

CSS

Cascading Style Sheets

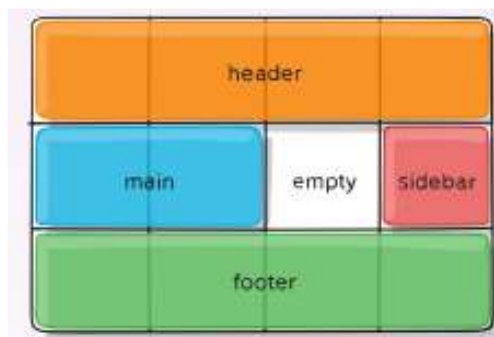
CSS Grid Layout

Tecnologias Web
2023/2024

CSS Grid Layout

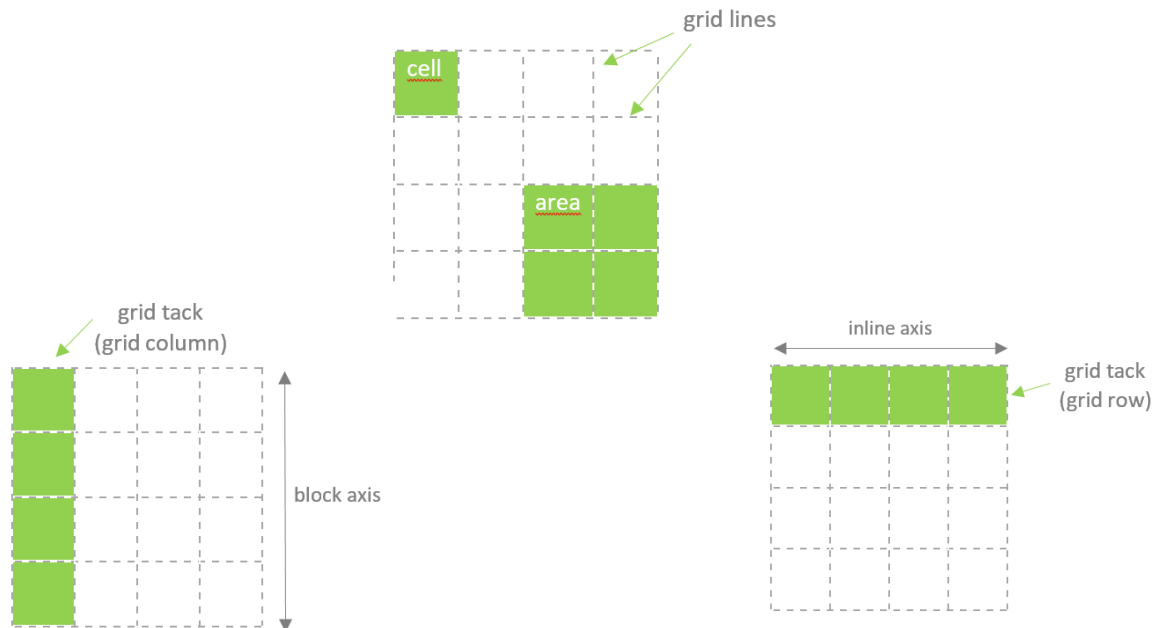
2

- A criação de um *grid container* é efetuada através de ***{display: grid}***
 - Deve ser estabelecido o número de linhas/colunas do grid container
 - Todos os filhos **diretos** de um *grid container* tornam-se *grid items*



CSS Grid Layout

3



CSS Grid Layout

4

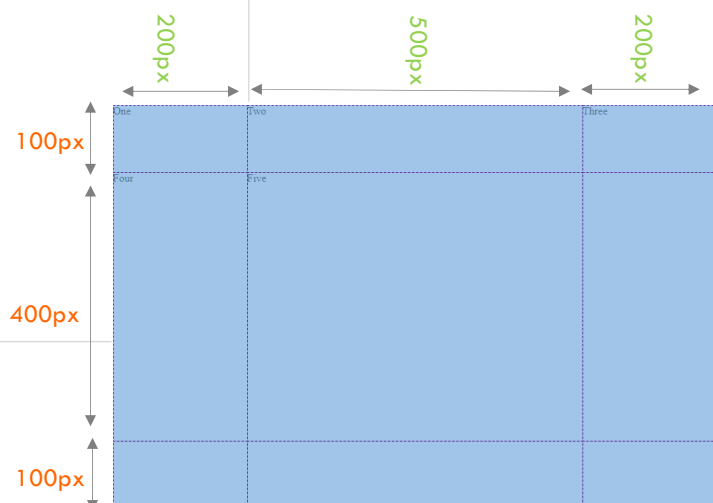
- **grid-template-rows; grid-template-columns** definem n.º e dimensões de linhas/colunas

```
#layout{display: grid;  
        grid-template-rows: 100px 400px 100px;  
        grid-template-columns: 200px 500px 200px}
```

```
</style>  
</head>  
<body>
```

```
<div id="layout">  
  <div id="one">One</div>  
  <div id="two">Two</div>  
  <div id="three">Three</div>  
  <div id="four">Four</div>  
  <div id="five">Five</div>  
</div>
```

Definidas 3 linhas e 3 colunas.
O template por ser dividido no número de
linhas e colunas definidos pelos valores
diferentes atribuídos a **grid-template-rows;**
grid-template-columns



CSS Grid Layout

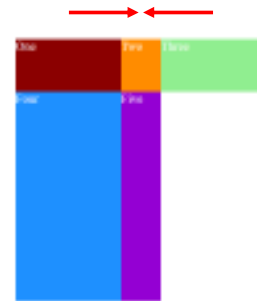
5

- Unidades

- px (fixed); %
- fr (*fractional unit*)
 - permite que os grid tracks se ajustem ao espaço disponível, expandam/retraiam de acordo com as variações do *viewport*.



```
#layout{  
  display: grid;  
  grid-template-rows: 100px 400px 100px;  
  grid-template-columns: 200px 1fr 200px}
```



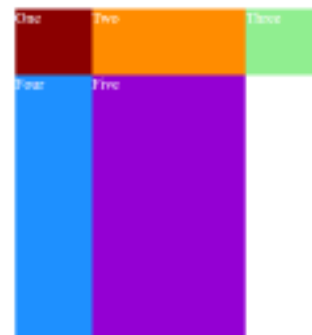
CSS Grid Layout

6

- As *fractional units* podem ser usadas para:
 - combinar áreas de dimensão variável com dimensão fixa (exemplo anterior)
 - combinar áreas de dimensão variável

```
#layout{  
  display: grid;  
  grid-template-rows: 100px 400px 100px;  
  grid-template-columns: 1fr 2fr 1fr}
```

A coluna central terá sempre o dobro da capacidade de ajustamento das colunas dos extremos



CSS Grid Layout - repeat

7

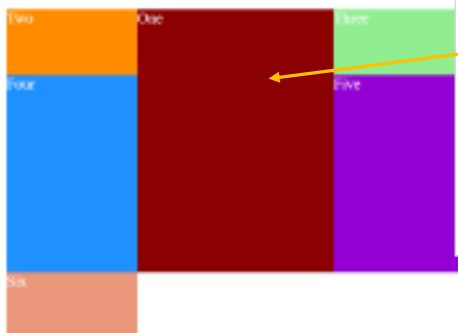
```
#layout{display: grid;
        grid-template-rows: 100px 400px 100px;
        grid-template-columns: repeat(2, 200px 1fr 200px)}
</style>
</head>
<body>
  <div id="layout">
    <div id="one">One</div>
    <div id="two">Two</div>
    <div id="three">Three</div>
    <div id="four">Four</div>
    <div id="five">Five</div>
    <div id="six">Six</div>
  </div>
```

Posicionamento Grid Items

8

- A alteração do posicionamento é efetuado a partir da identificação das linhas tendo por base as propriedades:

- grid-row-start
- grid-row-end
- grid-column-start
- grid-column-end



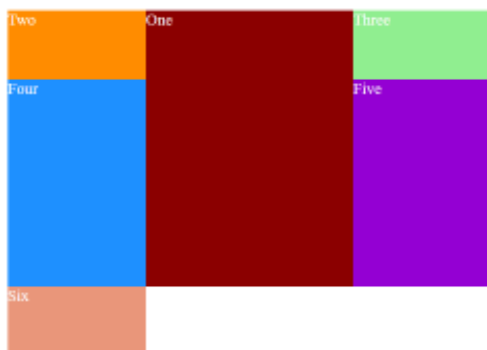
```
#layout{display: grid;
        grid-template-rows: 100px 300px 100px;
        grid-template-columns: 200px 300px 200px;
        grid-template-areas:
          "header header header"
          "ads main links"
          "footer footer footer"
      }
#one{
  grid-row-start: 1;
  grid-row-end: 3;
  grid-column-start: 2;
  grid-column-end: 3;
}
</style>
</head>
<body>
  <div id="layout">
    <div id="one">One</div>
```

Posicionamento Grid Items

9

- O posicionamento anterior pode ser efetuado de forma mais condensada:

- `grid-row: start line / end line`
- `grid-column: start line / end line`



```
#layout{display: grid;
  grid-template-rows: 100px 300px 100px;
  grid-template-columns: 200px 300px 200px;
  grid-template-areas:
    "header header header"
    "ads main links"
    "footer footer footer"
}

#one{
  grid-row: 1 / 3;
  grid-column: 2 / 3;
}

</style>
</head>
<body>
  <div id="layout">
    <div id="one">One</div>
    <div id="two">Two</div>
    <div id="three">Three</div>
    <div id="four">Four</div>
    <div id="five">Five</div>
    <div id="six">Six</div>
  </div>
```

Posicionamento Grid Items

10

- As propriedades **`grid-row-gap`**/**`grid-column-gap`** permitem estabelecer um espaço entre tracks
 - **`grid-gap`** permite agregar estes dois valores por linha e coluna

```
#littlegrid{
  display: grid;
  grid-auto-flow: column;
  grid-gap: 20px;
  grid-template-columns: 200px 200px;
  grid-template-rows: 200px 200px;
  grid-auto-columns: 100px;
  grid-auto-rows: 100px;
}
```

