



Display 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." - MDN Web Docs

Dimensões fixas ou flexíveis:

É possível criar grids com dimensões fixas, definindo o tamanho que cada divisão irá ocupar em pixels. Quando queremos fazer grids com dimensões flexíveis utilizamos porcentagem ou de uma medida especial com sintaxe de 'fr' criada justamente para este propósito

Posicionamento de itens:

É possível posicionar itens com precisão dentro de uma página a partir do uso de grid, pois, suas propriedades permitem que os itens sejam realocados conforme necessidade e gosto do desenvolvedor

Criação de grids extras:

É possível criar grids dentro de grids, e grids extras que influenciam no posicionamento de elementos dentro da página, criando dinamicidade e auxiliando na responsividade em diferentes aparelhos

Grid Container:

Aqui começa nosso uso da propriedade Grid, vamos supor que tenhamos o seguinte HTML:

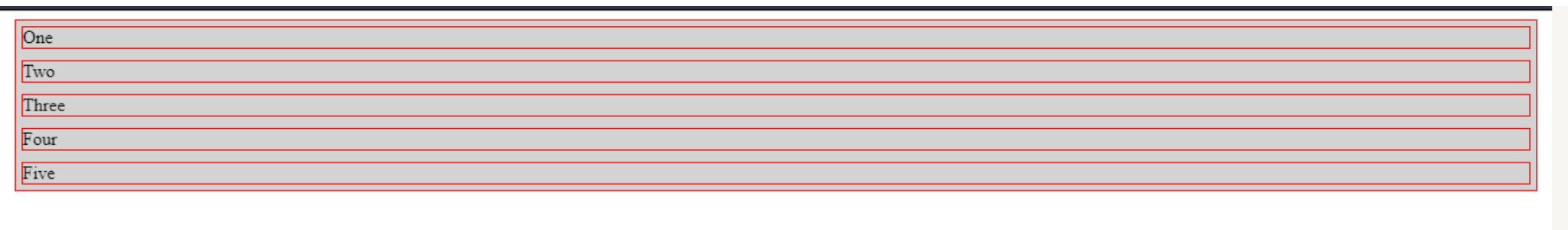
```
<div class="divider">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
  <div>Four</div>
  <div>Five</div>
</div>
```

Uma vez tendo feito nosso HTML, podemos começar a trabalhar no nosso CSS:

```
div {
  background-color: lightgrey;
  margin: 5px;
  border: 1px solid red;
}

.divider {
  display: grid;
}
```

E teremos como resultado algo parecido com isso:

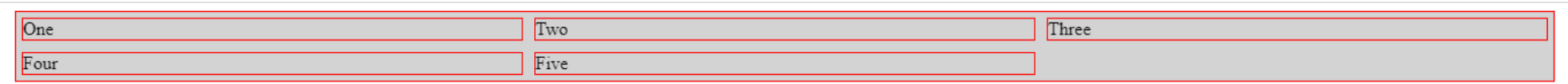


Grid Tracks:

Uma maneira bem comum de fazer divisões com a propriedade grid é fazer um rearranjo de divisões utilizando de 'grid-template-columns' e 'grid-template-rows', o primeiro irá arrumar os elemento em colunas e o segundo em linhas, vamos ver o funcionamento de ambos, (Ps.: Tenha em mente que estamos utilizando o mesmo HTML e css já editados em Grid Container):

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

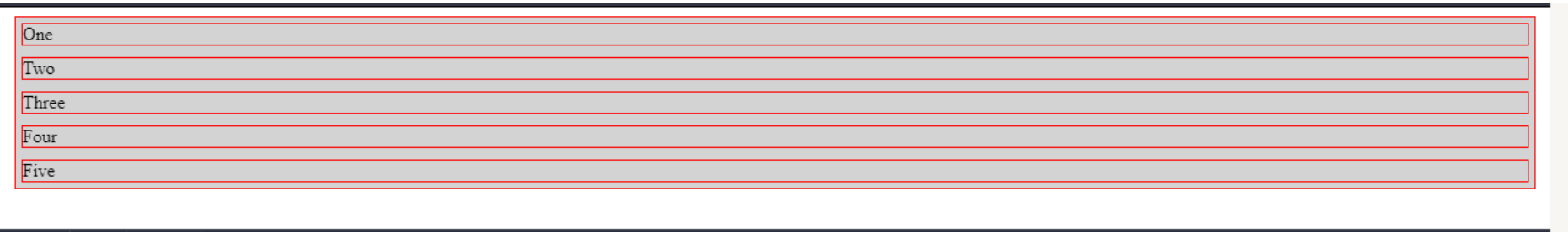
Teremos como resultado então algo deste tipo:



Agora iremos utilizar 'grid-template-rows':

```
.divider {
  display: grid;
  grid-template-rows: 200px 1fr 200px;
}
```

Teremos novamente algo parecido com a primeira imagem:

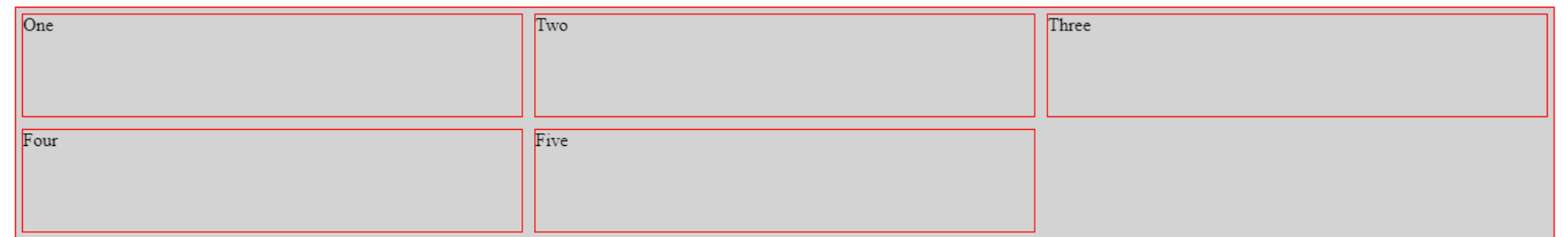


Propriedades MinMax():

Dentro de display grid temos algumas funcionalidades para determinar o tamanho de cada divisão, podemos utilizar minmax para definir o tamanho total de cada elemento, como segue no exemplo:

```
.divider {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-auto-rows: minmax(100px, auto);
}
```

Teremos como resultado algo como:



Posição de itens contra linhas:

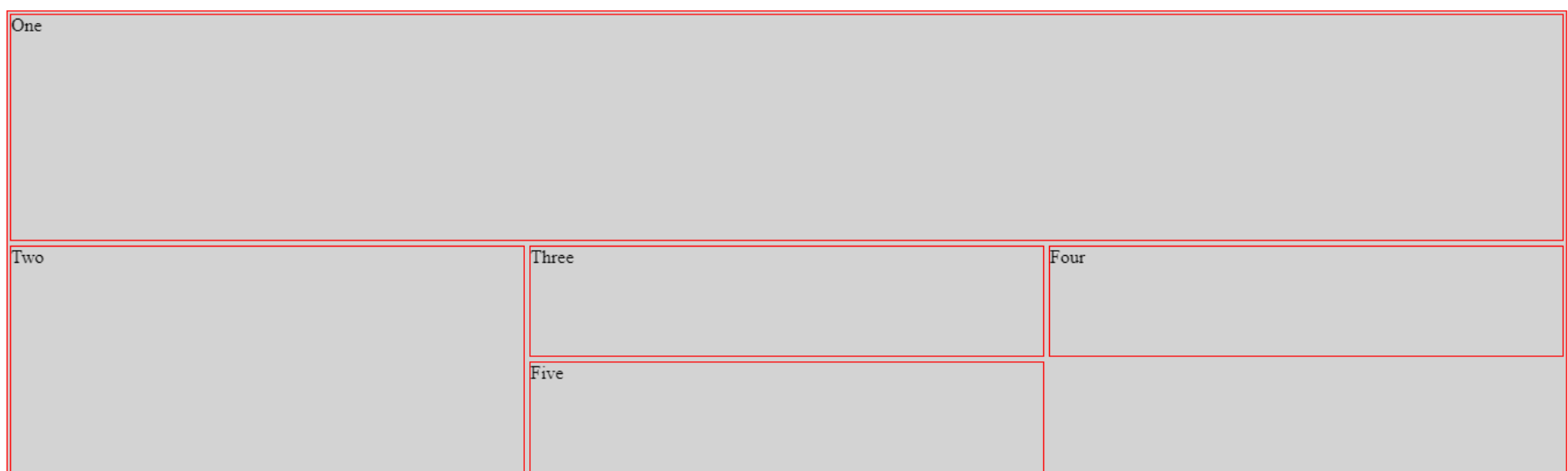
Vamos agora ver uma outra propriedade, que posiciona itens como linhas, para isto iremos utilizar o seguinte HTML:

```
<div class="divider">
  <div class="div1">One</div>
  <div class="div2">Two</div>
  <div class="div3">Three</div>
  <div class="div4">Four</div>
  <div class="div5">Five</div>
</div>
```

Isso feito podemos agora editar nosso CSS

```
.divider {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-auto-rows: 100px;
}
.div1 {
  grid-column-start: 1;
  grid-column-end: 4;
  grid-row-start: 1;
  grid-row-end: 3;
}
.div2 {
  grid-column-start: 1;
  grid-row-start: 3;
  grid-row-end: 5;
}
```

Teremos então a seguinte estrutura:

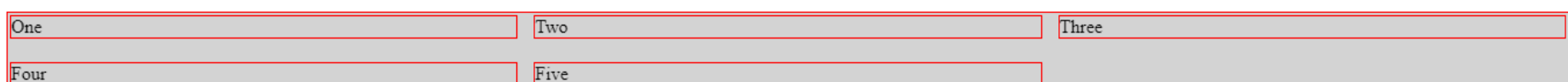


Gutters:

Caso queiramos criar um espaçamento entre divisões podemos usar gutters, os mais comuns são 'grid-column-gap', 'grid-row-gap' ou resumidamente 'grid-gap', vamos a um exemplo:

```
.divider {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-column-gap: 10px;
  grid-row-gap: 1em;
}
```

Teremos como resultado o seguinte:

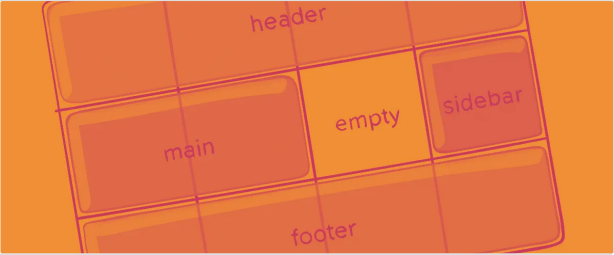


Para maiores informações e aprofundamentos, sugerimos que veja os links que seguem:

A Complete Guide to Grid | CSS-Tricks


CSS Grid Layout is the most powerful layout system available in CSS. It is a 2-dimensional system, meaning it can handle both columns and rows, unlike flexbox which is largely a 1-dimensional system. You work with Grid Layout by applying CSS rules both to a parent element (which becomes the Grid Container) and to that

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



CSS Grid Layout

Try it Yourself " The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning. The grid properties are supported in all modern browsers. A grid layout consists of a parent element, with one or more child elements.

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



Conceitos básicos de Grid Layout

CSS Grid Layout introduz um sistema bi-dimensional de grid (literalmente "grades") para CSS. Grids podem ser usados para o design de layouts de grandes seções de uma webpage, assim como de pequenos elementos de interface. Esse artigo apresenta o CSS Grid Layout e a terminologia que é parte da especificação

 https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout

