

**INSTITUTO
FEDERAL**
Santa Catarina

Posicionando elementos (position e grid)

Programação Frontend I

Análise e Desenvolvimento de Sistemas

Prof. Adriano Lima

adriano.lima@ifsc.edu.br



INSTITUTO FEDERAL

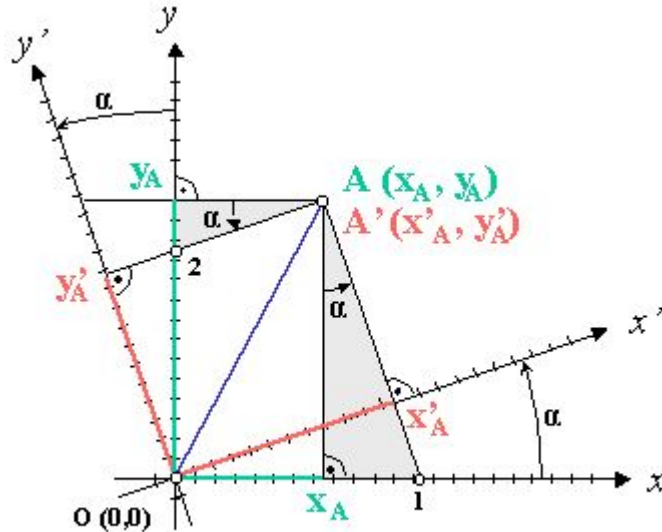
Santa Catarina

Câmpus São José

PENSAMENTO
COMPUTACIONAL E
ALGORITMOS

Análise e Desenvolvimento
de Sistemas

Posicionamento



- CSS oferece vários métodos de posicionamento dos elementos
- o elemento pode ser posicionado:
 - em relação ao seu posicionamento natural
 - fora do seu fluxo natural
 - em um ponto específico da página

Posicionamento

position

| | |
|--------------------|---|
| valores | <code>static relative absolute fixed</code> |
| padrão | <code>static</code> |
| se aplica a | todos os elementos |
| herança | não |

Posicionamento

top, right, bottom, left

valores *comprimento* | *percentual* | auto

padrão auto

se aplica a elementos posicionados (onde o valor da posição é relativo, absoluto ou fixo)

herança não

- define a distância que o elemento deve ser afastado da respectiva aresta

Posicionamento

```
em {  
  position: relative;  
  top: 2em;  
  left: 3em;  
  color: white;  
  background-color: #611B2E;  
}
```

posição relativa

Lorem, ipsum dolor sit amet elit. Alias, elemento reposicionado quod dolores corrupti unde dolorum ducimus deserunt sequi ullam autem sapiente harum itaque. Impedit modi odio ipsa? Lorem ipsum dolor sit amet, consectetur adipiscing elit. Omnis doloremque et itaque recusandae laudantium, sint est dignissimos deleniti aspernatur reiciendis maxime quo eum labore iure. Maiores veniam sint delectus minima.

Posicionamento

```
em {  
  position: relative;  
  top: 2em;  
  left: 3em;  
  color: white;  
  background-color: #611B2E;  
}
```

posição relativa

Lorem, ipsum dolor sit amet elit. Alias, elemento reposicionado quod dolores corrupti unde dolorum ducimus deserunt sequi ullam autem sapiente harum itaque. Impedit modi odio ipsa? Lorem ipsum consectetur adipisicing elit. Omnis doloremque et itaque recusandae laudantium, sint est dignissimos deleniti aspernatur reiciendis maxime quo eum labore iure. Maiores veniam sint delectus minima.



Posicionamento

```
em {  
  position: relative;  
  top: 2em;  
  left: 3em;  
  color: white;  
  background-color: #611B2E;  
}
```

- espaço original preservado
- sobreposição

posição relativa

Lorem, ipsum dolor sit amet elit. Alias, elemento reposicionado quod dolores corrupti unde dolorum ducimus deserunt sequi ullam autem sapiente harum itaque. Impedit modi odio ipsa? Lorem ipsum consectetur adipisicing elit. Omnis doloremque et itaque recusandae laudantium, sint est dignissimos deleniti aspernatur reiciendis maxime quo eum labore iure. Maiores veniam sint delectus minima.



Posicionamento

```
em {  
  position: absolute;  
  top: 2em;  
  left: 3em;  
  color: white;  
  background-color: #611B2E;  
}
```

posição absoluta

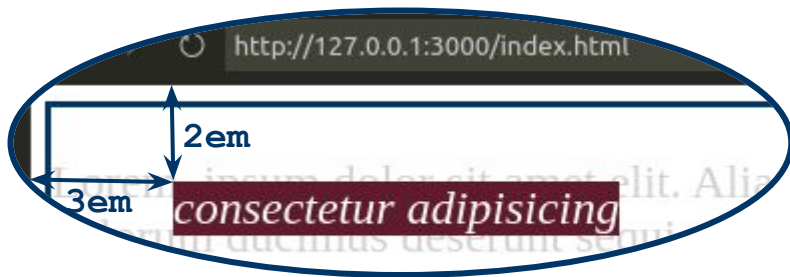
Lorem ipsum dolor sit amet elit. Alias, elemento reposicionado quod dolores corrupti unde dolorum ducimus deserunt sequi ullam autem sapiente harum itaque. Impedit modi odio ipsa? Lorem ipsum dolor sit amet consectetur adipisicing elit. Omnis doloremque et itaque recusandae laudantium, sint est dignissimos deleniti aspernatur reiciendis maxime quo eum labore iure. Maiores veniam sint delectus minima.

Posicionamento

```
em {  
  position: absolute;  
  top: 2em;  
  left: 3em;  
  color: white;  
  background-color: #611B2E;  
}
```

posição absoluta

consectetur adipisicing elit. Alias, elemento reposicionado quod dolores corrupti unde dolorum duemus deserunt sequi ullam autem sapiente harum itaque. Impedit modi odio ipsa? Lorem ipsum dolor sit amet consectetur adipisicing elit. Omnis doloremque et itaque recusandae laudantium, sint est dignissimos deleniti aspernatur reiciendis maxime quo eum labore iure. Maiores veniam sint delectus minima.



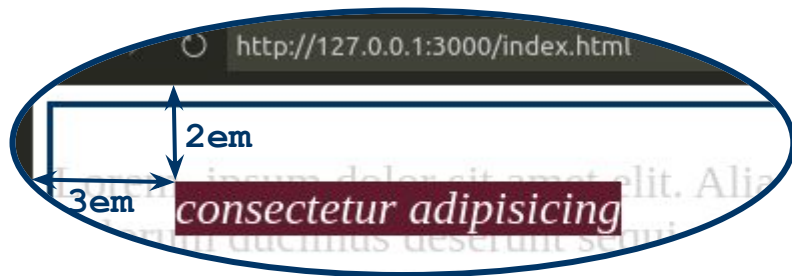
Posicionamento

```
em {  
  position: absolute;  
  top: 2em;  
  left: 3em;  
  color: white;  
  background-color: #611B2E;  
}
```

- posicionado em relação ao contêiner posicionado mais próximo

posição absoluta

Lorem ipsum dolor sit amet elit. Alias, elemento reposicionado quod dolores corrupti unde dolorum duimus deserunt sequi ullam autem sapiente harum itaque. Impedit modi odio ipsa? Lorem ipsum dolor sit amet consectetur adipiscing elit. Omnis doloremque et itaque recusandae laudantium, sint est dignissimos deleniti aspernatur reiciendis maxime quo eum labore iure. Maiores veniam sint delectus minima.

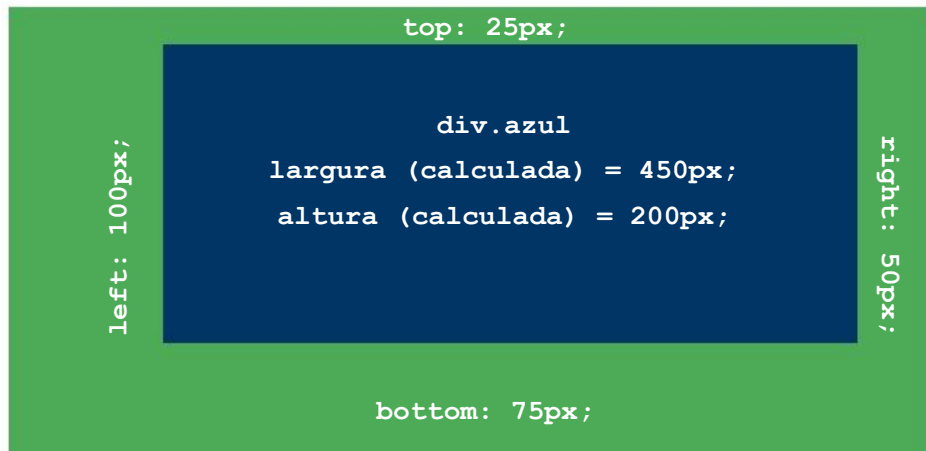


Posicionamento

```
div.verde {  
  position: relative;  
  height: 300px;  
  width: 600px;  
}
```

```
div.azul {  
  position: absolute;  
  top: 25px;  
  right: 50px;  
  bottom: 75px;  
  left: 100px;  
}
```

posição absoluta com pixels



Posicionamento

```
div.cinza {  
  position: absolute;  
  top: 50%;  
  left: 0%;  
}
```

```
div.vermelho {  
  position: absolute;  
  right: 0%;  
  bottom: 0%;  
}
```

posição absoluta com porcentagem



Posicionamento

```
div.cinza {  
  position: absolute;  
  top: 50%;  
  left: 0%;  
}
```

```
div.vermelho {  
  position: absolute;  
  right: 0%;  
  bottom: 0%;  
}
```

posição absoluta com porcentagem



Posicionamento

```
div.cinza {  
  position: relative;  
  top: 50px;  
  left: 250px;  
}
```

```
div.vermelho {  
  position: absolute;  
  top: 100px;  
  left: 200px;  
}
```

```
div.amarelo {  
  position: absolute;  
  top: 125px;  
  left: 275px;  
}
```

posição absoluta com empilhamento



Posicionamento

z-index

| | |
|--------------------|------------------------|
| valores | <i>número</i> auto |
| padrão | auto |
| se aplica a | elementos posicionados |
| herança | não |

Posicionamento

```
div.cinza {  
  position: relative;  
  top: 50px;  
  left: 250px;  
}
```

```
div.vermelho {  
  position: absolute;  
  top: 100px;  
  left: 200px;  
}
```

```
div.amarelo {  
  position: absolute;  
  top: 125px;  
  left: 275px;  
}
```

posição absoluta com empilhamento



Posicionamento

```
div.cinza {  
  position: relative;  
  top: 50px;  
  left: 250px;  
}
```

```
div.vermelho {  
  position: absolute;  
  top: 100px;  
  left: 200px;  
}
```

```
div.amarelo {  
  position: absolute;  
  top: 125px;  
  left: 275px;  
}
```

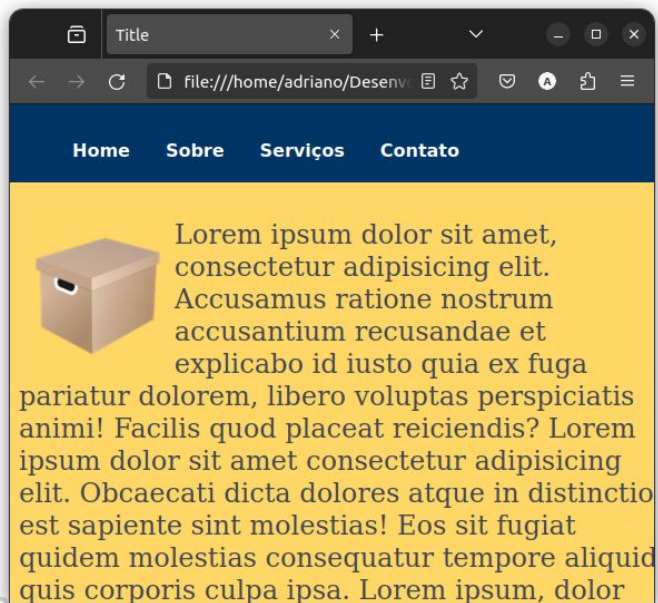
posição absoluta com empilhamento



Posicionamento

```
#header {  
  position: fixed;  
  top: 0%;  
  left: 0%;  
  z-index: 1000;  
}
```

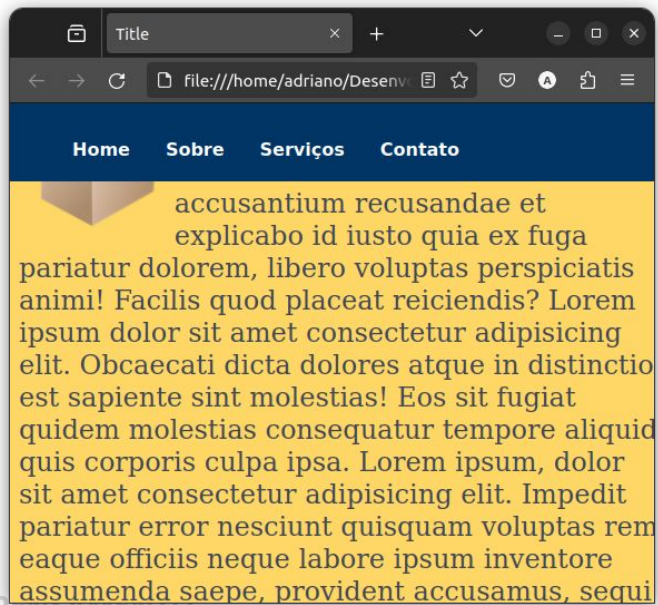
posição fixa



Posicionamento

```
#header {  
  position: fixed;  
  top: 0%;  
  left: 0%;  
  z-index: 1000;  
}
```

posição fixa



Grid



- sistema para organizar elementos em linhas e colunas
- flexível para caber em uma variedade de tamanhos de tela
- permite criar layouts sofisticados com tipografia e espaços em branco
- uma das especificações mais complexas do CSS



THE EXPERIMENTAL LAYOUT LAB OF JEN SIMMONS

LAYOUT LAND VIDEOS
JENSIMMONS.COM

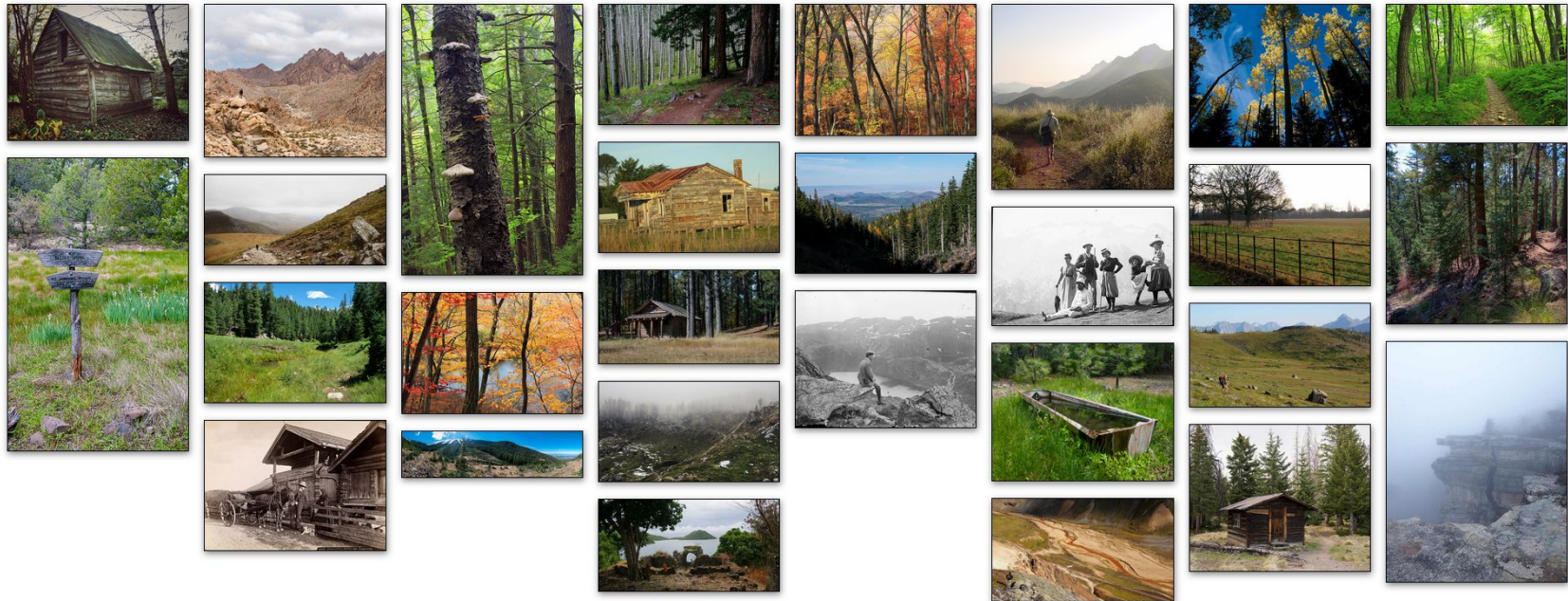
THE CONFERENCE TALKS

1. MODERN LAYOUTS: GETTING OUT OF OUR RUTS
2. REVOLUTIONIZE YOUR PAGE: REAL ART DIRECTION ON THE WEB
3. DESIGNING WITH GRID
4. EVERYTHING YOU KNOW ABOUT WEB DESIGN JUST CHANGED
5. DESIGNING INTRINSIC LAYOUTS

Follow @JENSIMMONS on Twitter for more as it happens.



The Woods in Multicolumn Layout



Our Team



PENSAMENTO
COMPUTACIONAL E
ALGORITMOS

Análise e Desenvolvimento
de Sistemas

Grid

display

valores

inline | block | run-in | flex | **grid** | flow |
flow-root | list-item | table | table-row-group |
table-header-group | table-footer-group | table-row |
table-cell | table-column-group | table-column |
table-caption | ruby | ruby-base | ruby-text |
ruby-base-container | ruby-text-container |
inline-block | inline-table | inline-flex |
inline-grid | contents | none

padrão

inline

se aplica a

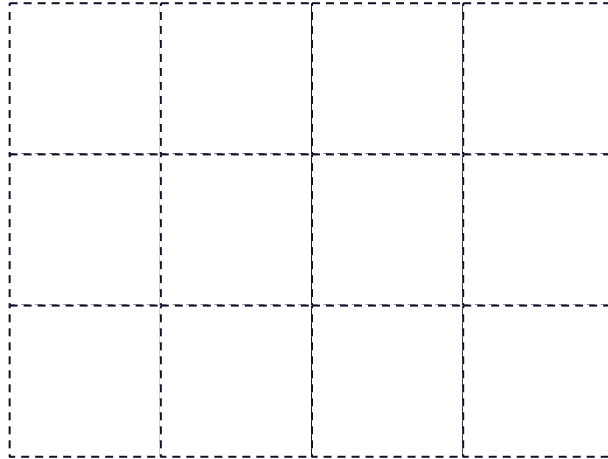
todos os elementos

herança

sim



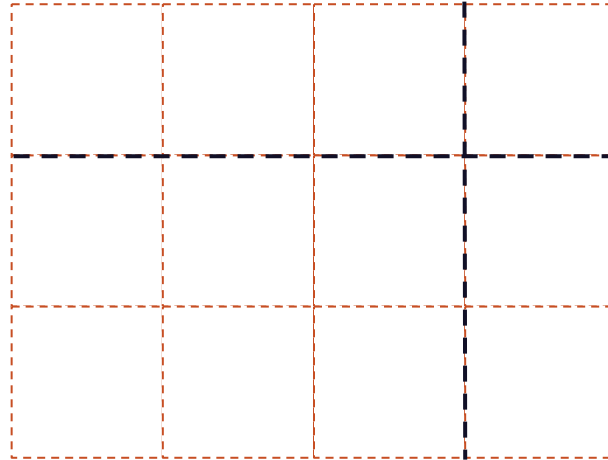
Grid



contêiner grid



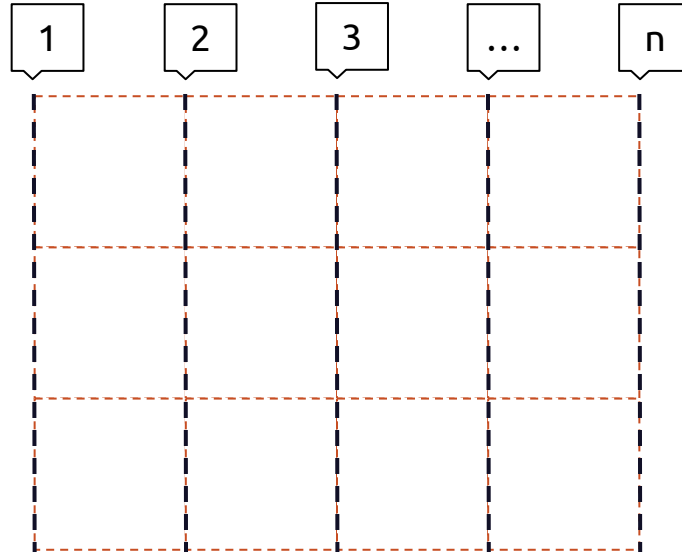
Grid



divisórias do grid

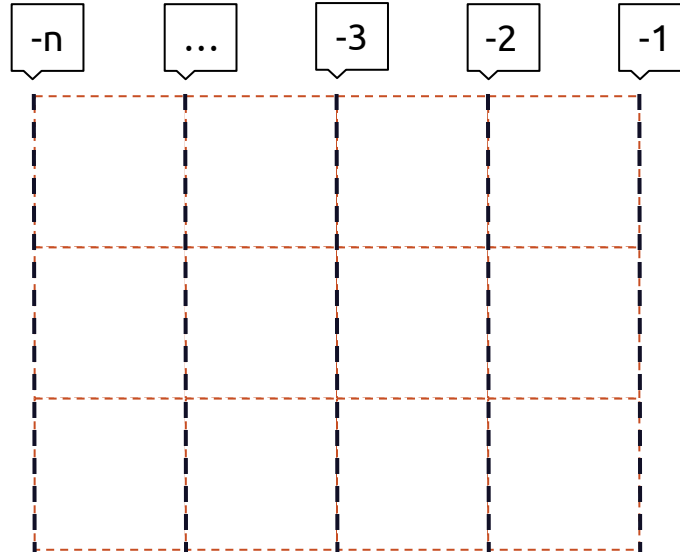


Grid



divisórias verticais

Grid



divisórias verticais

Grid



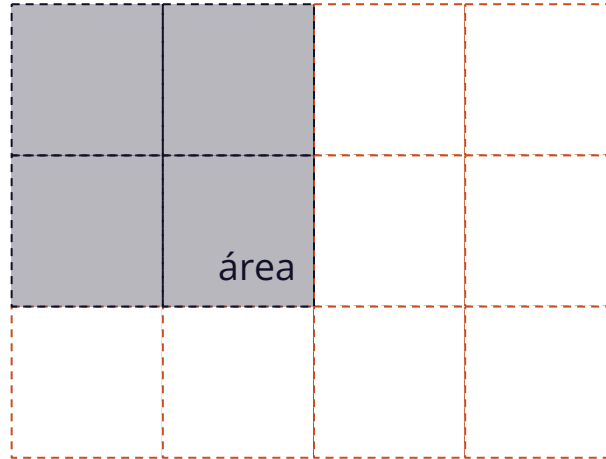
Grid



célula do grid

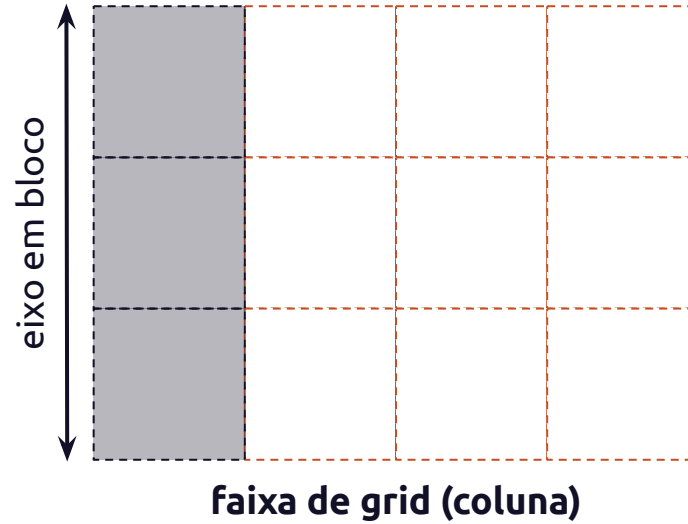


Grid



área de grid

Grid




Grid



Grid

contêiner / layout



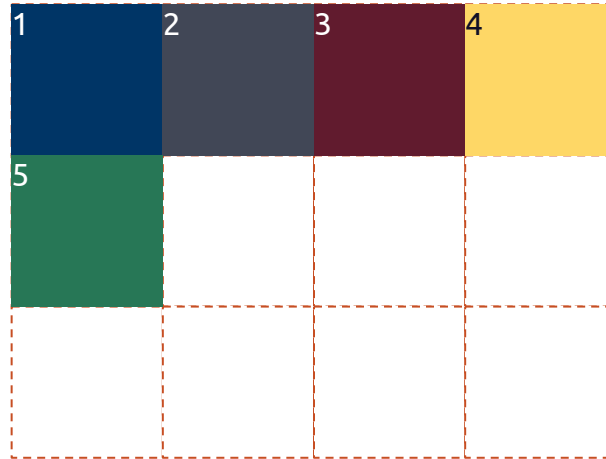
```
<div id="layout">  
  <div id="um"> 1 </div>  
  <div id="dois"> 2 </div>  
  <div id="tres"> 3 </div>  
  <div id="quatro"> 4 </div>  
  <div id="cinco"> 5 </div>  
</div>
```

```
#layout {  
  display: grid;  
}
```

itens / conteúdo

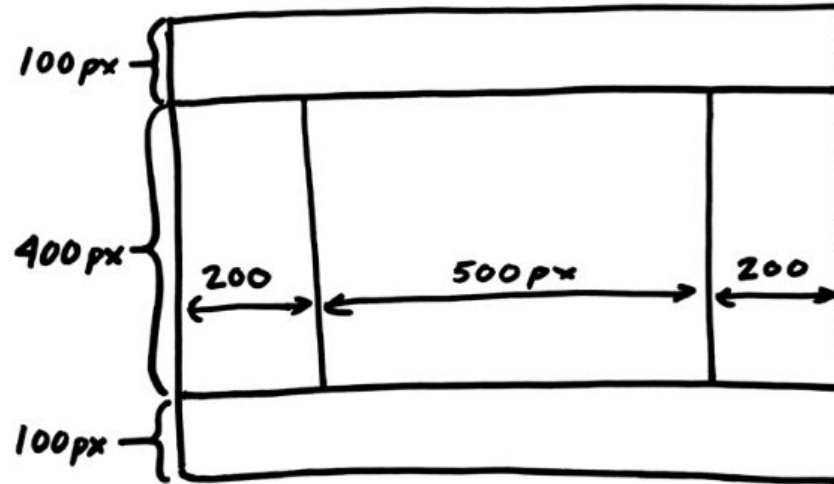


Grid



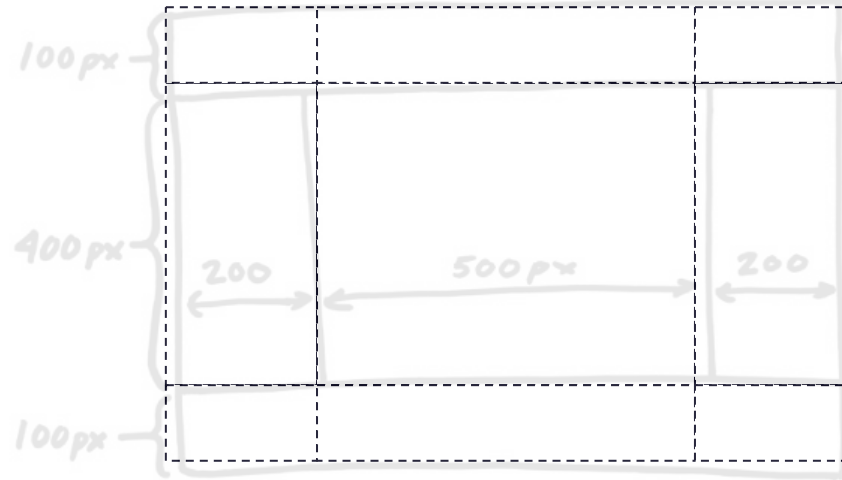
contêiner e itens

Grid



ideia de layout

Grid

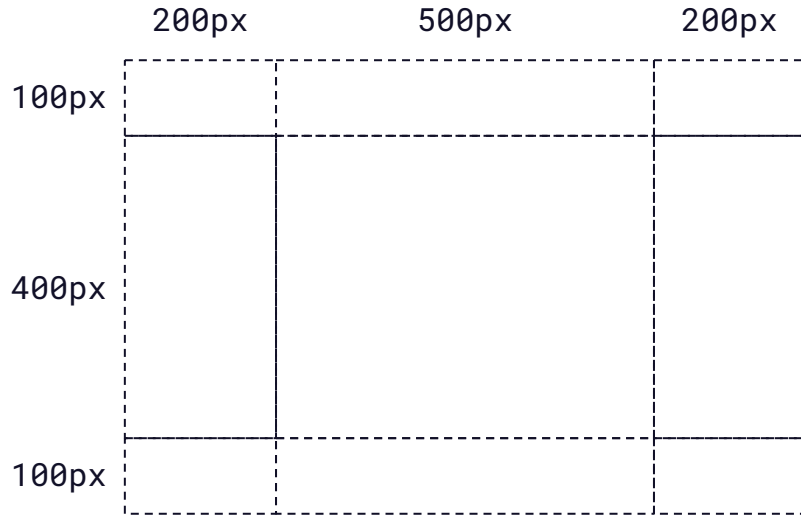


Grid

grid-template-rows, grid-template-columns

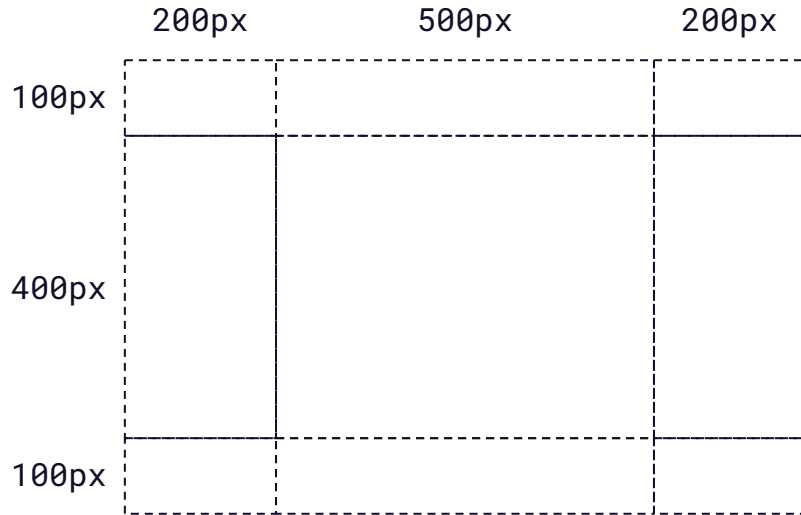
| | |
|--------------------|---|
| valores | auto <i>lista de tamanhos de faixa e nome de linha opcional</i> |
| padrão | none |
| se aplica a | contêineres grid |
| herança | não |

Grid



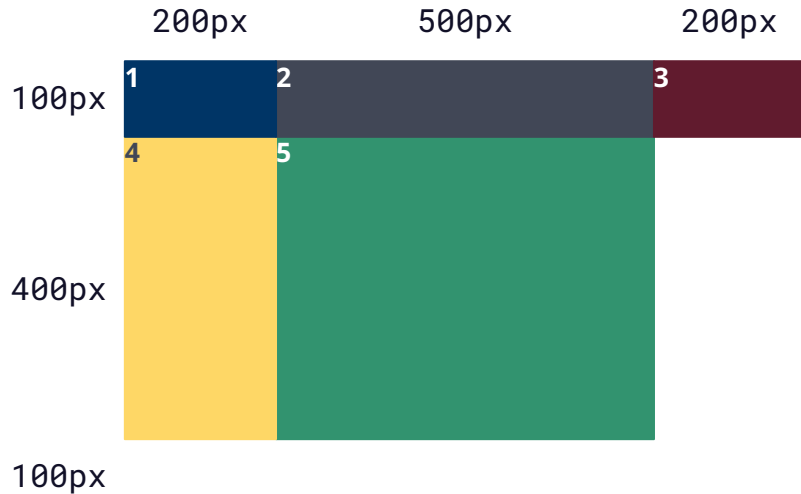
```
#layout {  
  display: grid;  
  grid-template-rows: 100px 400px 100px;  
  grid-template-columns: 200px 500px 200px;  
}
```

Grid



```
#layout {  
  display: grid;  
  grid-template-rows: 100px 400px 100px;  
  grid-template-columns: 200px 500px 200px;  
}  
  
<div id="layout">  
  <div id="um">1</div>  
  <div id="dois">2</div>  
  <div id="tres">3</div>  
  <div id="quatro">4</div>  
  <div id="cinco">5</div>  
</div>
```

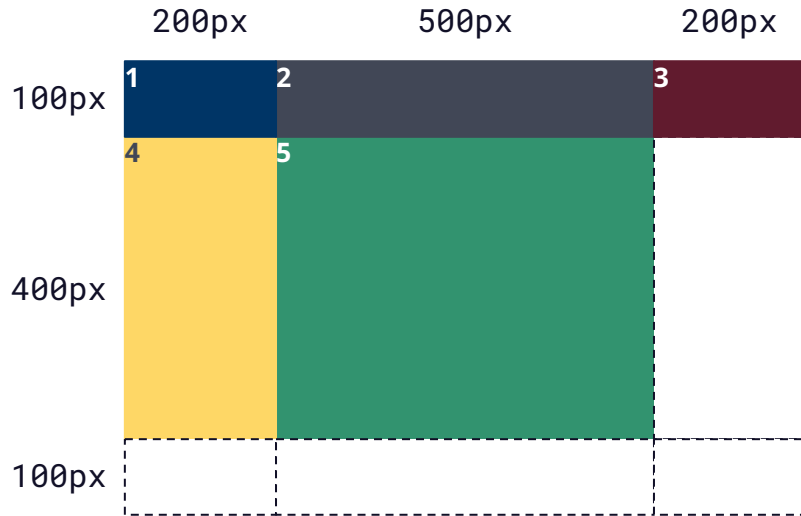

Grid



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

```
<div id="layout">  
  <div id="um">1</div>  
  <div id="dois">2</div>  
  <div id="tres">3</div>  
  <div id="quatro">4</div>  
  <div id="cinco">5</div>  
</div>
```

Grid

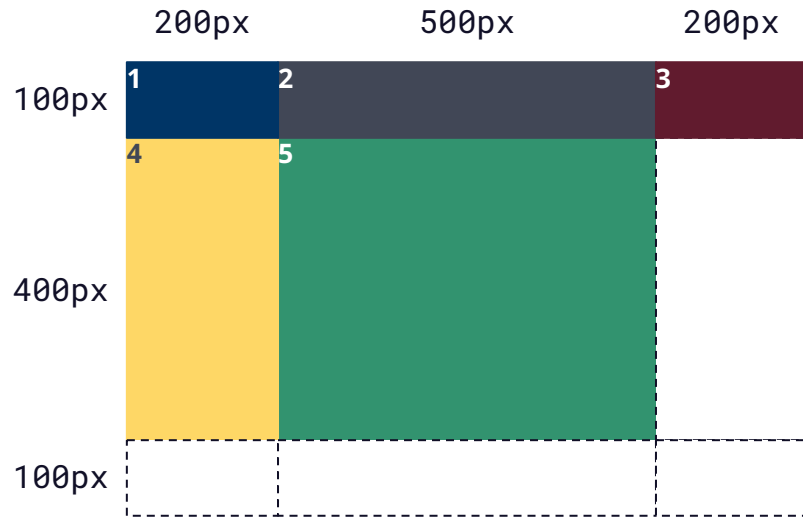


```
#layout {  
  display: grid;  
  grid-template-rows: 100px 400px 100px;  
  grid-template-columns: 200px 500px 200px;  
}
```

```
<div id="layout">  
  <div id="um">1</div>  
  <div id="dois">2</div>  
  <div id="tres">3</div>  
  <div id="quatro">4</div>  
  <div id="cinco">5</div>  
</div>
```

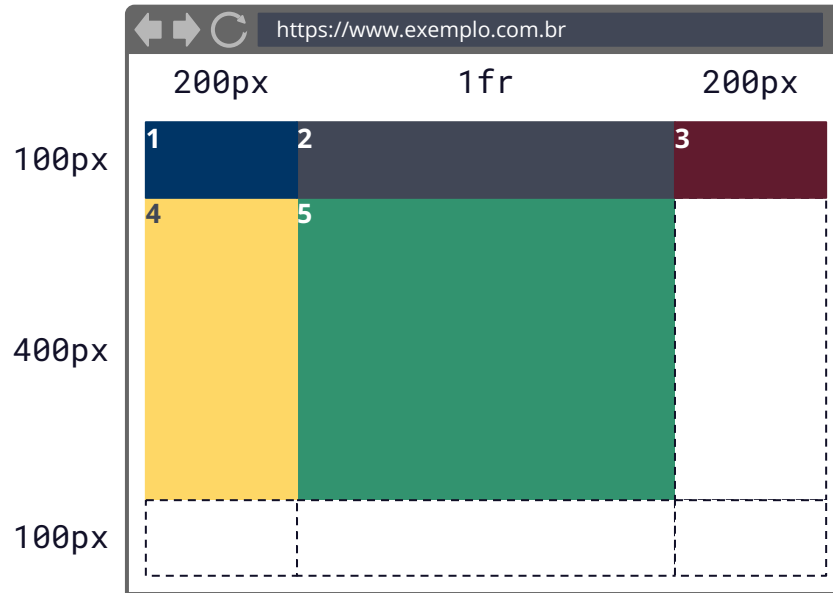
Grid

definindo valores de tamanho de faixa



Grid

definindo valores de tamanho de faixa

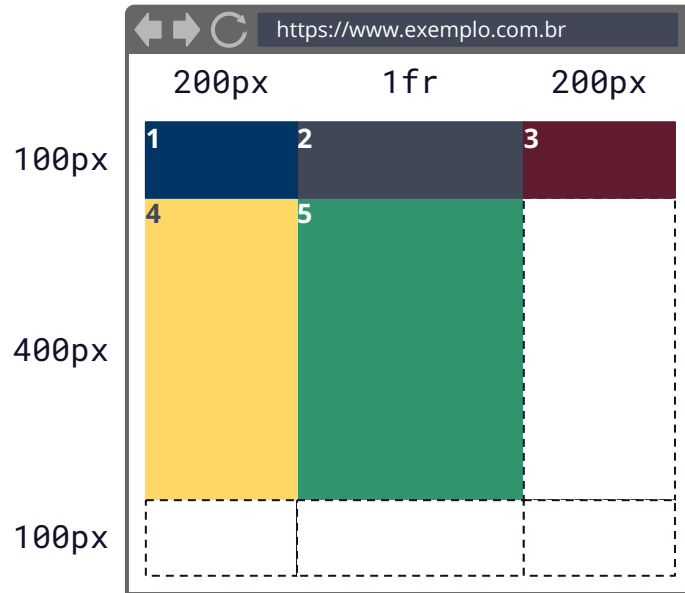


fr fração

```
grid-template-columns: 200px 1fr 200px;
```

Grid

definindo valores de tamanho de faixa

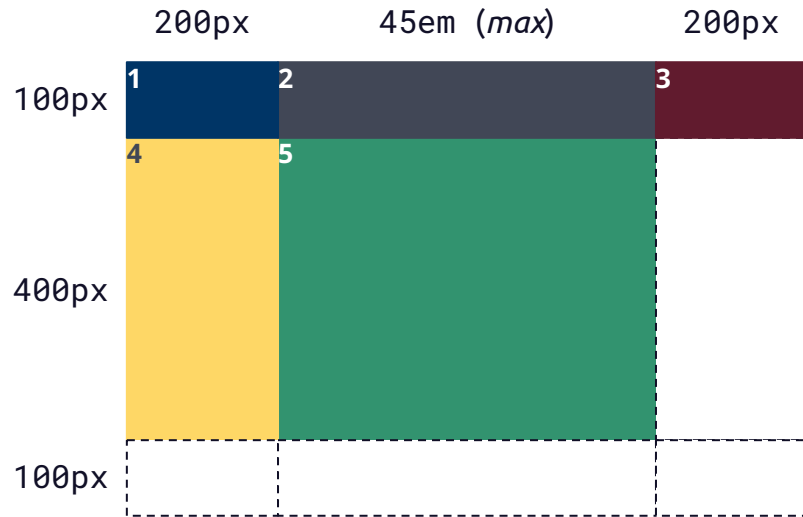


fr fração

```
grid-template-columns: 200px 1fr 200px;
```

Grid

definindo valores de tamanho de faixa

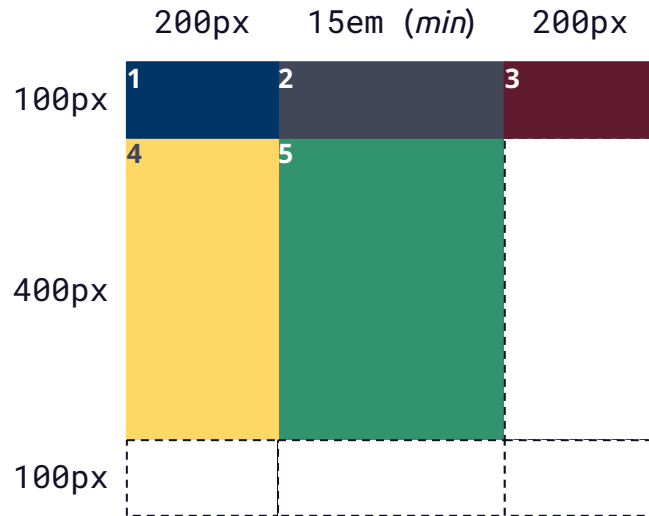


maxmin() tamanho mínimo e máximo

```
grid-template-columns: 200px minmax(15em, 45em)  
200px;
```

Grid

definindo valores de tamanho de faixa



maxmin() tamanho mínimo e máximo

```
grid-template-columns: 200px minmax(15em, 45em)  
200px;
```

Grid

definindo valores de tamanho de faixa

repeat() repetindo valores

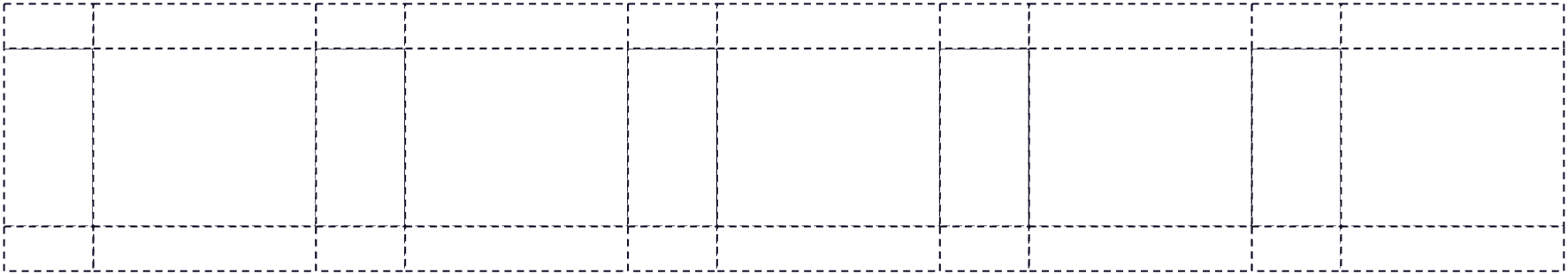
```
grid-template-columns: repeat(5, 20px, 1fr);
```


Grid

definindo valores de tamanho de faixa

repeat() repetindo valores

```
grid-template-columns: repeat(5, 20px, 1fr);
```

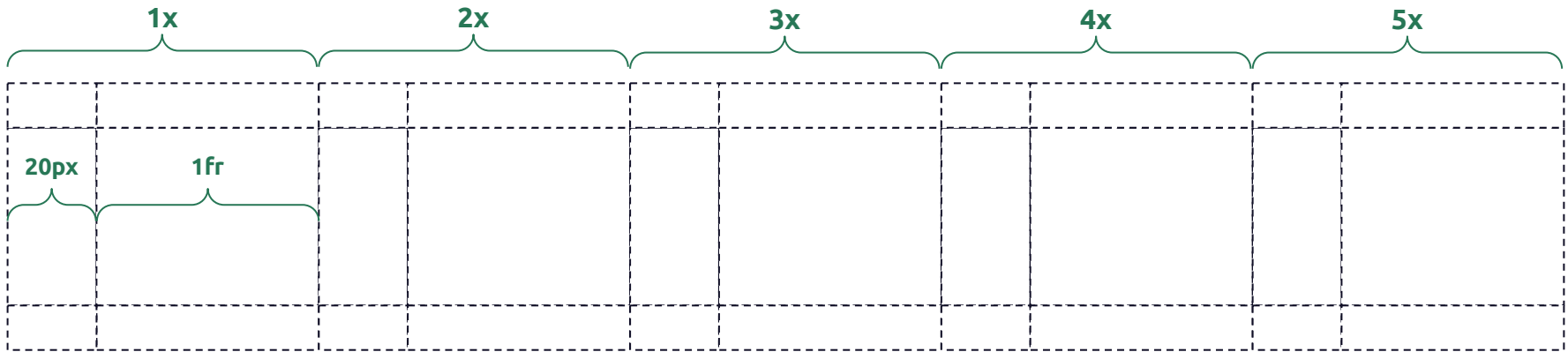


Grid

definindo valores de tamanho de faixa

`repeat()` repetindo valores

```
grid-template-columns: repeat(5, 20px, 1fr);
```



Grid

definindo valores de tamanho de faixa



repeat() repetindo valores

```
grid-template-rows: repeat(auto-fill, 10em);
```

- faixas são repetidas até preencher o contêiner

Grid

definindo valores de tamanho de faixa



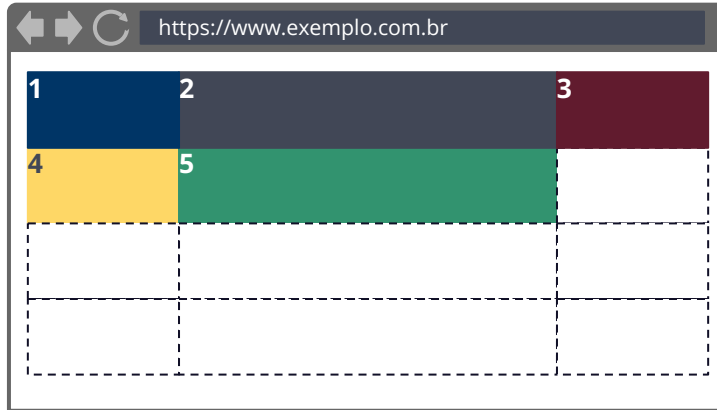
repeat() repetindo valores

```
grid-template-rows: repeat(auto-fill, 10em);
```

- faixas são repetidas até preencher o contêiner

Grid

definindo valores de tamanho de faixa



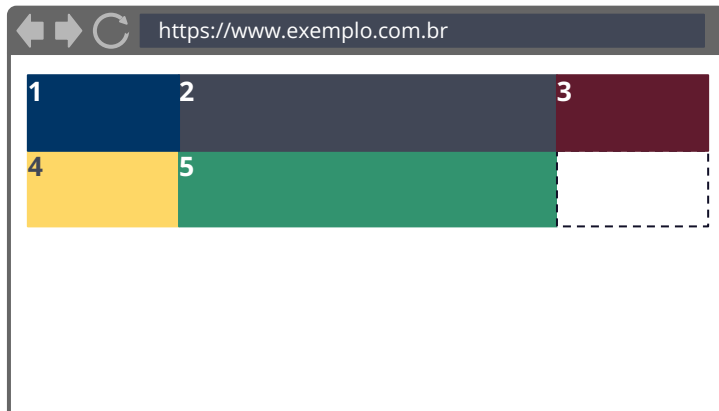
repeat() repetindo valores

```
grid-template-rows: repeat(auto-fill, 10em);
```

- faixas sem conteúdo são excluídas

Grid

definindo valores de tamanho de faixa



repeat() repetindo valores

```
grid-template-rows: repeat(auto-fill, 10em);
```

- faixas sem conteúdo são excluídas

Grid

grid-template-areas

valores none | *série de nomes de área*

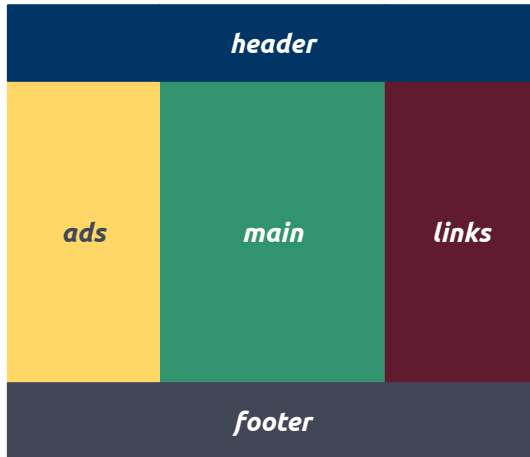
padrão none

se aplica a contêineres grid

herança não

Grid

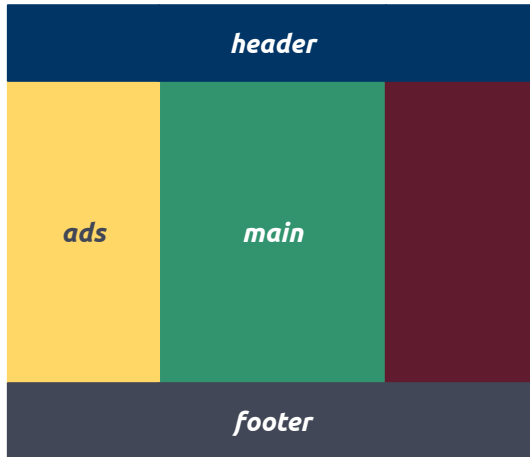
definindo nomes de áreas de grid



```
grid-template-areas:  
  "header header header"  
  "ads main links"  
  "footer footer footer";
```


Grid

definindo nomes de áreas de grid

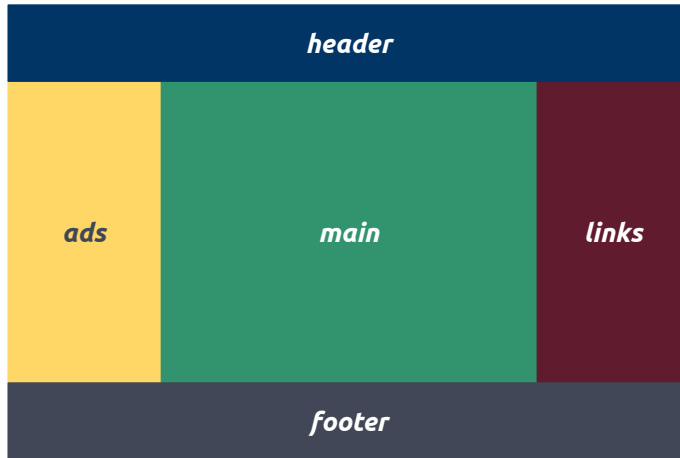


```
grid-template-areas:  
  "header header header"  
  "ads main ."  
  "footer footer footer";
```

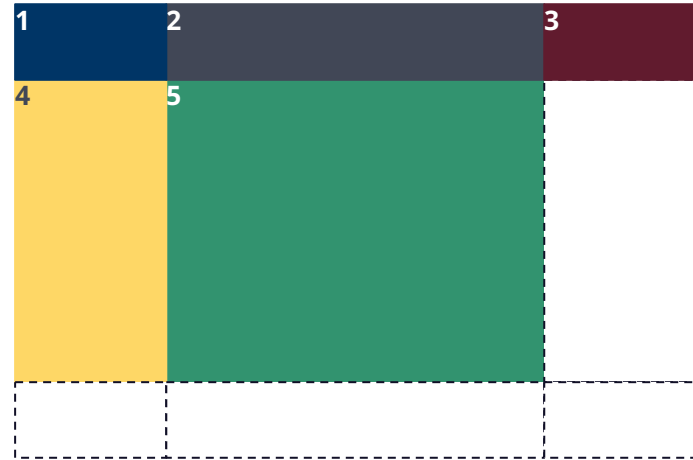
Grid

posicionando o conteúdo

contêiner / layout



itens / conteúdo



Grid

**grid-row-start, grid-row-end,
grid-column-start, grid-column-end**

valores none | *informações de linha/coluna*

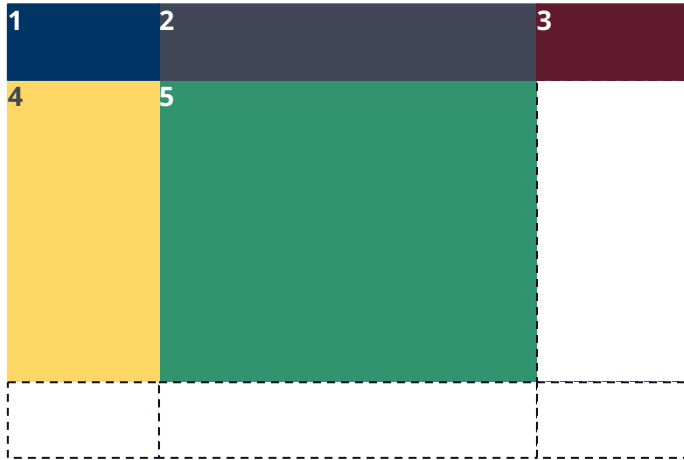
padrão none

se aplica a items grid

herança não

Grid

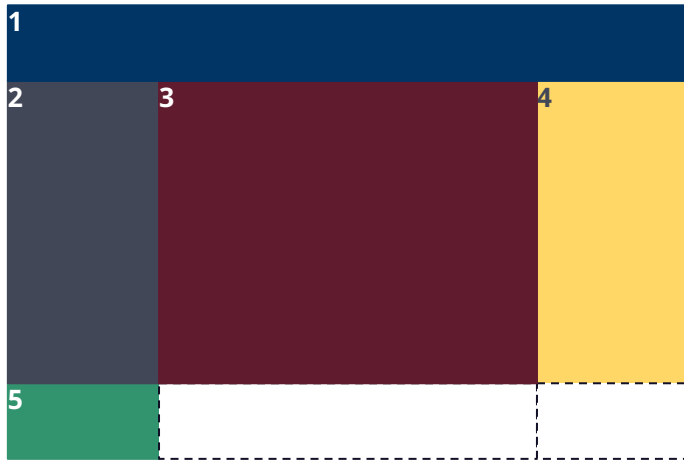
posicionando o conteúdo



```
#um {  
  grid-row-start: 1;  
  grid-row-end: 2;  
  grid-column-start: 1;  
  grid-column-end: 4;  
}
```

Grid

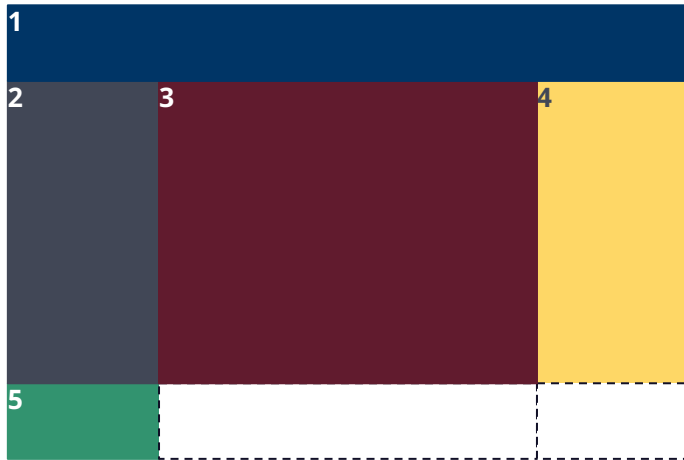
posicionando o conteúdo



```
#um {  
  grid-row-start: 1;  
  grid-row-end: 2;  
  grid-column-start: 1;  
  grid-column-end: 4;  
}
```

Grid

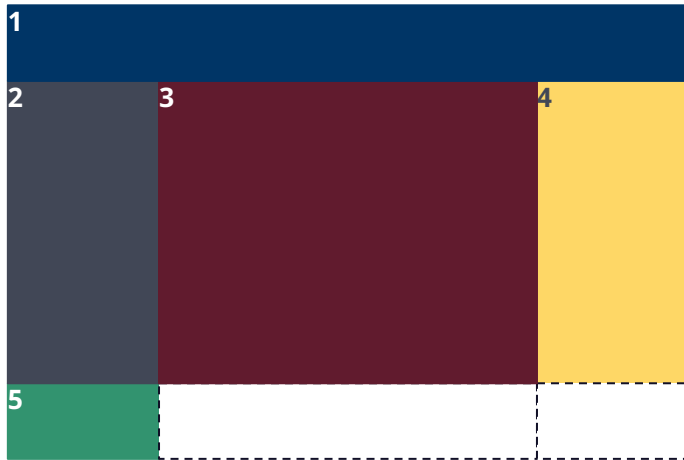
posicionando o conteúdo



```
#um {  
  grid-row-start: 1;  
  grid-row-end: 2;  
  grid-column-start: 1;  
  grid-column-end: -1;  
}
```

Grid

posicionando o conteúdo



```
#um {  
  grid-row-start: 1;  
  grid-row-end: 2;  
  grid-column-start: 1;  
  grid-column-end: span 3;  
}
```

Grid

grid-row, grid-column

valores *divisória inicial / divisória final*

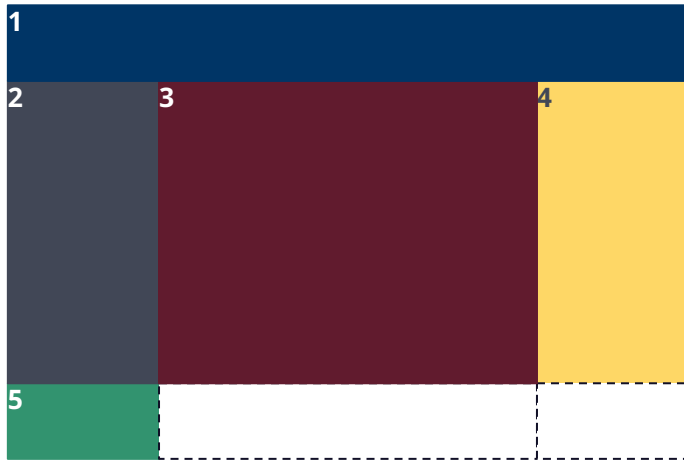
padrão veja propriedades individuais

se aplica a items grid

herança não

Grid

posicionando o conteúdo



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

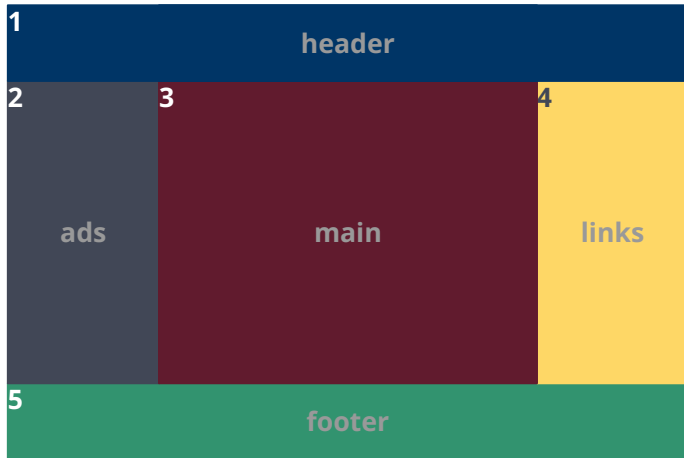
Grid

grid-area

| | |
|--------------------|-------------------------------|
| valores | <i>nome da área</i> |
| padrão | veja propriedades individuais |
| se aplica a | items grid |
| herança | não |

Grid

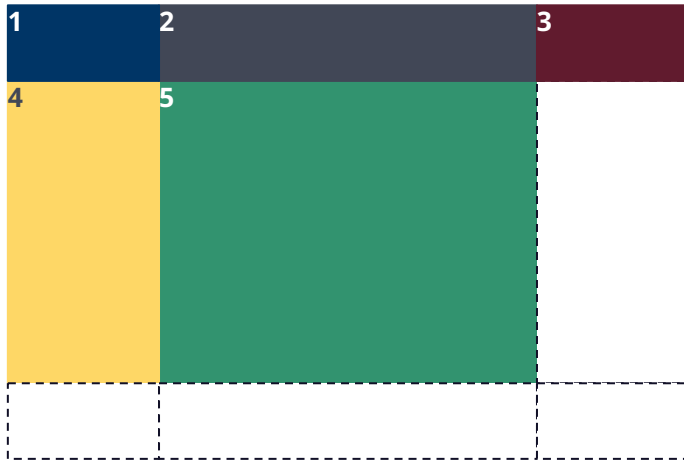
posicionando o conteúdo



```
#um { grid-area: header; }  
#dois { grid-area: ads; }  
#tres { grid-area: main; }  
#quatro { grid-area: links; }  
#cinco { grid-area: footer; }
```

Grid

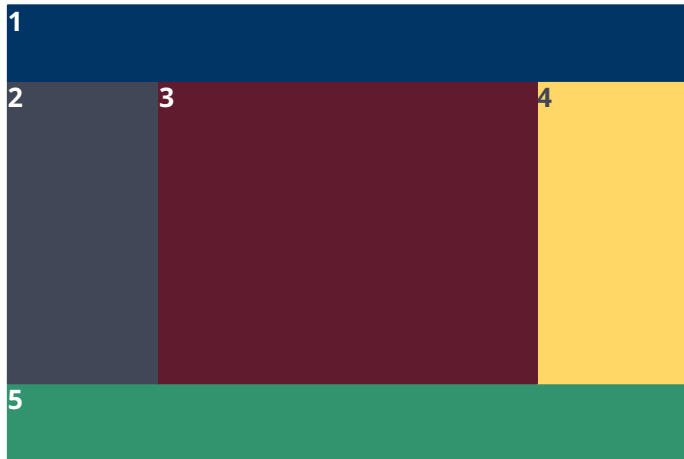
comportamento implícito



- posicionamento em sequência

Grid

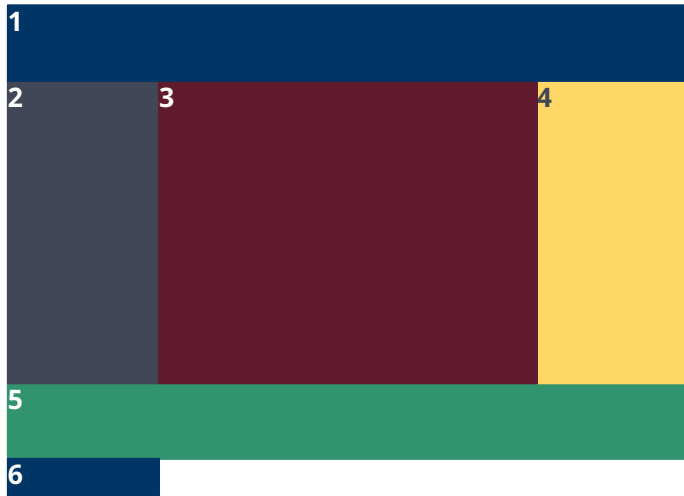
comportamento implícito



- posicionamento em sequência
- **criação de faixas sob demanda**

Grid

comportamento implícito



- posicionamento em sequência
- **criação de faixas sob demanda**

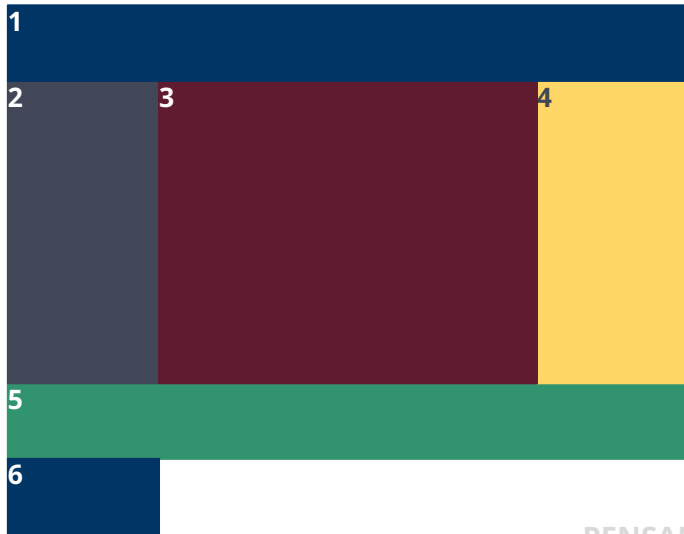
Grid

grid-auto-rows, grid-auto-columns

| | |
|--------------------|-----------------------------------|
| valores | <i>lista de tamanhos de pista</i> |
| padrão | auto |
| se aplica a | contêineres grid |
| herança | não |

Grid

comportamento implícito



- posicionamento em sequência
- criação de faixas sob demanda

`grid-auto-rows: 100px;`

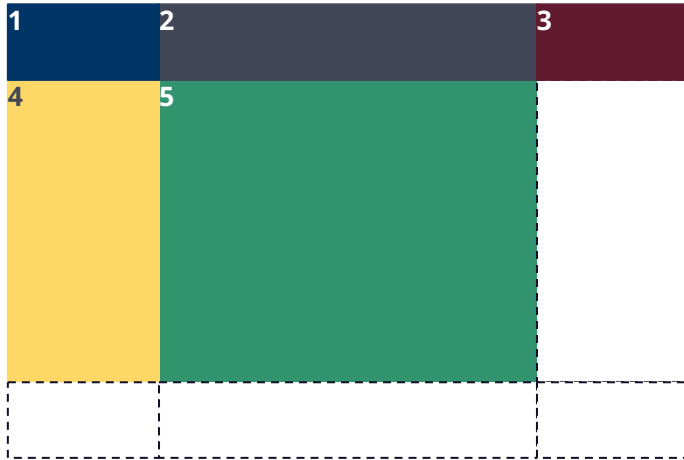
Grid

grid-auto-flow

| | |
|--------------------|------------------|
| valores | row column |
| padrão | row |
| se aplica a | contêineres grid |
| herança | não |

Grid

comportamento implícito

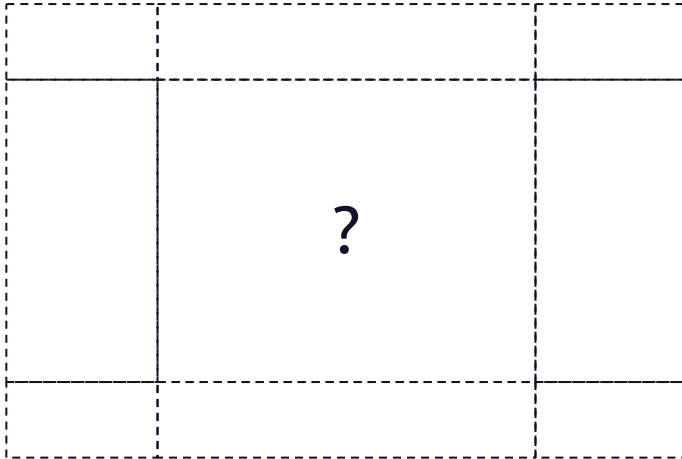


- posicionamento em sequência
- criação de faixas sob demanda

```
grid-auto-flow: row;
```

Grid

comportamento implícito

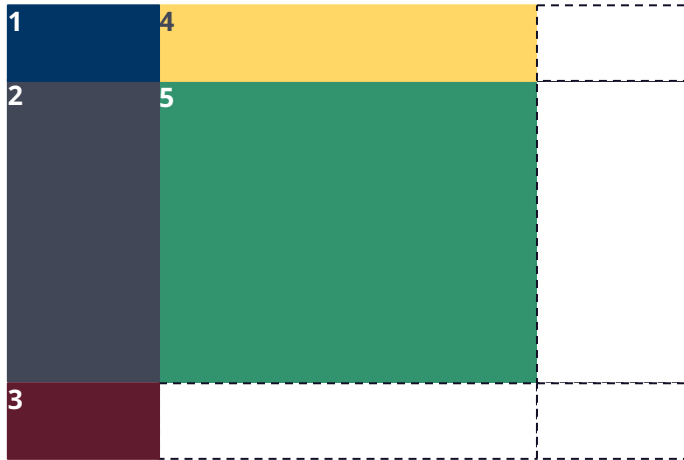


- posicionamento em sequência
- criação de faixas sob demanda

`grid-auto-flow: column;`

Grid

comportamento implícito



- posicionamento em sequência
- criação de faixas sob demanda

`grid-auto-flow: column;`

Grid

grid-row-gap, grid-column-gap

| | |
|--------------------|--|
| valores | <i>comprimento (não deve ser negativo)</i> |
| padrão | 0 |
| se aplica a | contêineres grid |
| herança | não |

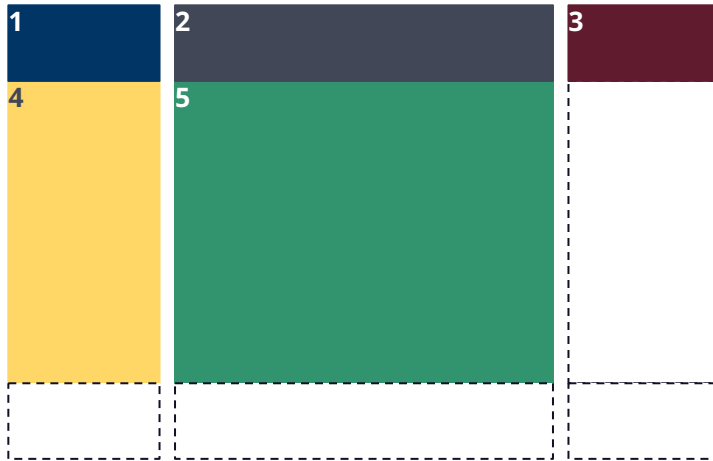
Grid

espaçamento entre as células



Grid

espaçamento entre as células



`grid-column-gap: 20px;`

Grid

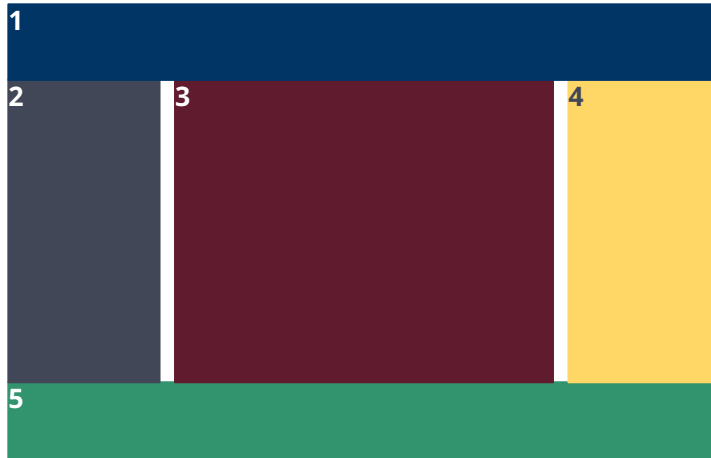
espaçamento entre as células



`grid-column-gap: 20px;`

Grid

espaçamento entre as células



```
grid-column-gap: 20px;
```

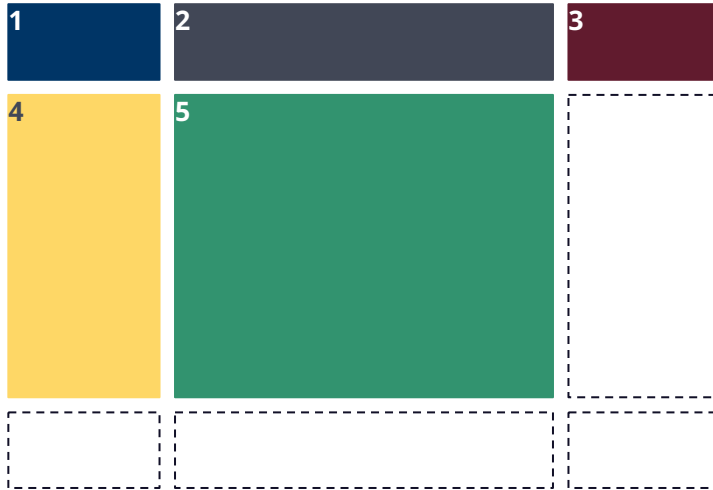
Grid

grid-gap

| | |
|--------------------|---------------------------------|
| valores | grid-row-gap grid-column-gap |
| padrão | 0 |
| se aplica a | contêineres grid |
| herança | não |

Grid

espaçamento entre as células



`grid-gap: 20px 20px;`

Grid

justify-self

| | |
|--------------------|--|
| valores | start end center left right self-start self-end stretch normal auto |
| padrão | auto |
| se aplica a | itens de grid |
| herança | não |

- alinhamento no eixo principal

Grid

alinhamento das células

`justify-self`



Grid

alinhamento das células

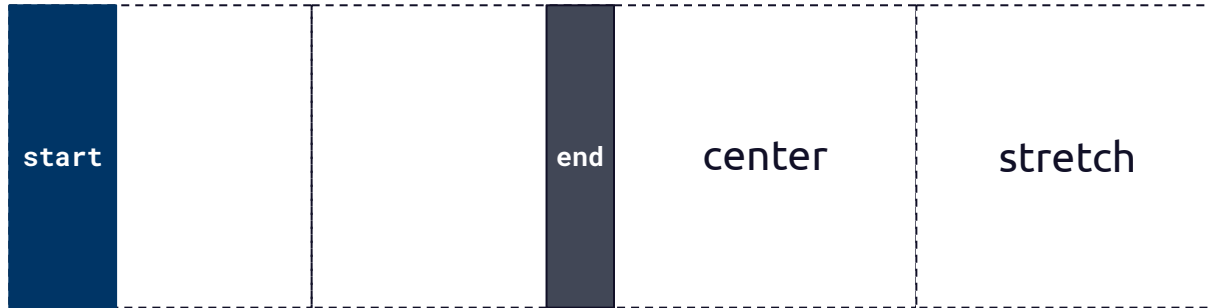
justify-self



Grid

alinhamento das células

justify-self



Grid

alinhamento das células

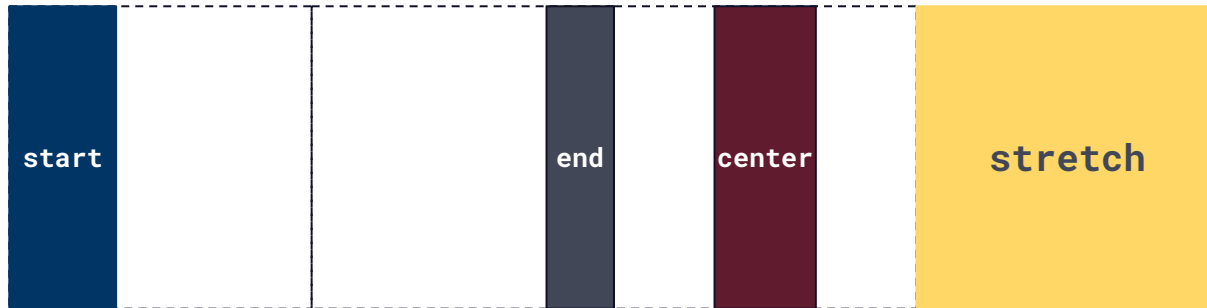
justify-self



Grid

alinhamento das células

justify-self



Grid

align-self

| | |
|--------------------|--|
| valores | start end center left right self-start self-end stretch normal auto |
| padrão | auto |
| se aplica a | itens de grid |
| herança | não |

- alinhamento individual no eixo cruzado

Grid

alinhamento das células

align-self



Grid

alinhamento das células

align-self



Grid

alinhamento das células

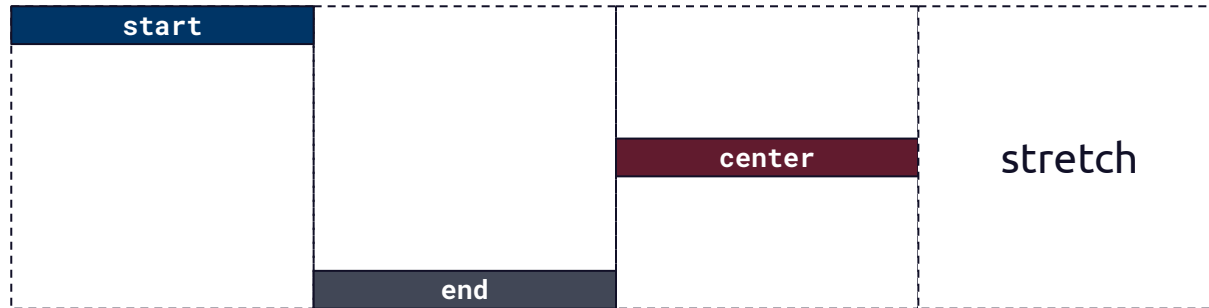
align-self



Grid

alinhamento das células

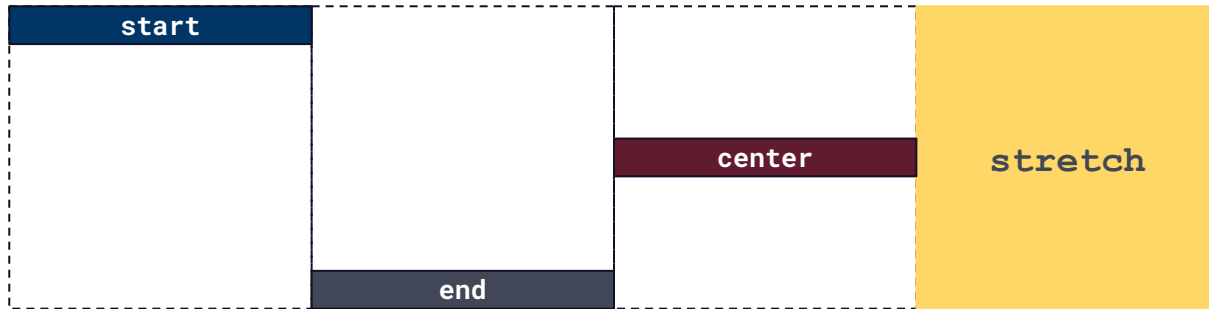
align-self



Grid

alinhamento das células

align-self



Grid

justify-content

| | |
|--------------------|--|
| valores | start end left right center stretch space-around space-between space-evenly |
| padrão | start |
| se aplica a | contêineres grid |
| herança | não |

- alinhamento do conteúdo no eixo principal

Grid

align-content

valores start | end | left | right | center | stretch |
 space-around | space-between | space-evenly

padrão start

se aplica a contêineres grid

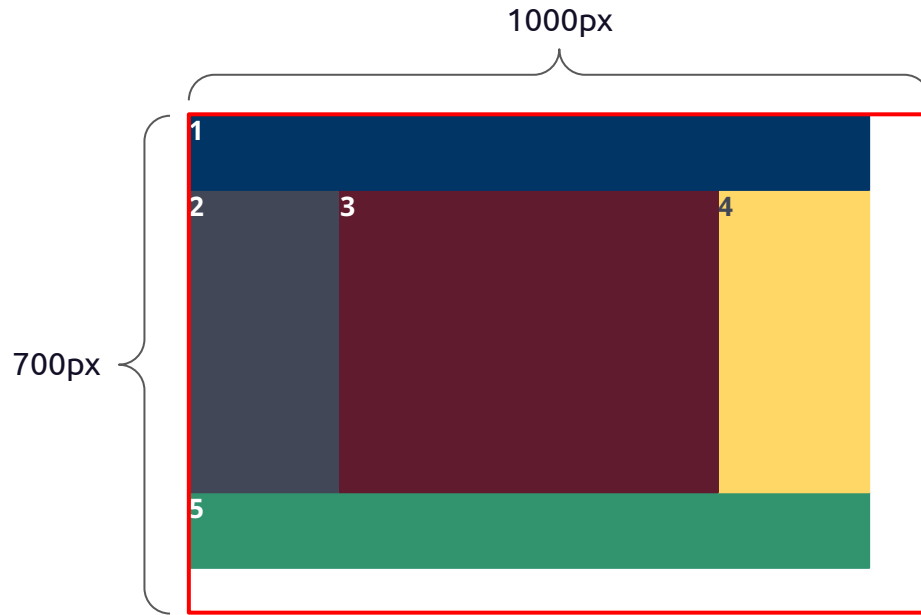
herança não

■ alinhamento do conteúdo no eixo cruzado



Grid

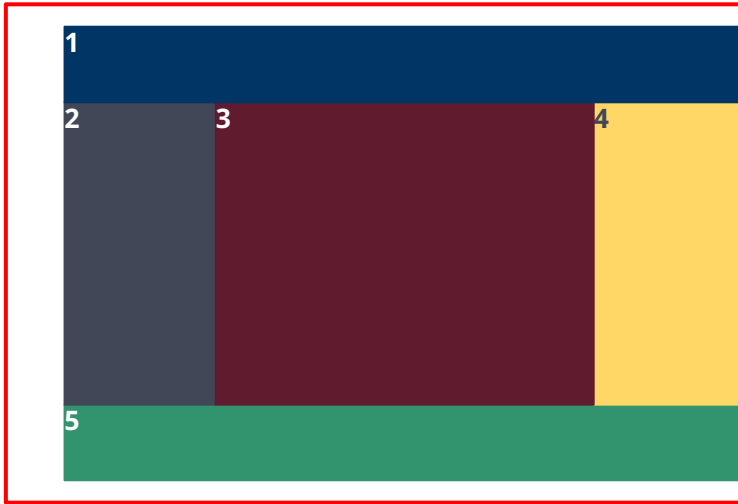
espaçamento entre as células



```
width: 1000px;  
height: 700px;  
grid-template-rows: 100px 400px 100px;  
grid-template-columns: 200px 500px 200px;  
justify-content: start;  
align-content: start;
```

Grid

espaçamento entre as células



```
justify-content: right;  
align-content: center;
```

Grid

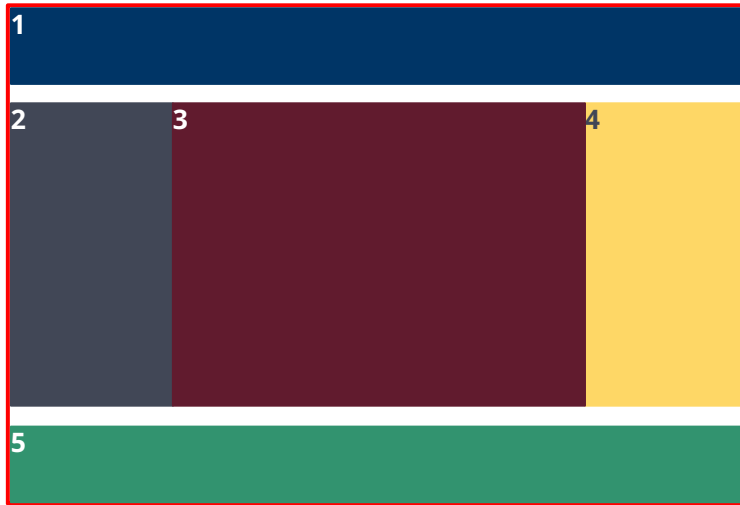
espaçamento entre as células



```
justify-content: stretch;  
align-content: space-between;
```

Grid

espaçamento entre as células

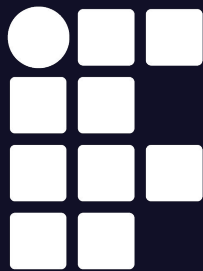


```
justify-content: stretch;  
align-content: space-between;
```

Exercícios

1. Clone o repositório enviado na aulas anterior.
2. Crie uma nova página. Ela deve ser acessado a partir da página inicial através de um link “Acesse minha loja”.
3. Coloque uma barra superior fixa, que esteja sempre visível quando a tela rolar. Essa barra pode ser um menu ou um título, por exemplo.
4. No restante da página faça um grid de produtos. Cada um deve conter uma imagem, seu nome e seu preço.
5. O grid deve ter pelo menos 20 produtos
6. Distribua os produtos no grid (linhas e colunas) como desejar.
7. O documento deve ser validado!





**INSTITUTO
FEDERAL**
Santa Catarina

Posicionando elementos com CSS (parte 2)

Programação Frontend I

Análise e Desenvolvimento de Sistemas

Prof. Adriano Lima

adriano.lima@ifsc.edu.br



INSTITUTO FEDERAL

Santa Catarina

Câmpus São José

PENSAMENTO
COMPUTACIONAL E
ALGORITMOS

Análise e Desenvolvimento
de Sistemas