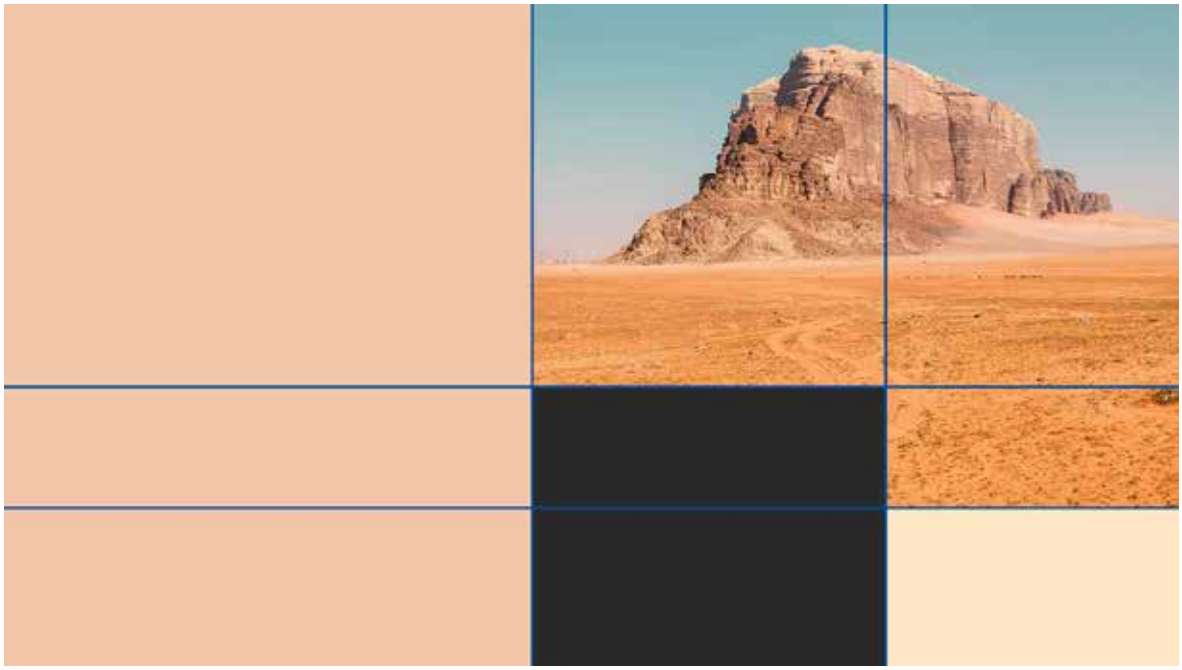
Os elementos podem também ter um nome único e alocados na grid pelo nome. Exemplo:

```
header{
      grid-area: header;
}

sidebar{
      grid-area: sidebar;
}

text{
      grid-area: text;
}

image{
      grid-area: image;
}

footer{
      grid-area: footer;
}

body{
      display: grid;
      grid-template-columns: repeat (3, 1fr);
      grid-template-rows: repeat (3, 1fr);
      grid-column-gap: 15px;
      grid-row-gap: 15px;
      grid-template-areas:
      “header header header”
      “sidebar text image”
      “sidebar footer footer”
}
```

Esta é a forma mais visual de codigo css.

do photoshop para o css

Mais importante que saber sobre os conceitos do css grid é saber a sua aplicação num caso real. Considera a imagem de exemplo seguinte:



Primeiro define-se a grid exterior, a que estará presente no body. Pode-se usar percentagem, unidades fixas ou auto para preencher as partes em falta.

```
body{
     display: grid;
     height: 100vh; // coloca a altura do body igual à altura
da janela do browser
     grid-template-columns: 45% 30% auto;
     grid-template-rows: 450px 130px auto;
}

.main{
     grid-column: 1/2;
     grid-row: 1/4;
}
```

```
img{
      grid-column: 2/4;
      grid-row: 1/3;
}

.dark-div{
      grid-column: 2/3;
      grid-row: 2/4;
      z-index: 2; // faz com que este elemento fique por cima
do outro elemento na mesma célula.
}

.light-div{
      grid-column: 3/4;
      grid-row: 3/4;
}
```

Depois de estar tudo definido na grid exterior, adiciona outra grid para cada child, e novamente caso necessário.

Então o .main é um elemento da grid com a sua própria grid dentro.

```
.main{
     grid-column: 1/2;
     grid-row: 1/4;
     display: grid;
     margin: 0 70px; // poderia ser usado colunas para criar os espaçamentos, mas visto que não haverá elementos não excedem as margens é mais simple assim
     grid-template-rows: 80px 200px repeat(3, auto); // forma mais rápida para repetir a mesma medida. a linha em auto irá ganhar a altura do childs, e a última linha ficará com o espaço em sobra
}

.brand{
     grid-row: 1/2;
     align-self: end; // posiciona o elemento na parte de baixo da linha
}

.title{
     grid-row: 3/4;
}

.text{
     grid-row: 3/4;
}
```

conclusões finais

Apesar do css grid estar disponível já a alguns anos, apenas recentemente a critec começou a usar nos seu sites, como web developer, sinto que foi uma grande alteração, para melhor, permite trabalhar mais rápido, com código mais simples e fácil de ler, o que nos permite ir mais longe na construção de layout mais alternativos e inexplorados.

Para aprenderes mais sobre css grid, podes explorar estes artigos:

Css tricks
https://css-tricks.com/snippets/css/complete-guide-grid/

MDN web docs
https://developer.mozilla.org/pt-PT/docs/Web/CSS/layout_de_grelha_css

W3schools
https://www.w3schools.com/css/css_grid.asp

find more in

critec.pt/blog

Critec, Lda.
Travessa da Gândara
3750 – 727 Recardães
Águeda – Portugal

Talk to us
critec@critec.pt
+351 234 100 049
+351 934 786 974

Social Media
@critec