CSS
Posicionando Elementos

FUTURE4

Big Picture

Lembrando...

- Posicionar elementos com CSS pode ser **bem** chato
- Para atingir um mesmo resultado, existem várias estratégias
- Novas tecnologias ajudam, mas não resolvem todos os problemas

Flexbox e Grid

FUTURE4


Finally.

Visão geral

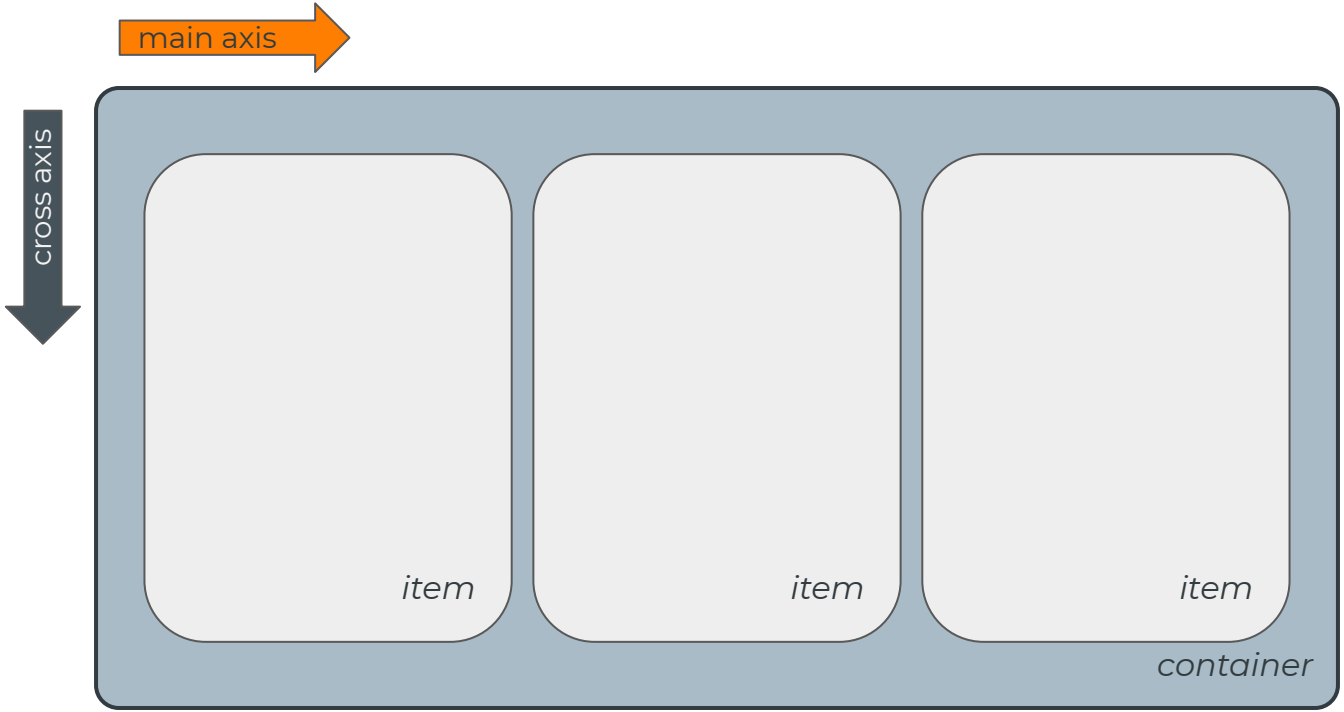
- **Novas** tecnologias que surgiram para facilitar a criação de layouts **complexos** e **responsivos** com mais facilidade

- Semelhantes, mas possuem vantagens e desvantagens

- Não são compatíveis com navegadores antigos

Flexbox

Estrutura



FUTURE4

Boilerplate

```
<div class="container">
  <div class="item" id="item1">W: 100</div>
  <div class="item" id="item2">W/H: 80</div>
  <div class="item" id="item3">H: 300</div>
</div>
```

```
div {
  border: 2px solid #313B3E;
  border-radius: 10px;
}
.container {
  max-width: 600px;
  background-color: #A8BBC6;
}
.item {
  background-color: #EEEEEE;
}
#item1 {
  width: 100px;
}
#item2 {
  width: 80px;
  height: 80px;
}
#item3 {
  height: 300px;
}
```

FUTURE4

Boilerplate

FUTURE4

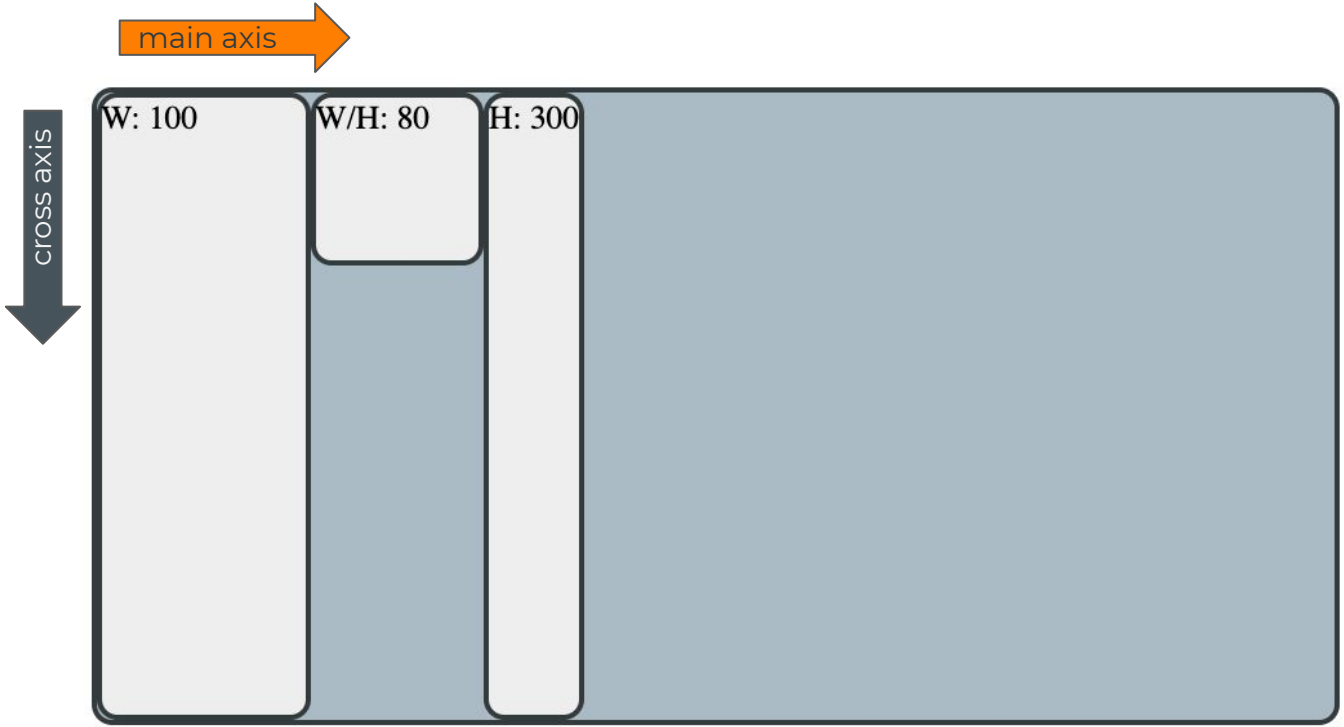
Propriedades do **container**

FUTURE4

Display *flex*

- Comporta-se como **block-level** element
- Afeta o comportamento dos elementos **diretamente** filhos

Display *flex*



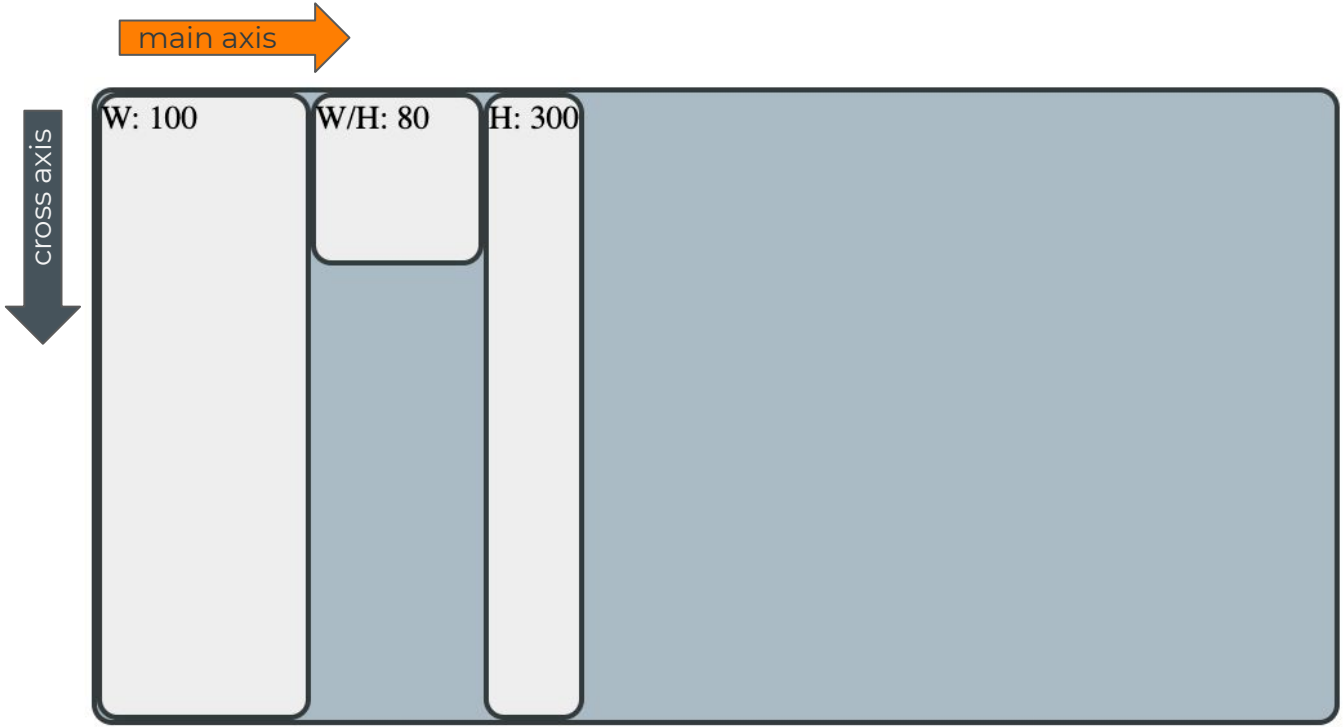
FUTURE4

flex-direction

- Altera a direção dos **eixos** (axis)

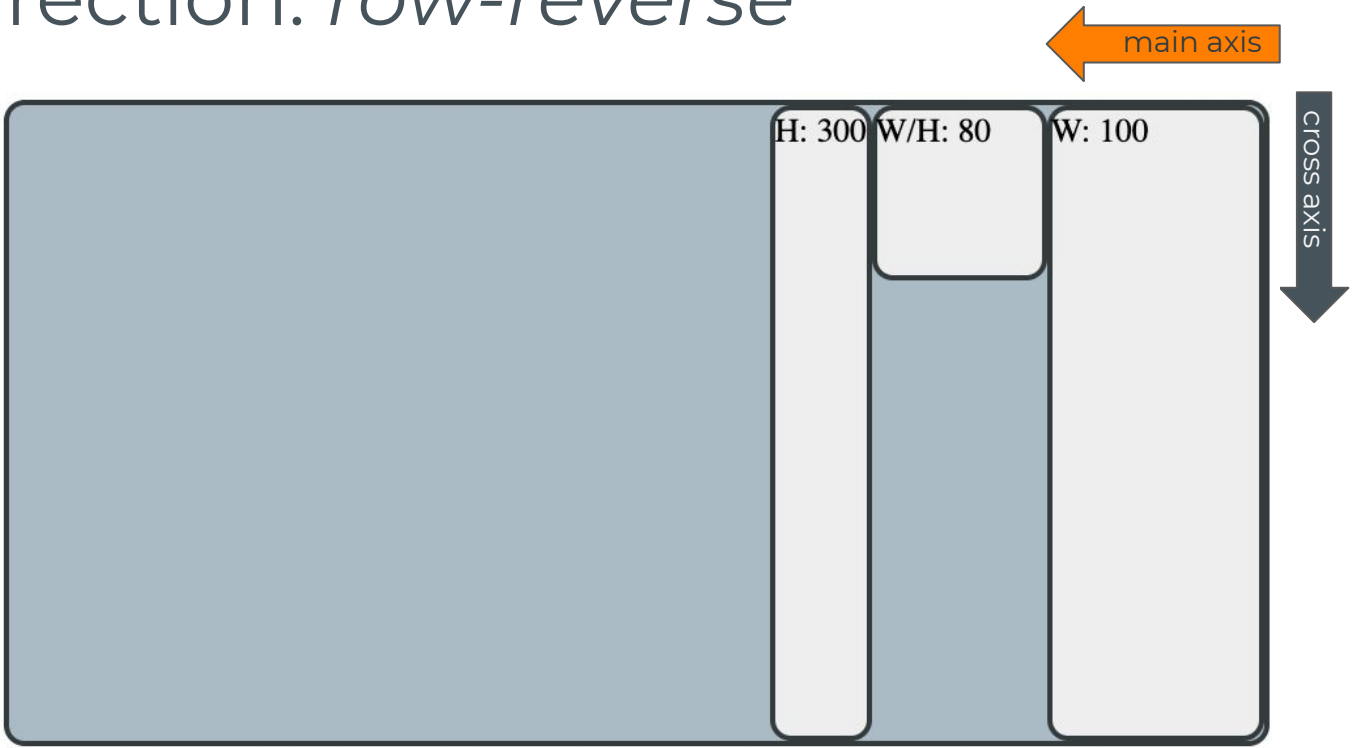
- Valores **possíveis**:
  - ***row***
  - *column*
  - *row-reverse*
  - *column-reverse*

flex-direction: *row*



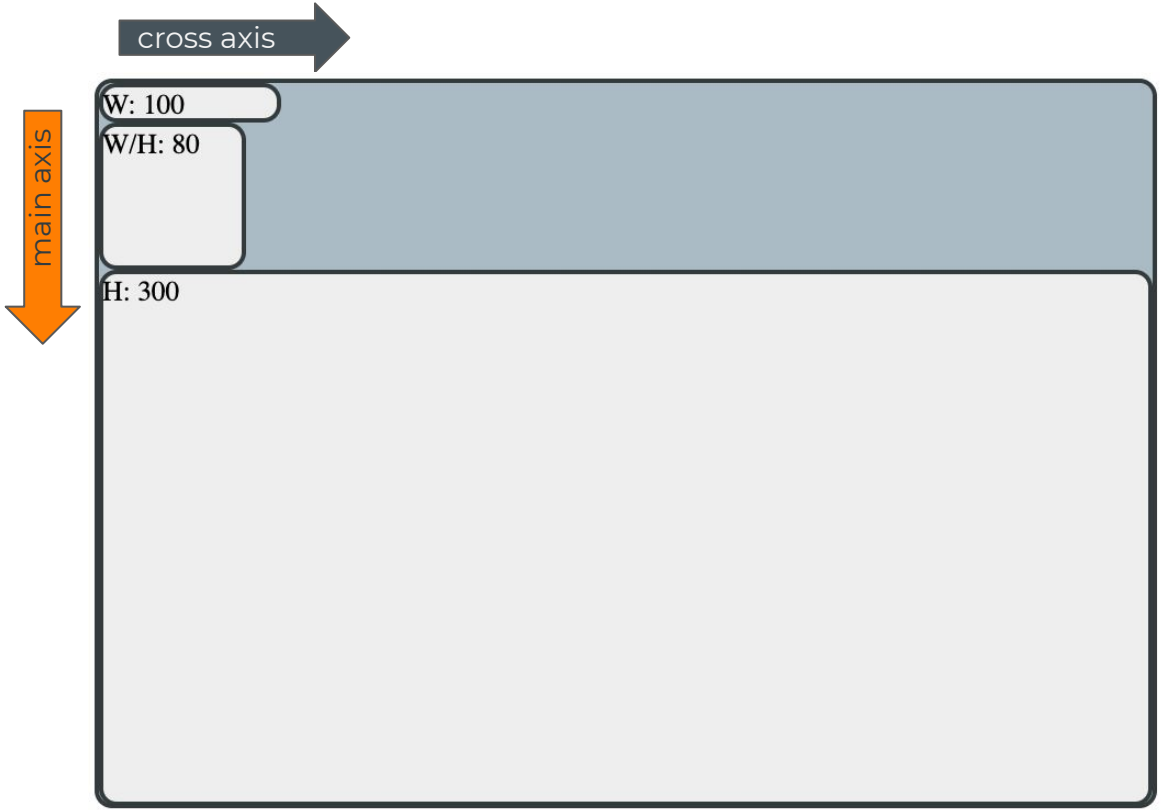
FUTURE4

flex-direction: *row-reverse*



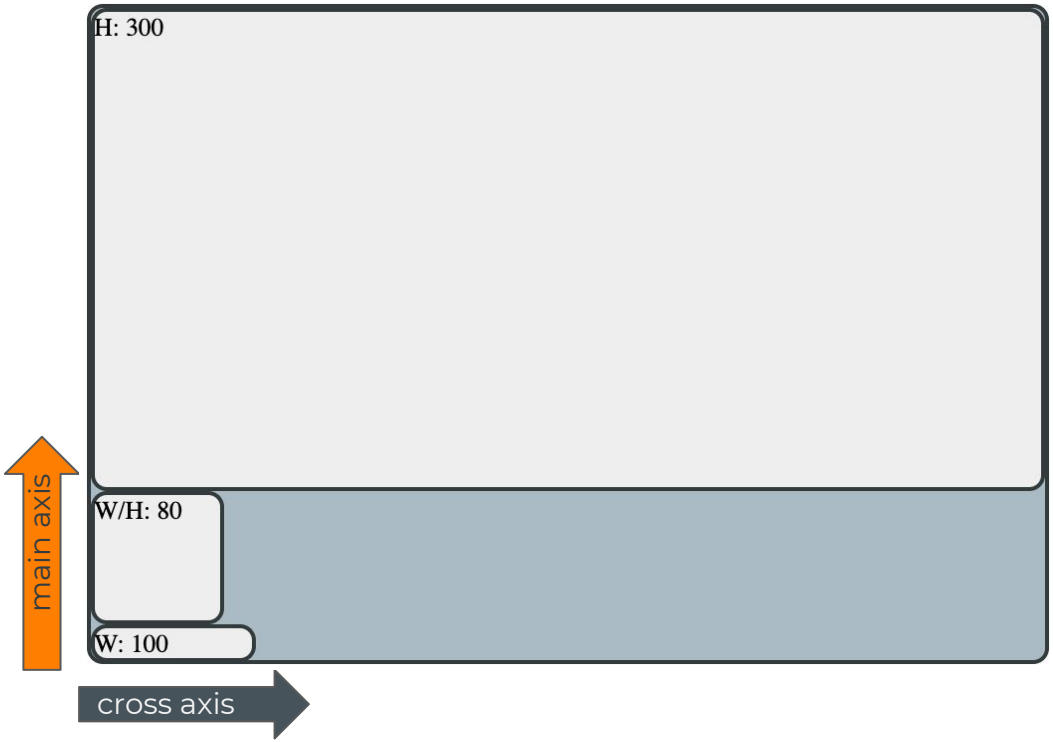
FUTURE4

flex-direction: *column*



FUTURE4

flex-direction: *column-reverse*



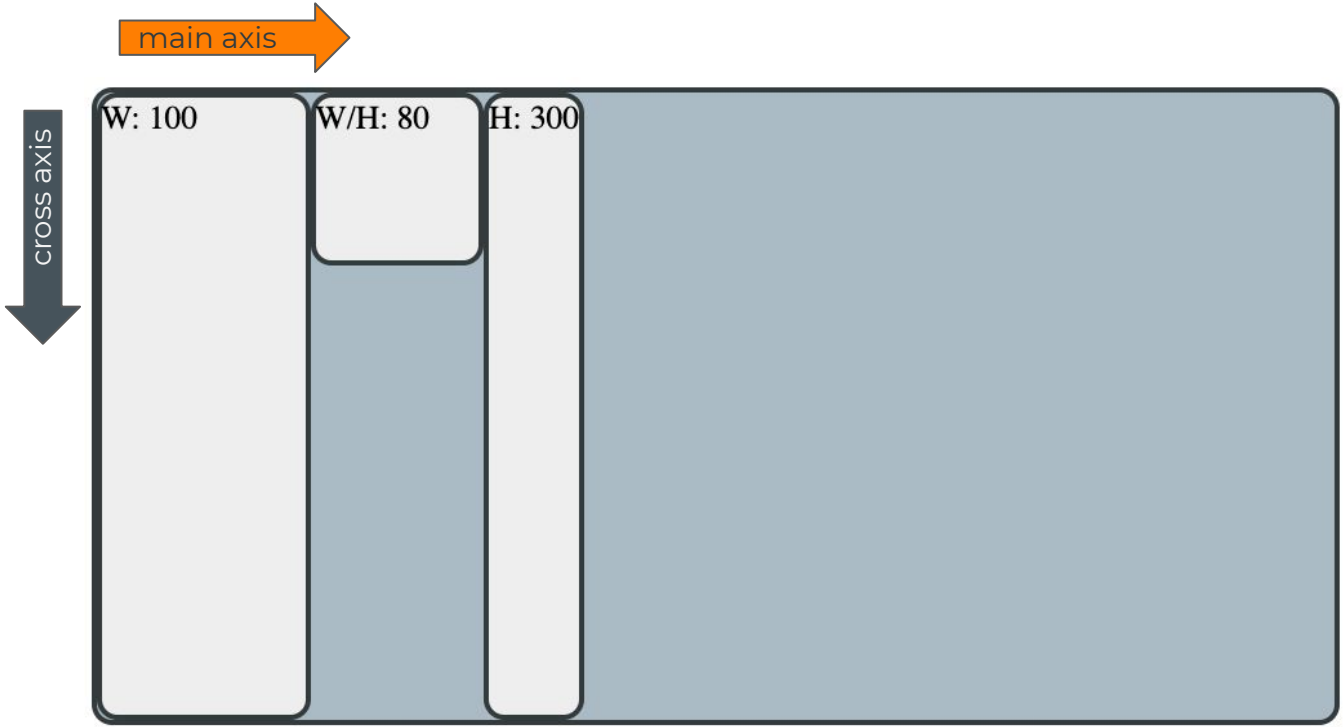
FUTURE4

justify-content

- Define a **disposição** dos elementos no **main axis**

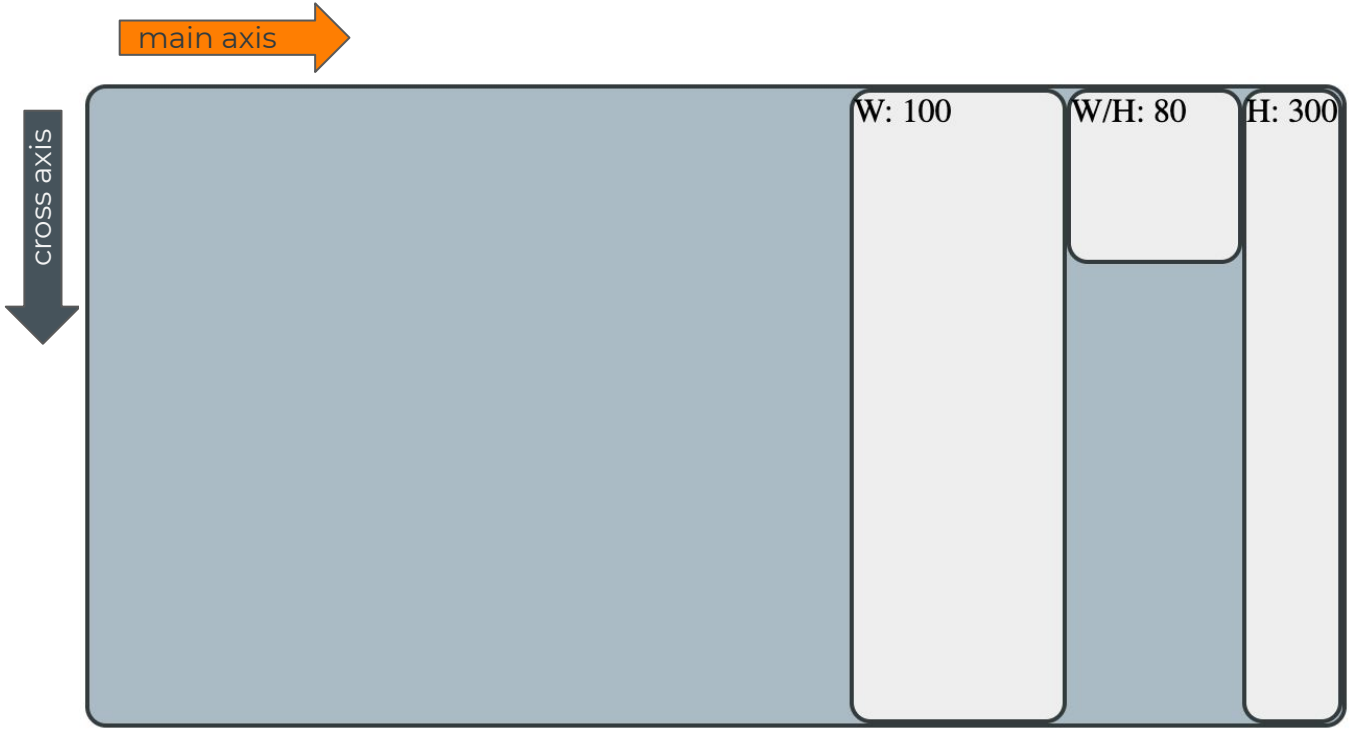
- Valores **possíveis**:
  - ***flex-start***
  - *flex-end*
  - *center*
  - *space-between*
  - *space-around*
  - *space-evenly*

justify-content: *flex-start*



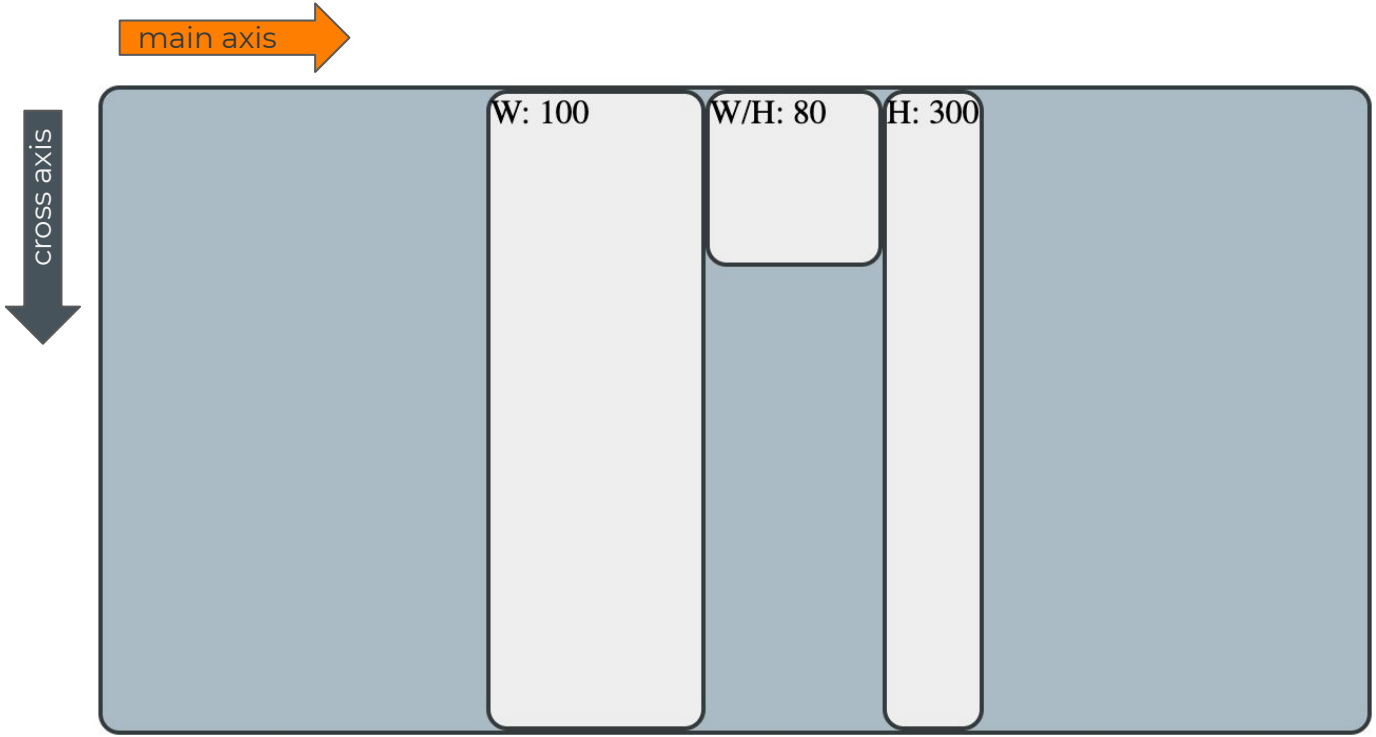
FUTURE4

justify-content: *flex-end*



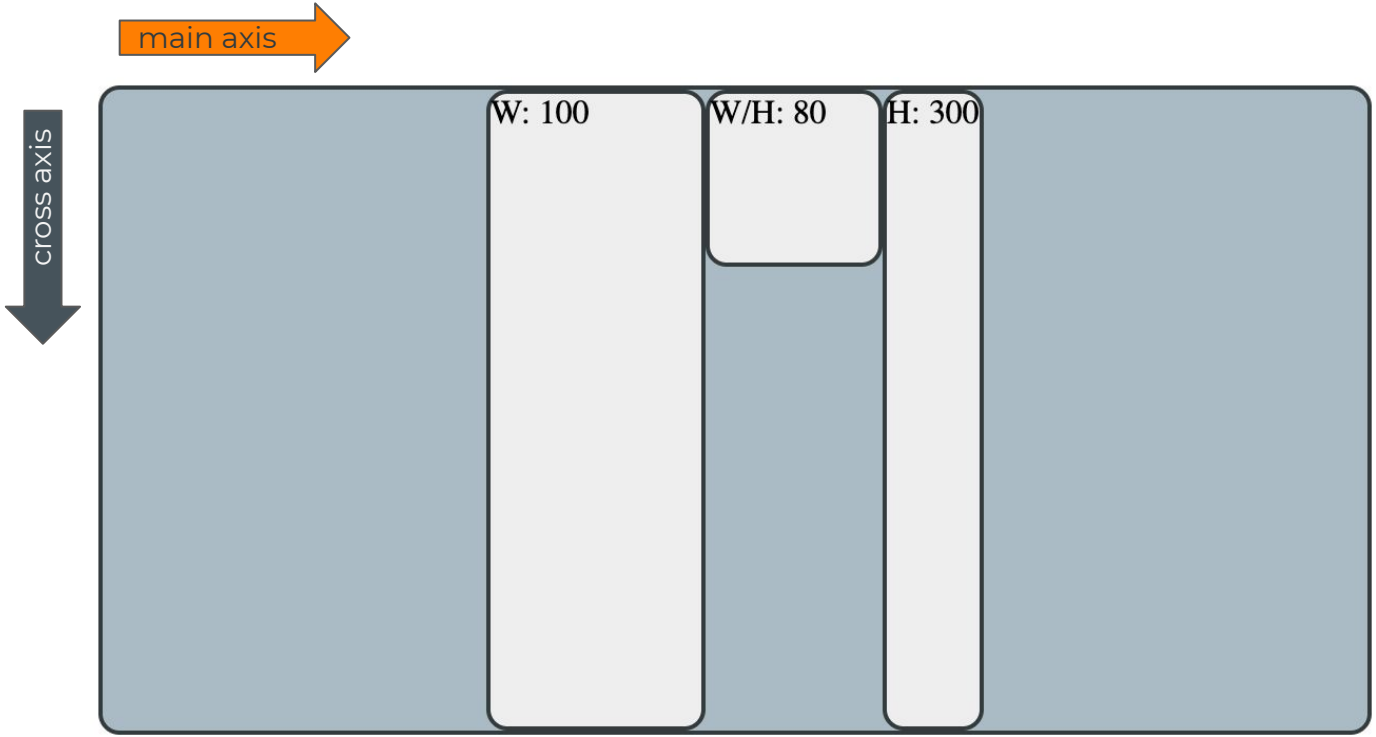
FUTURE4

justify-content: *center*



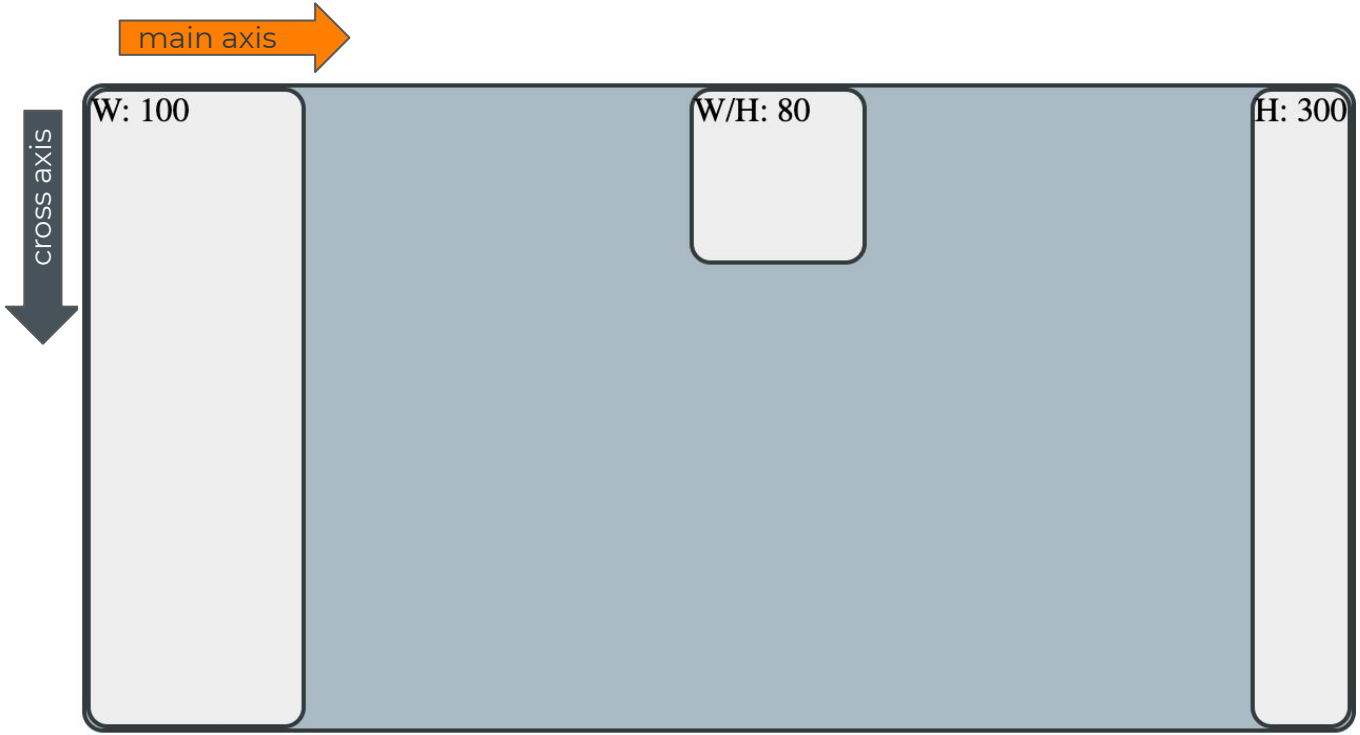
FUTURE4

justify-content: *space-between*



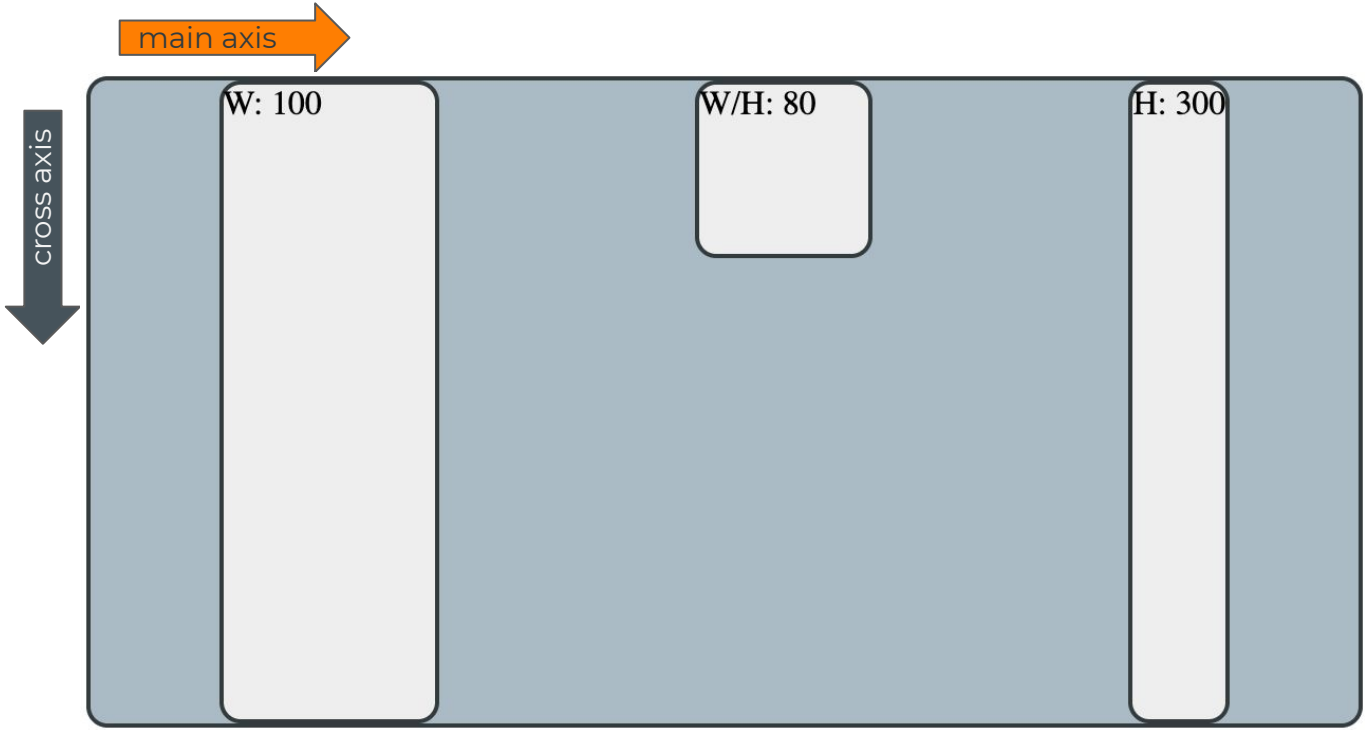
FUTURE4

justify-content: *space-between*



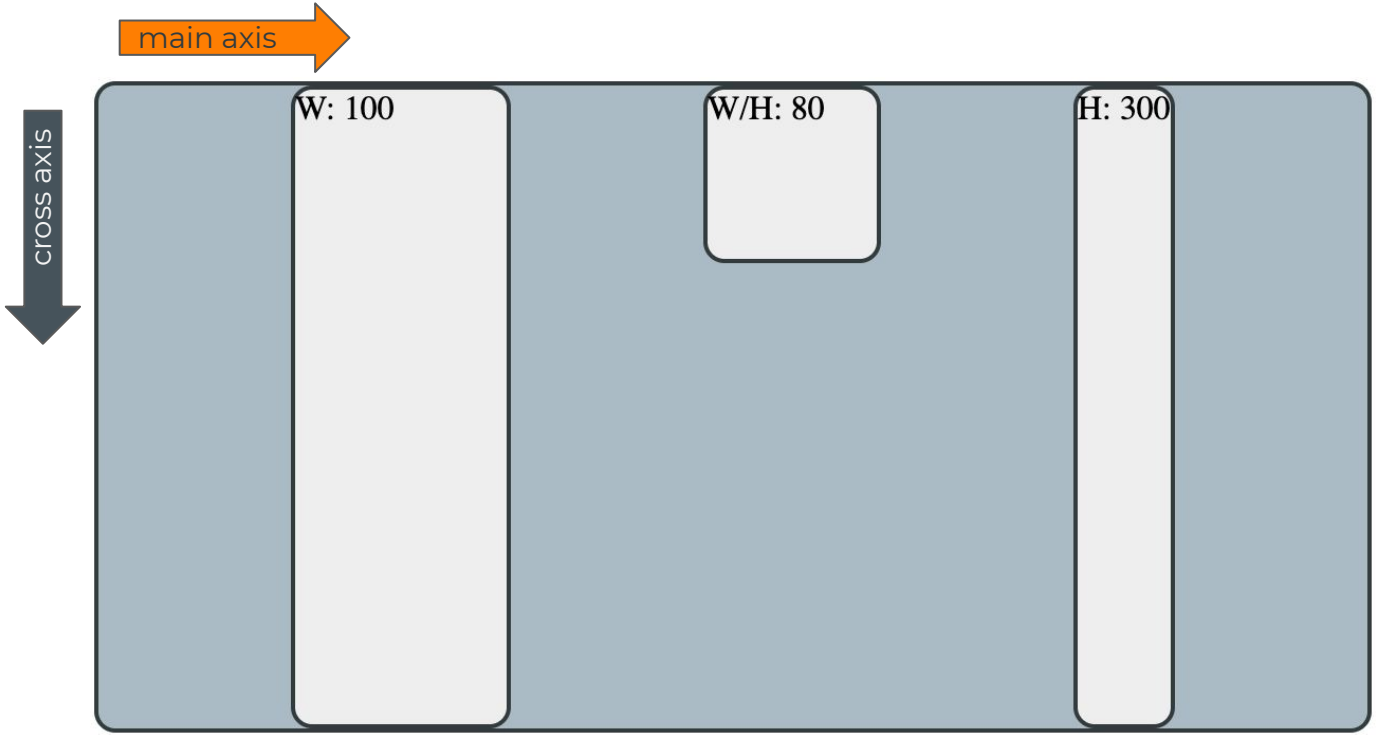
FUTURE4

justify-content: *space-around*



FUTURE4

justify-content: *space-evenly*



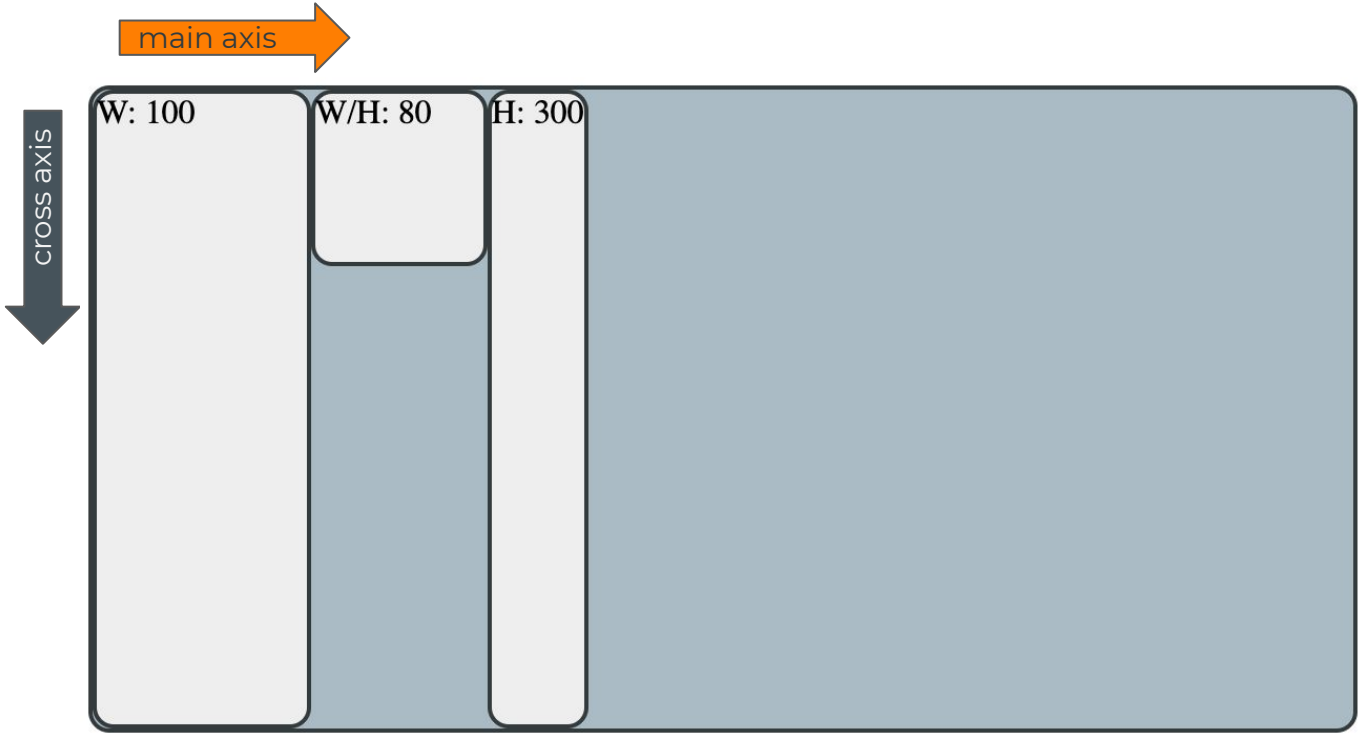
FUTURE4

align-items

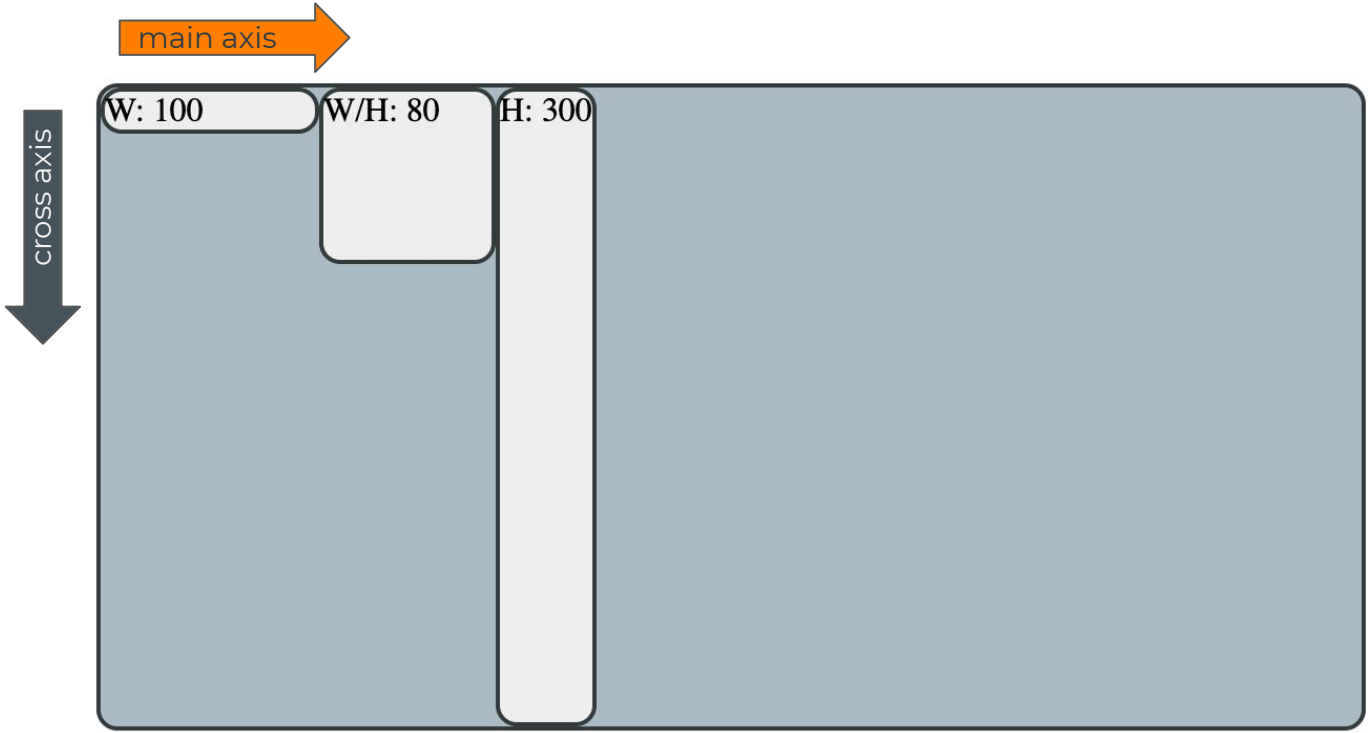
- Define a **posição** dos elementos no **cross axis**

- Valores **possíveis**:
  - ***stretch***
  - *flex-start*
  - *flex-end*
  - *center*
  - *baseline*

align-items: *stretch*

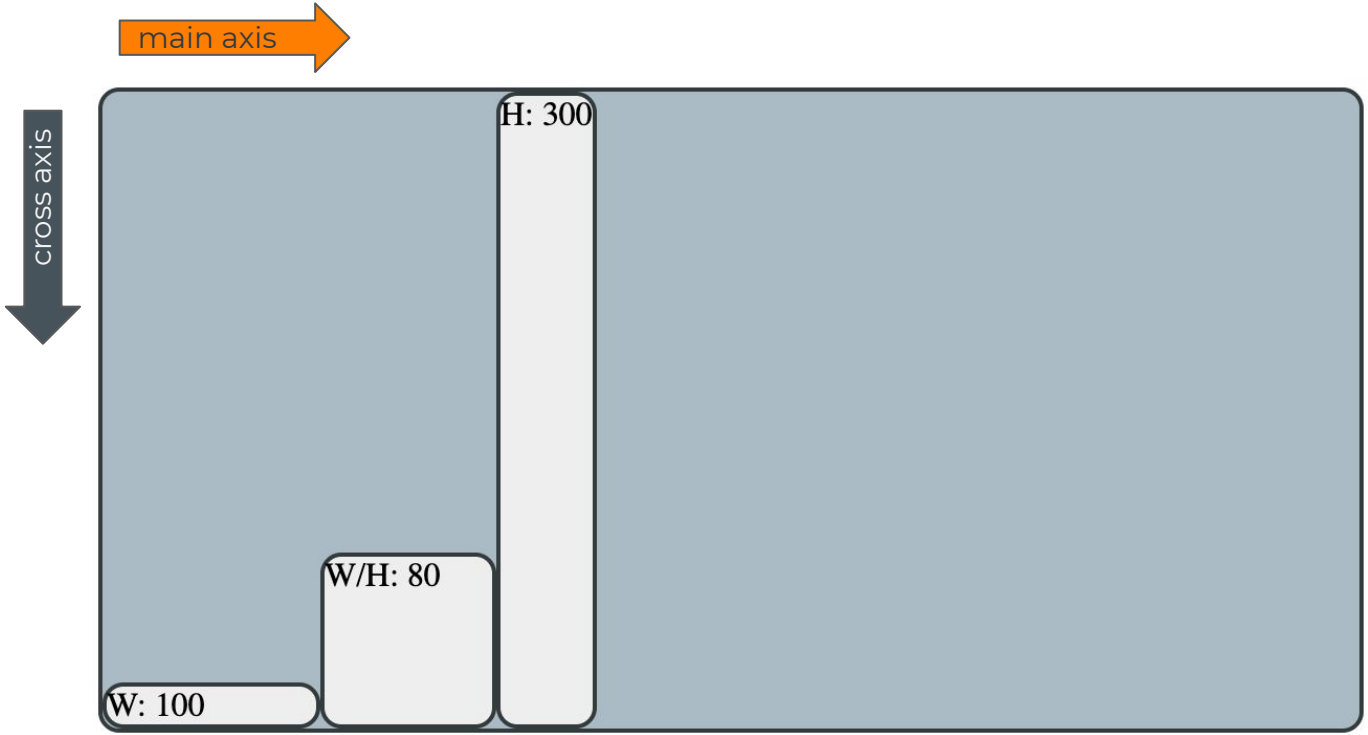


align-items: *flex-start*



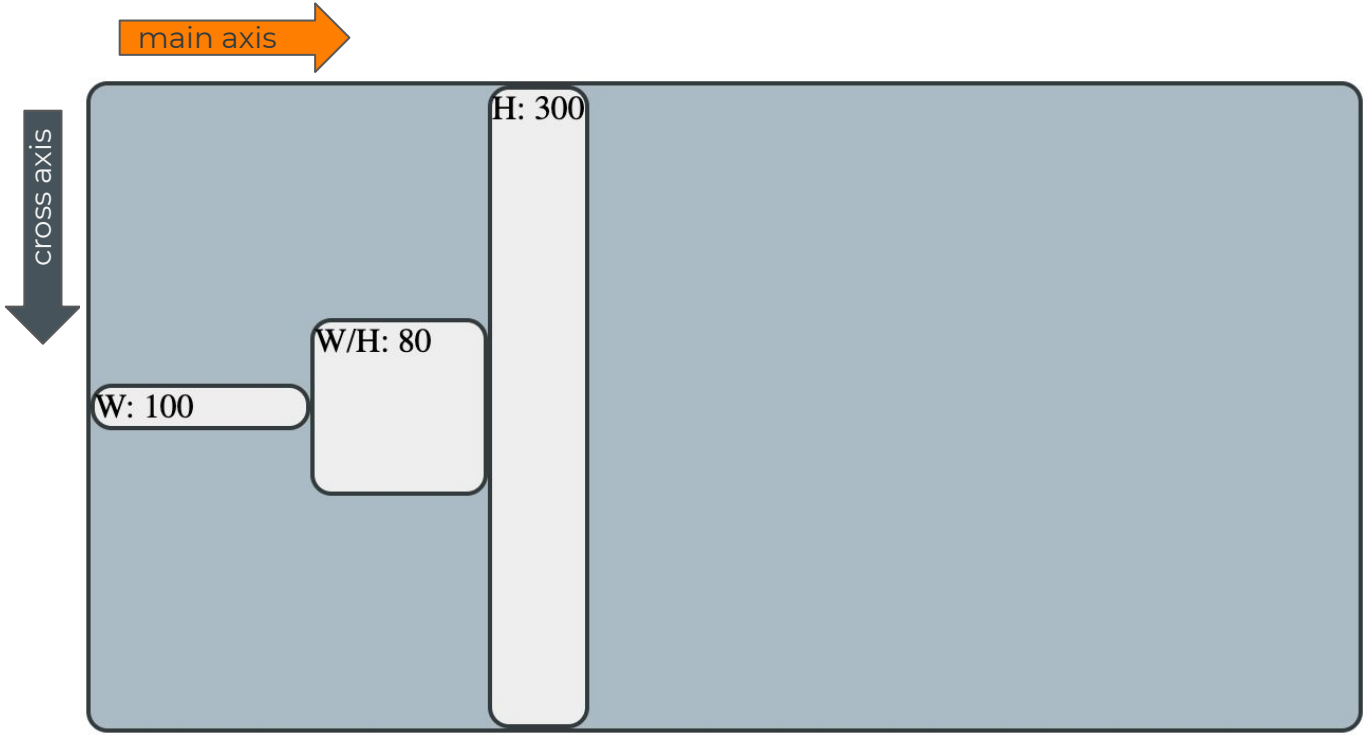
FUTURE4

align-items: *flex-end*

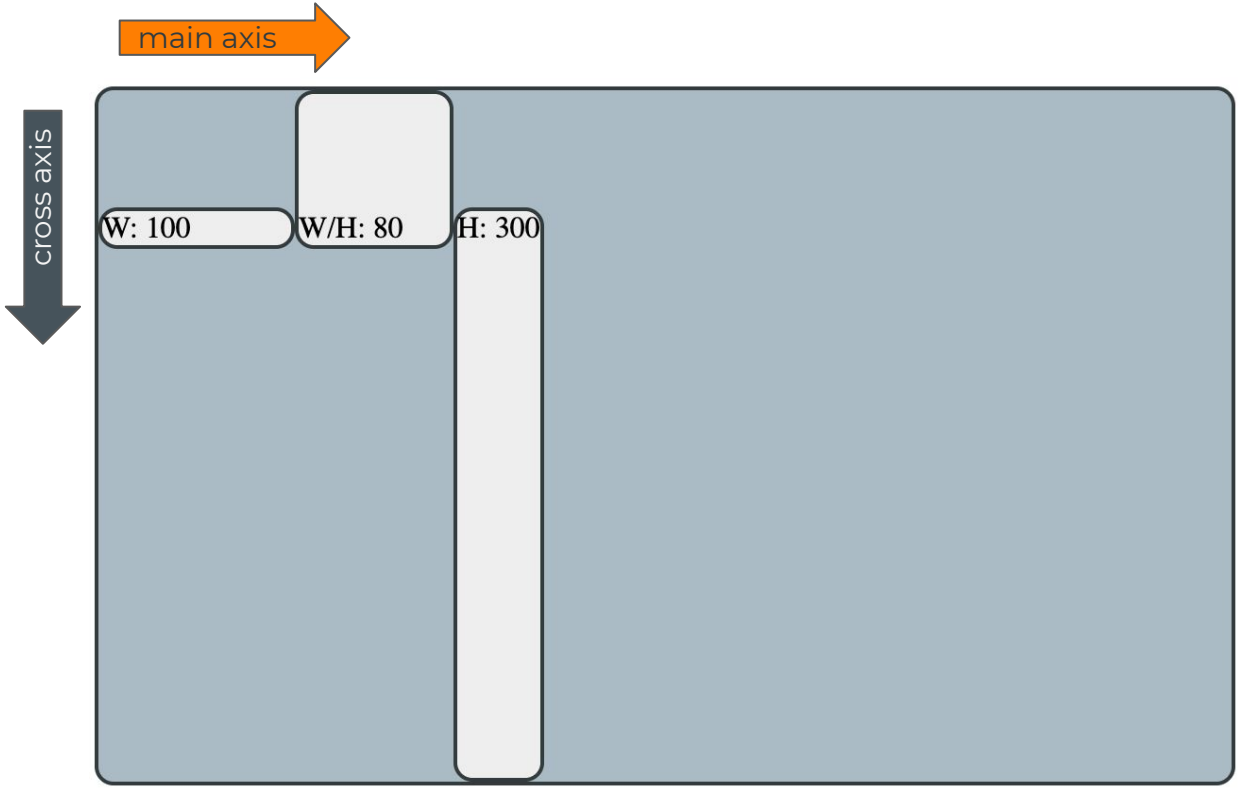


FUTURE4

align-items: *center*



align-items: *baseline*



FUTURE4

Pausa

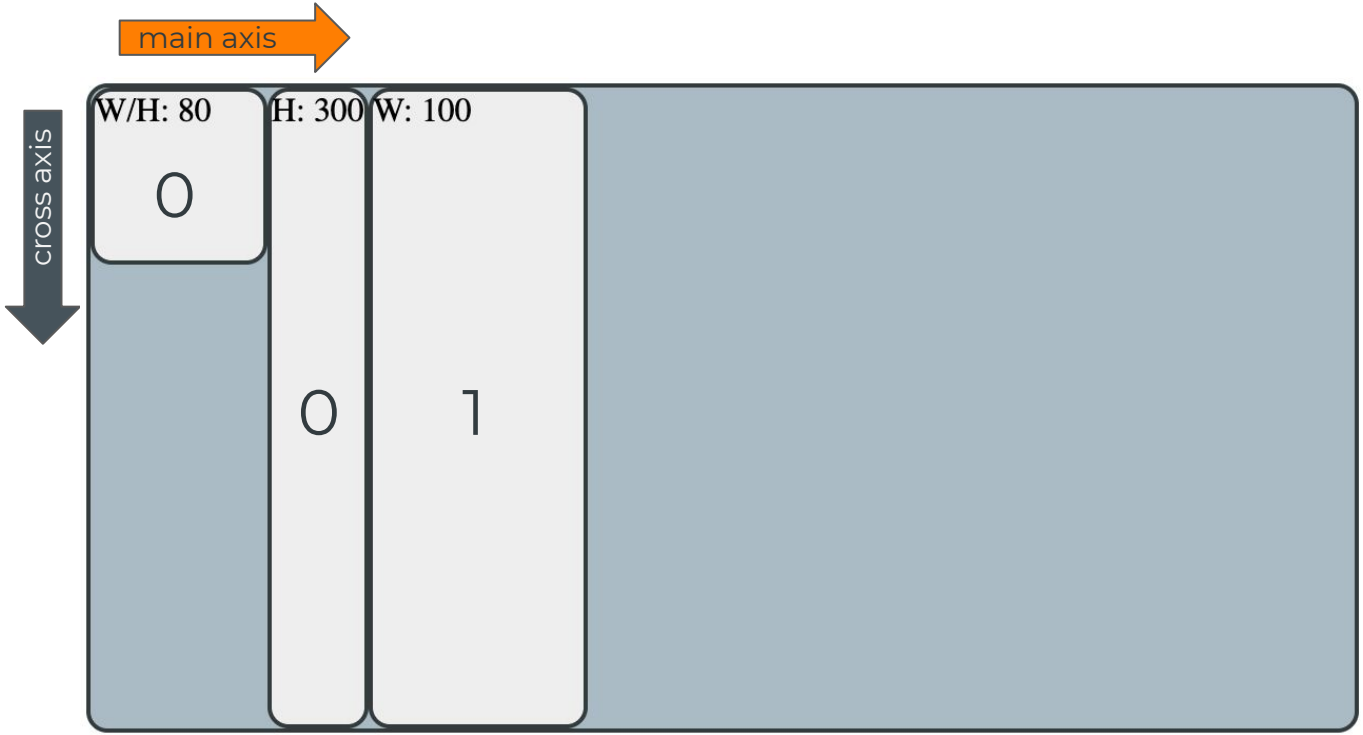
Propriedades do **item**

FUTURE4

order

- Define a **ordem** dos elementos no **main axis**.
- Quanto maior o número, mais pro final do eixo o item fica. Se dois itens tiverem o mesmo número, ficam no ordem do HTML.
- Padrão é **0**.

order



FUTURE4

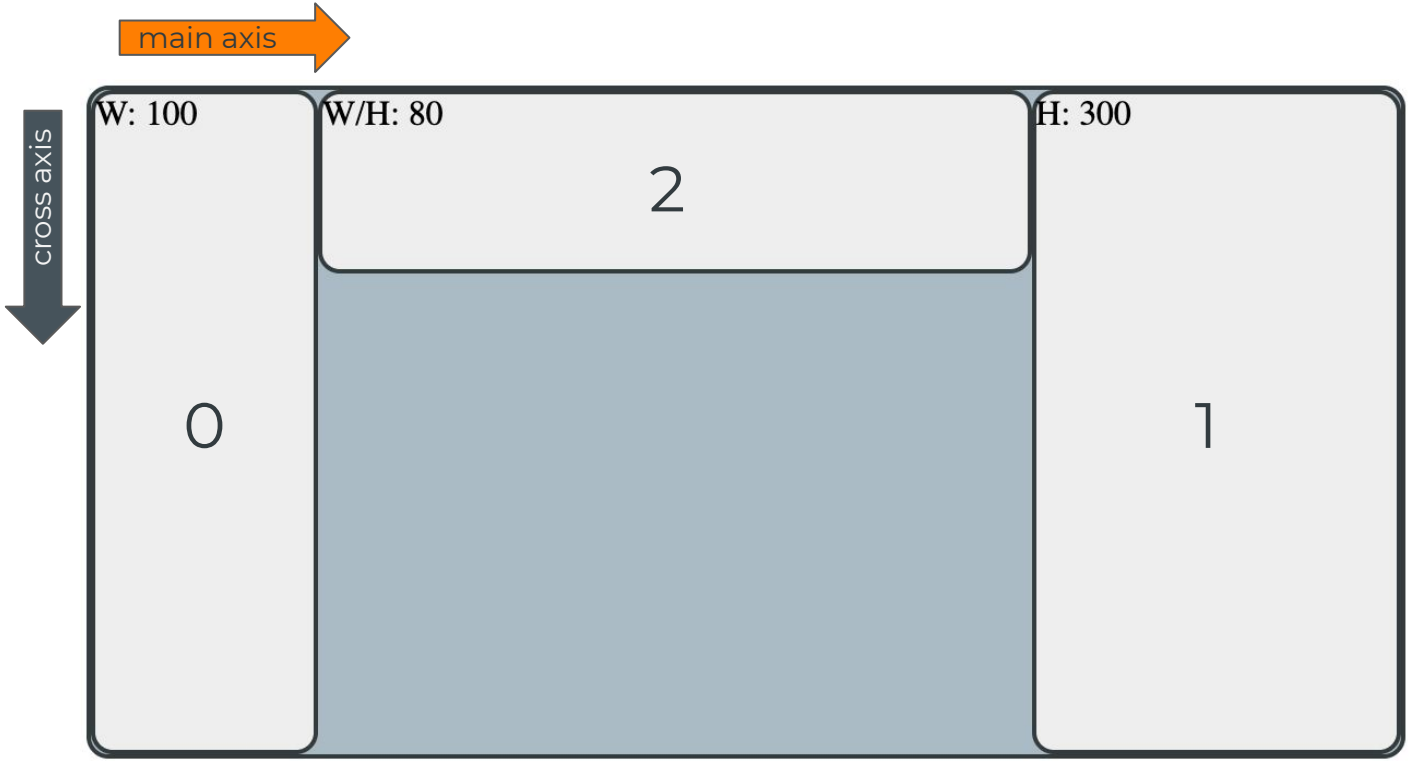
flex-grow

- Define a habilidade dos elementos de **crescer** para ocupar o espaço vazio no **main axis**.

- Quanto maior o número, mais o elemento irá crescer, proporcionalmente.

- Padrão é **0**.

flex-grow



FUTURE4

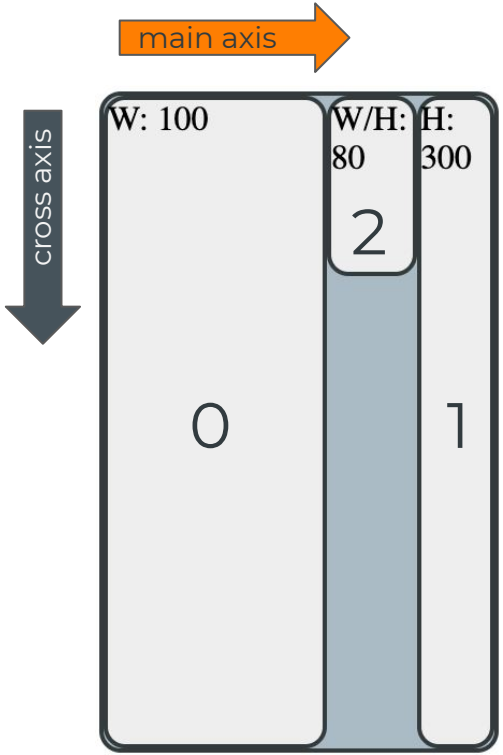
flex-shrink

- Define a habilidade dos elementos de **diminuir** para se ajustar à falta de espaço no **main axis**.

- Quanto maior o número, mais o elemento irá crescer, proporcionalmente.

- Padrão é **1**. (elementos diminuem por padrão)

flex-shrink



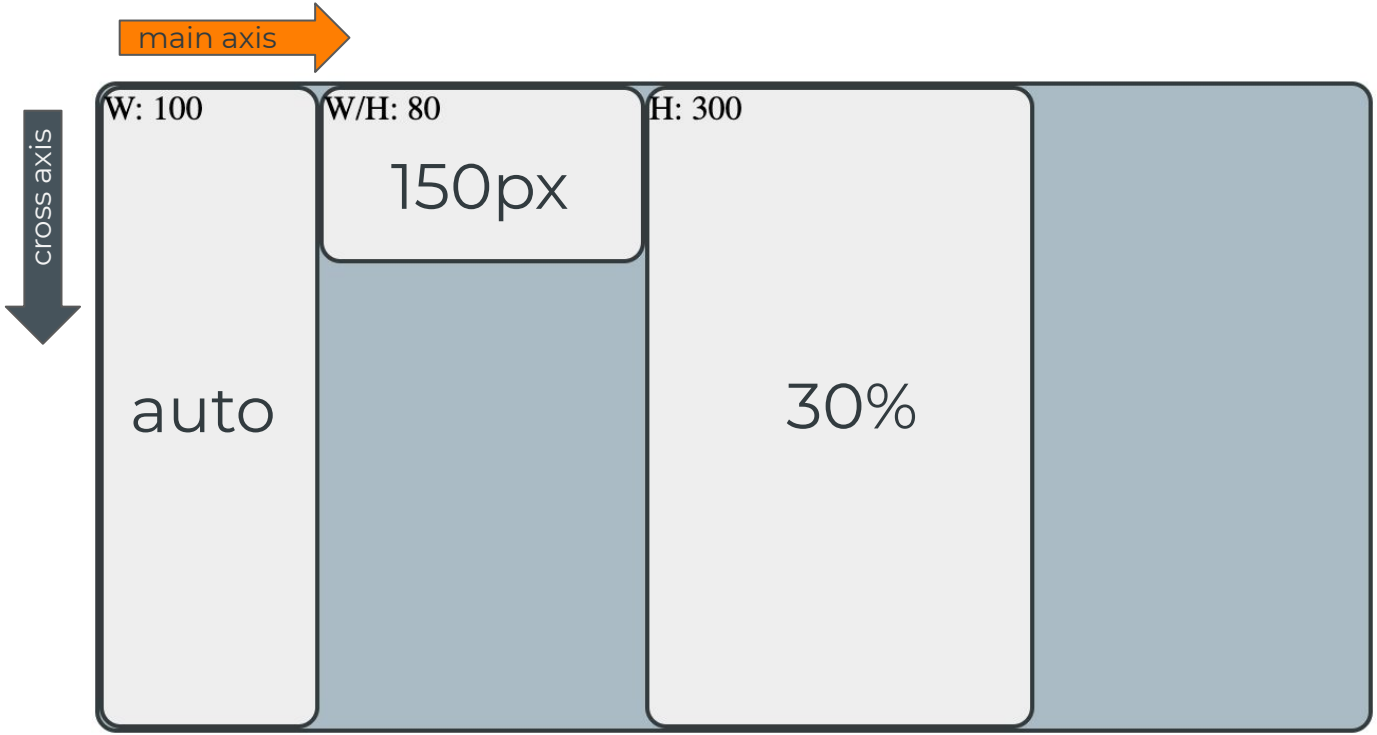
flex-basis

- Define o tamanho do elemento no **main-axis** antes de o espaço vazio ser distribuído.

- Valor padrão é **auto**, que deixa essa definição pra *width/height* (depende da *flex-direction*).

- Também pode assumir um *tamanho* (em px, %, etc.)

flex-basis



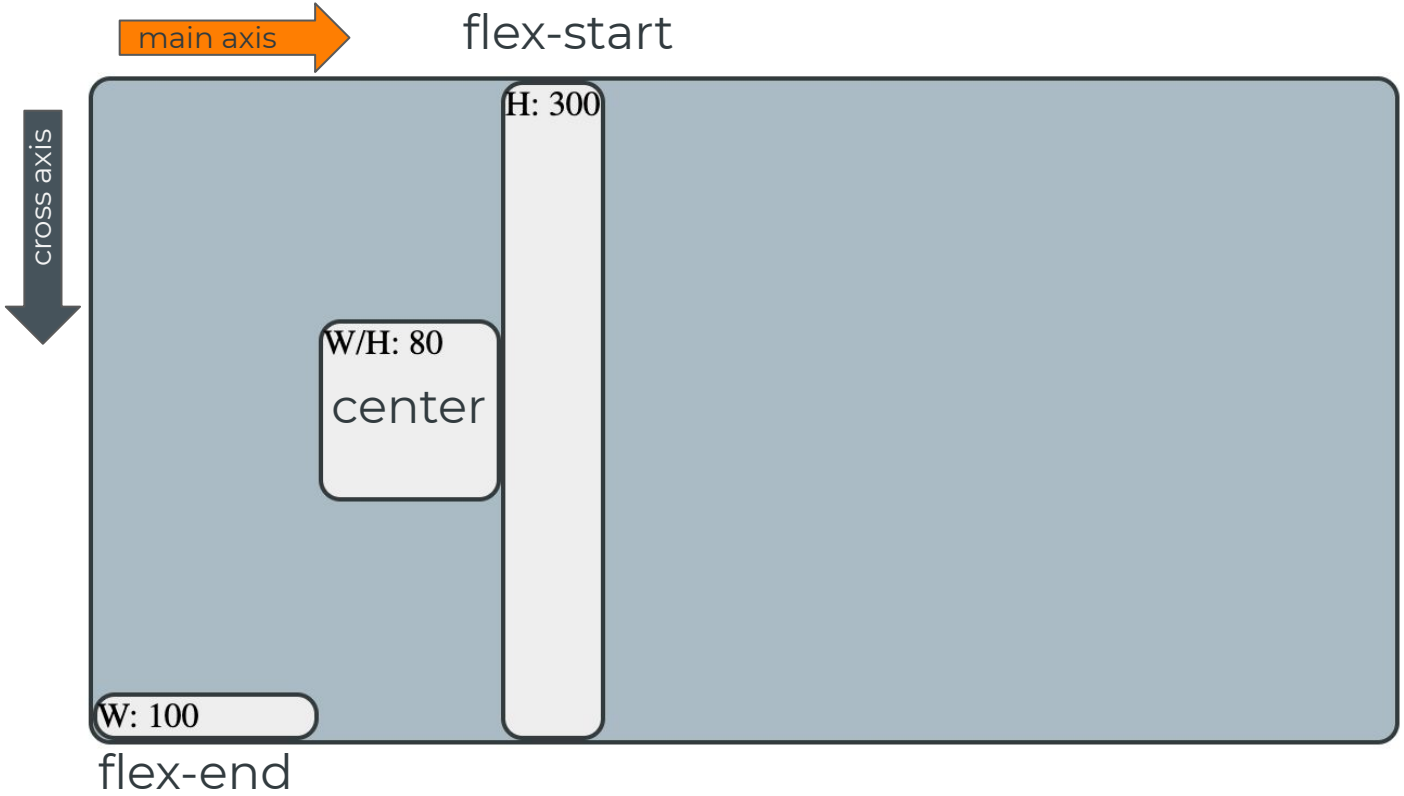
FUTURE4

align-self

- Define a **disposição** dos elementos no **cross axis**

- Valores **possíveis**:
  - ***auto***
  - *stretch*
  - *flex-start*
  - *flex-end*
  - *center*
  - *baseline*

align-self



FUTURE4

margin: auto e o *flexbox*

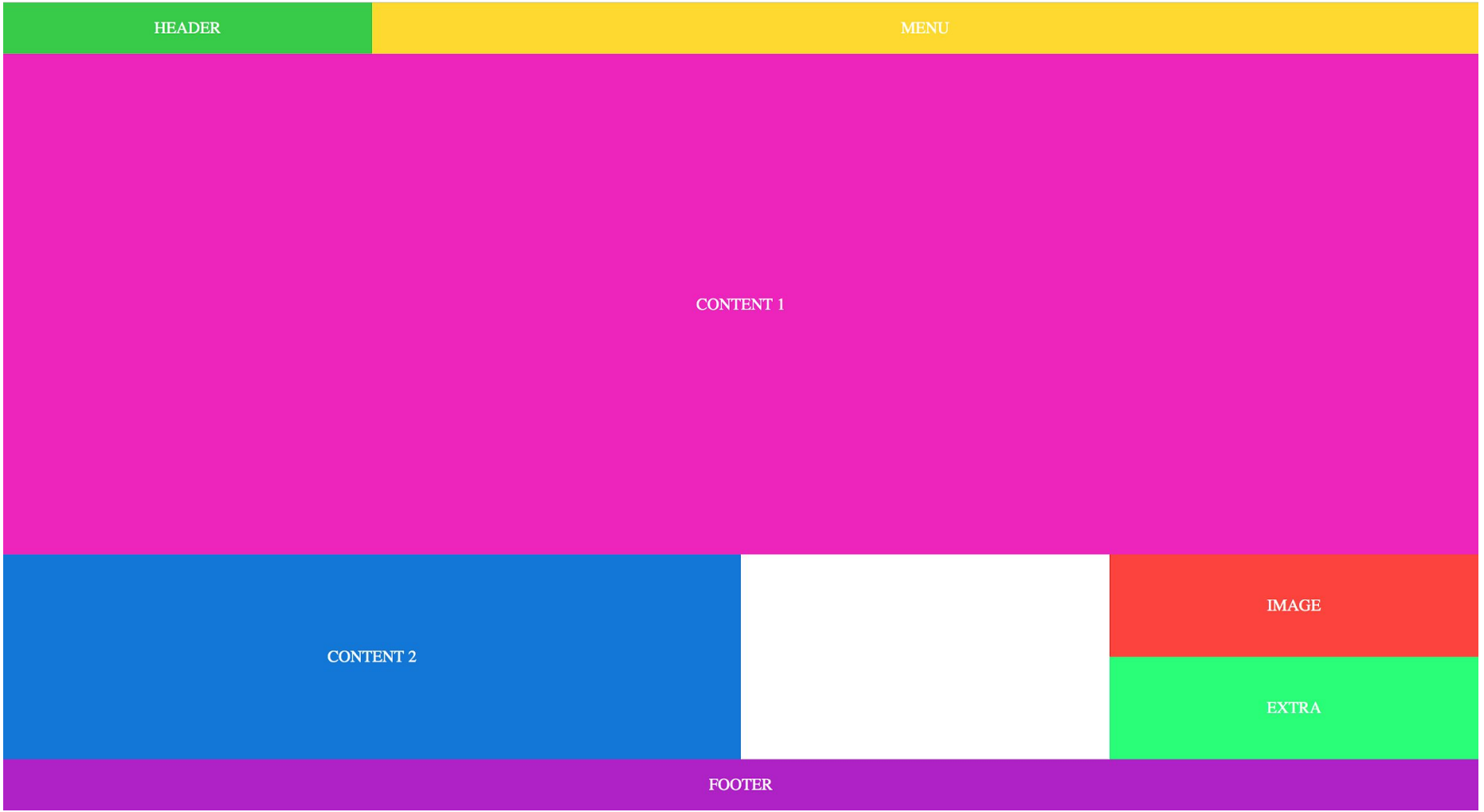
- margin: auto em itens do *flexbox* possuem um comportamento **curioso**

- Se margin: auto for usada em um item, as propriedades *justify-content* ou *aline-items* deixam de contar, dependendo do eixo em que a margem está como *auto*

- A margem toma todo o espaço disponível para si, “empurrando” os outros elementos

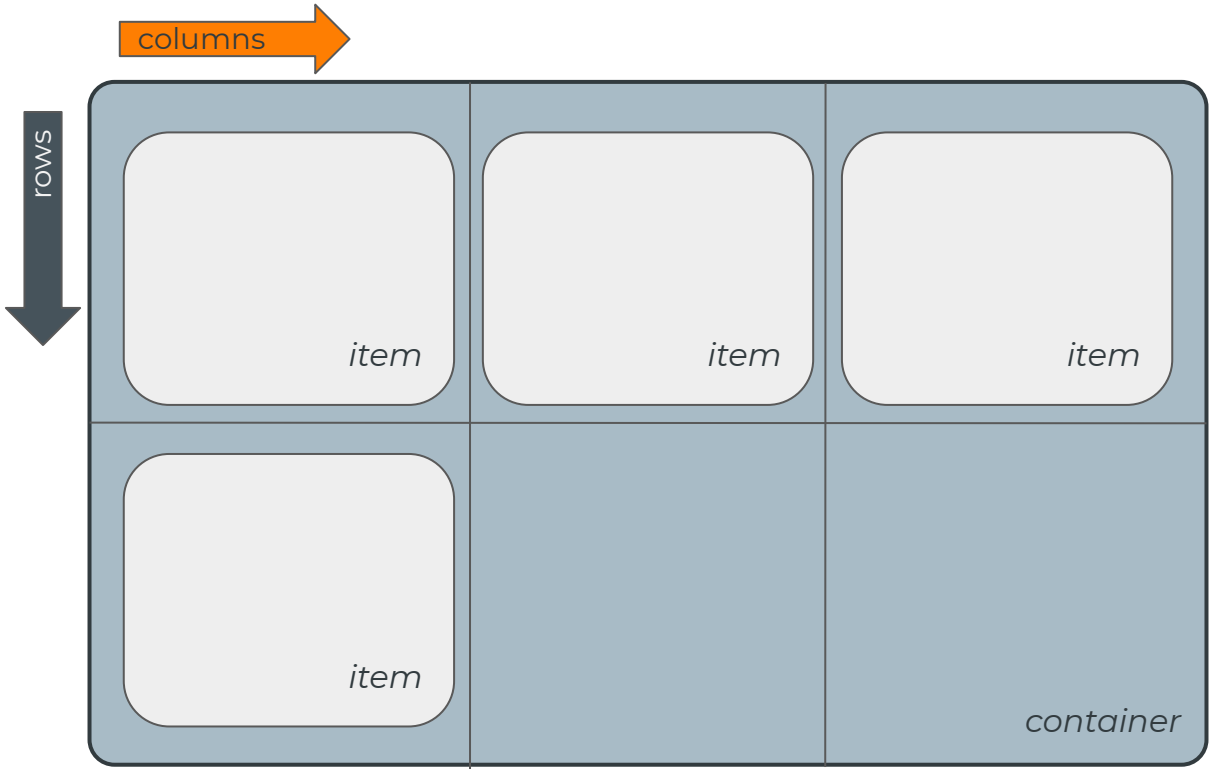
Pausa

Coding together


HEADER
MENU
CONTENT 1
CONTENT 2
IMAGE
EXTRA
FOOTER

Grid

Estrutura



FUTURE4

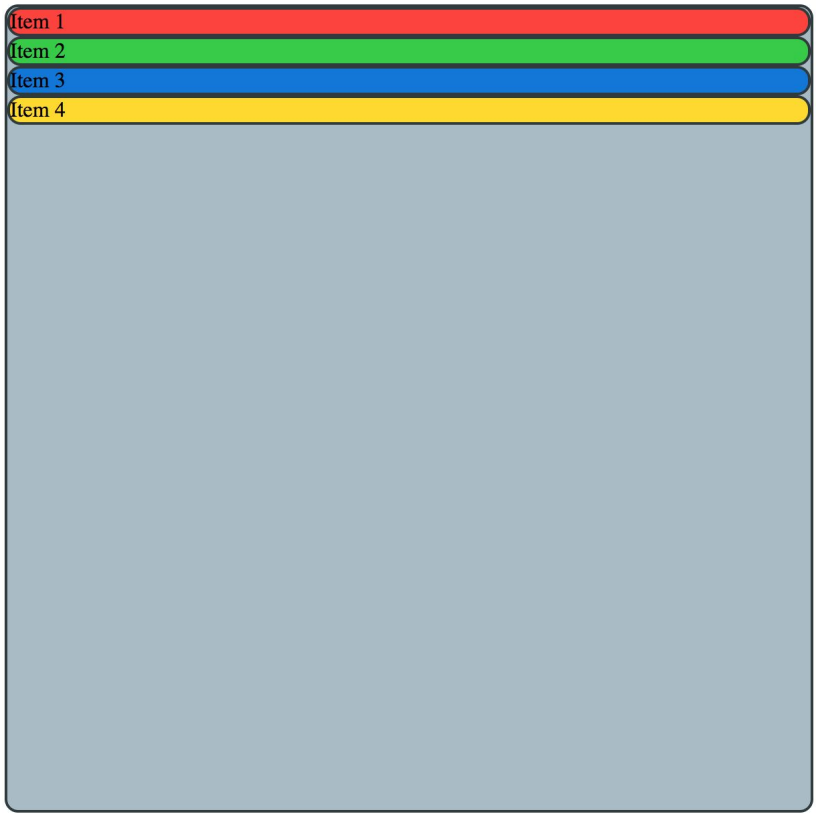
Boilerplate

```html
<div class="container">
  <div class="item" id="item1">Item 1</div>
  <div class="item" id="item2">Item 2</div>
  <div class="item" id="item3">Item 3</div>
  <div class="item" id="item4">Item 4</div>
</div>
```

```css
div {
  border: 2px solid #313B3E;
  border-radius: 10px;
}
.container {
  width: 600px;
  height: 600px;
  background-color: #A8BBC6;
}
#item1 {
  background-color: #FF4136; /*Vermelho*/
}
#item2 {
  background-color: #2ECC40; /*Verde*/
}
#item3 {
  background-color: #0074D9; /*Verde*/
}
#item4 {
  background-color: #FFDC00; /*Amarelo*/
}
```

FUTURE4

Boilerplate



FUTURE4

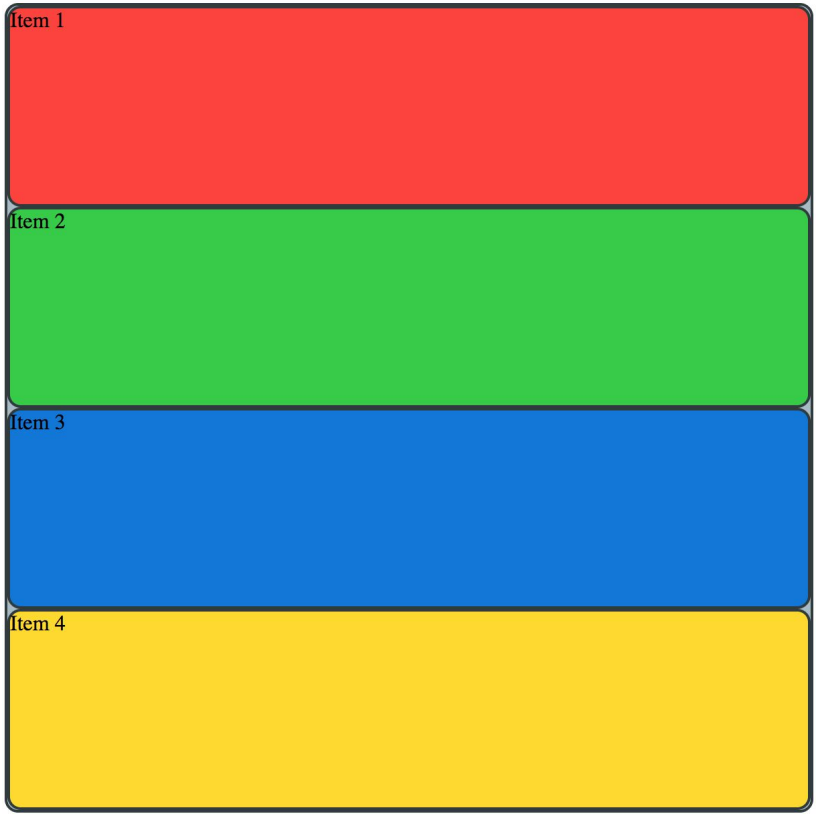
Propriedades do **container**

FUTURE4

Display *grid*

- Comporta-se como **block-level** element
- Afeta o comportamento dos elementos **diretamente** filhos

Display *grid*



FUTURE4

grid-template-[rows/columns]

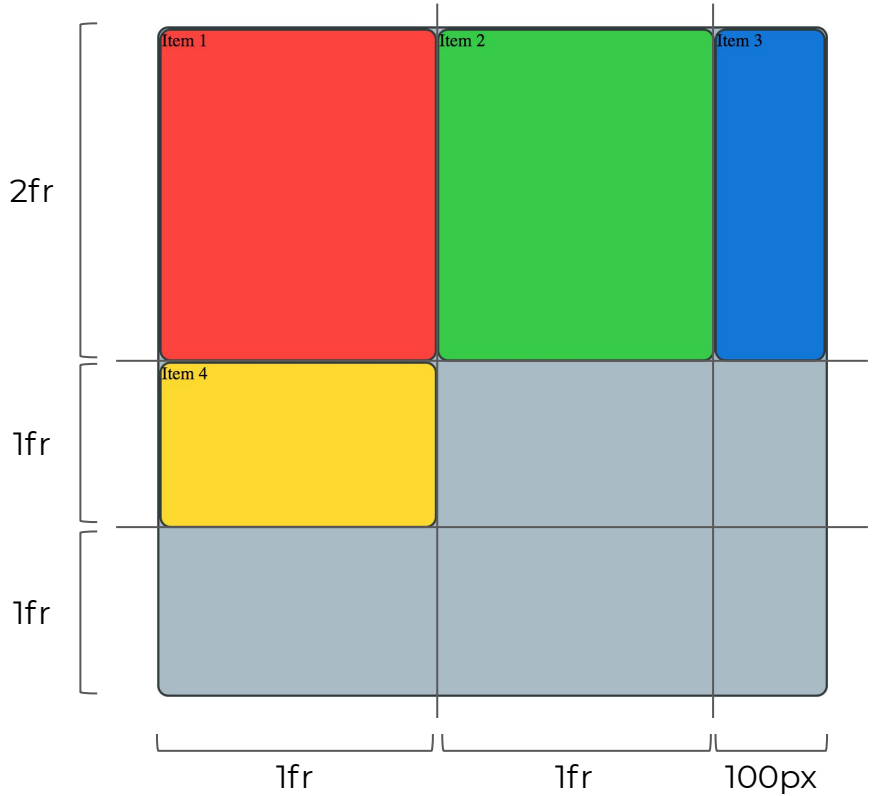
- Determina **template** das linhas e colunas, ou seja, qual o **tamanho** e **quantidade** de linhas e colunas.

- Recebe série de tamanhos separadas por espaço

- Nova unidade: **fr** - 1fr representa uma fração do espaço vazio

grid-template-rows: *2fr 1fr 1fr*

grid-template-columns: *1fr 1fr 100px*



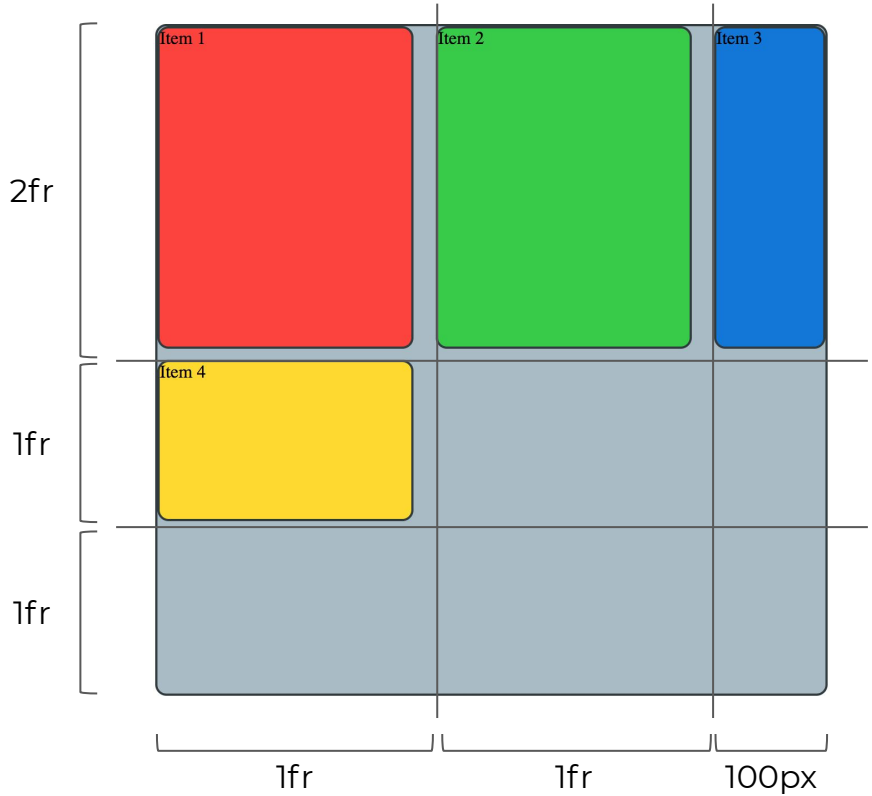
FUTURE4

grid-[row/column]-gap

- Determina **espaço** entre as linhas e colunas.
- Recebe um tamanho
- Novidade: nome mudou para [row/column]-gap

grid-row-gap: *10px*

grid-column-gap: *20px*



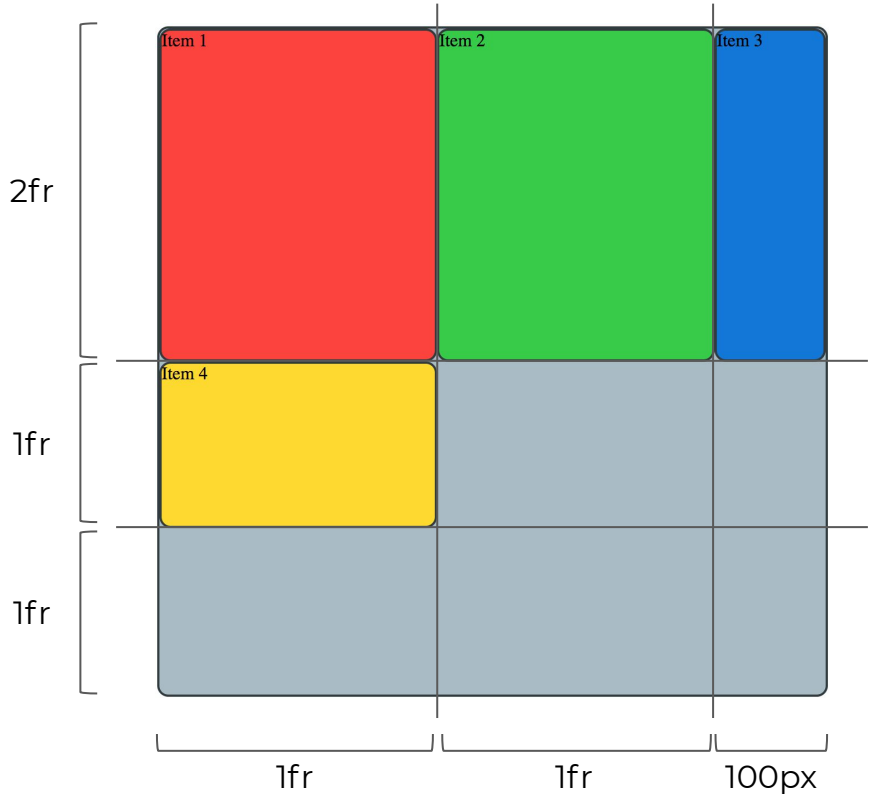
FUTURE4

justify-items

- Define a **posição** dos elementos no eixo das **linhas**, dentro de cada célula

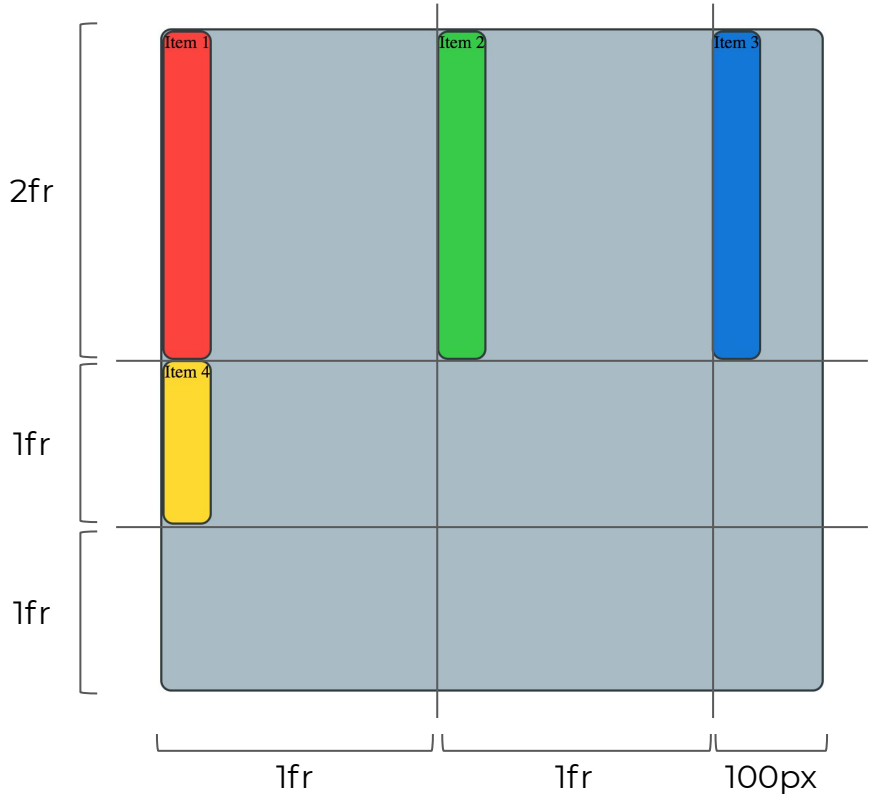
- Valores **possíveis**:
  - ***stretch***
  - *start*
  - *end*
  - *center*

justify-items: *stretch*



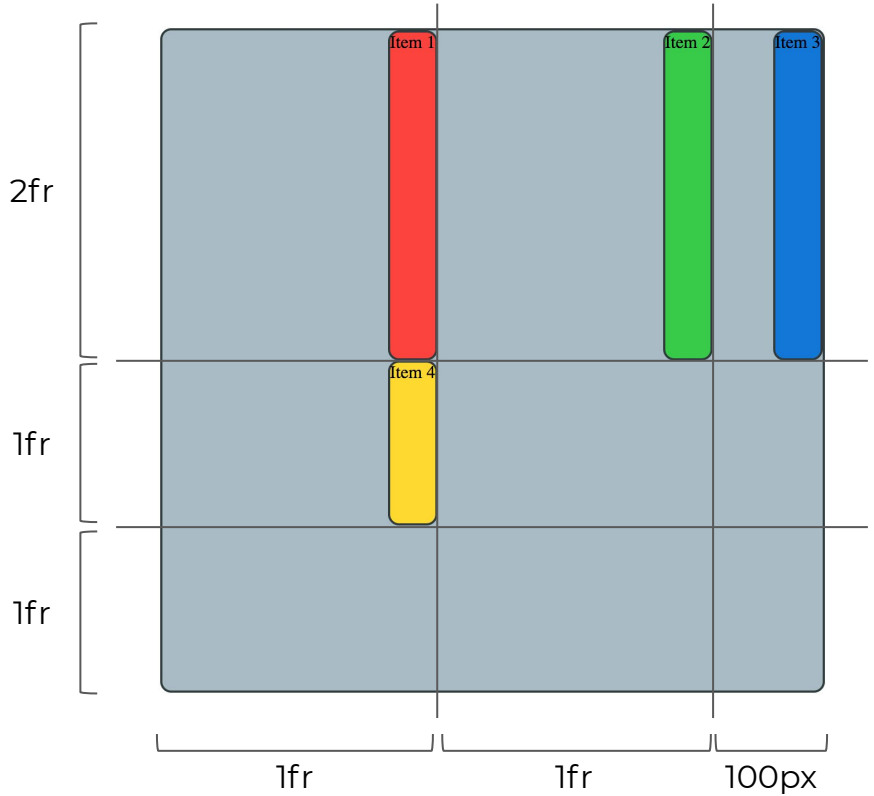
FUTURE4

justify-items: *start*



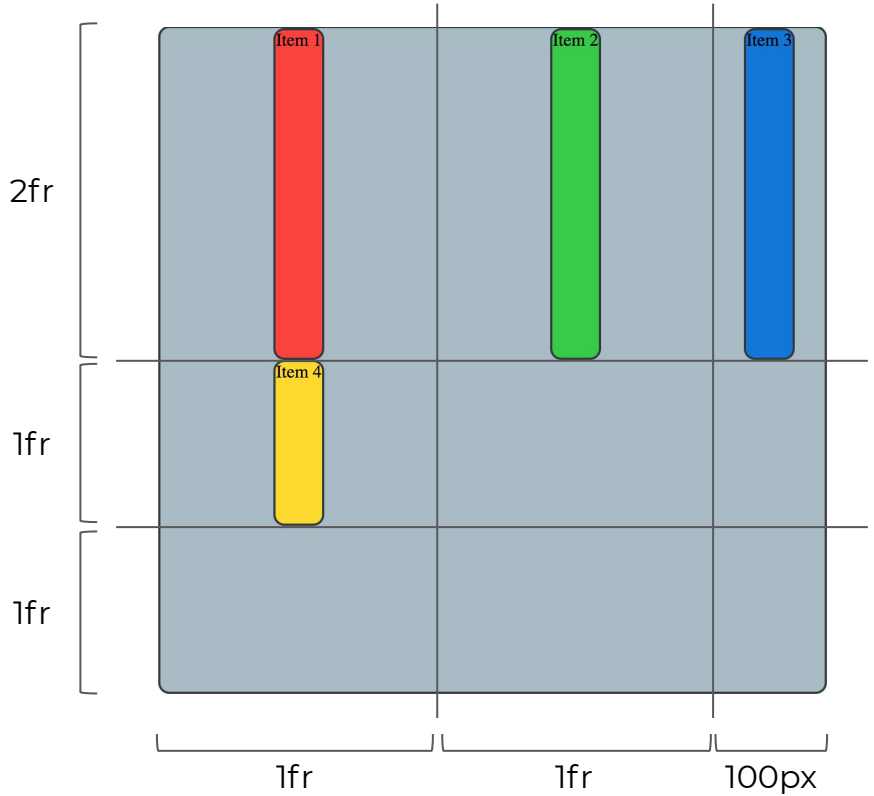
FUTURE4

justify-items: *end*



FUTURE4

justify-items: *center*



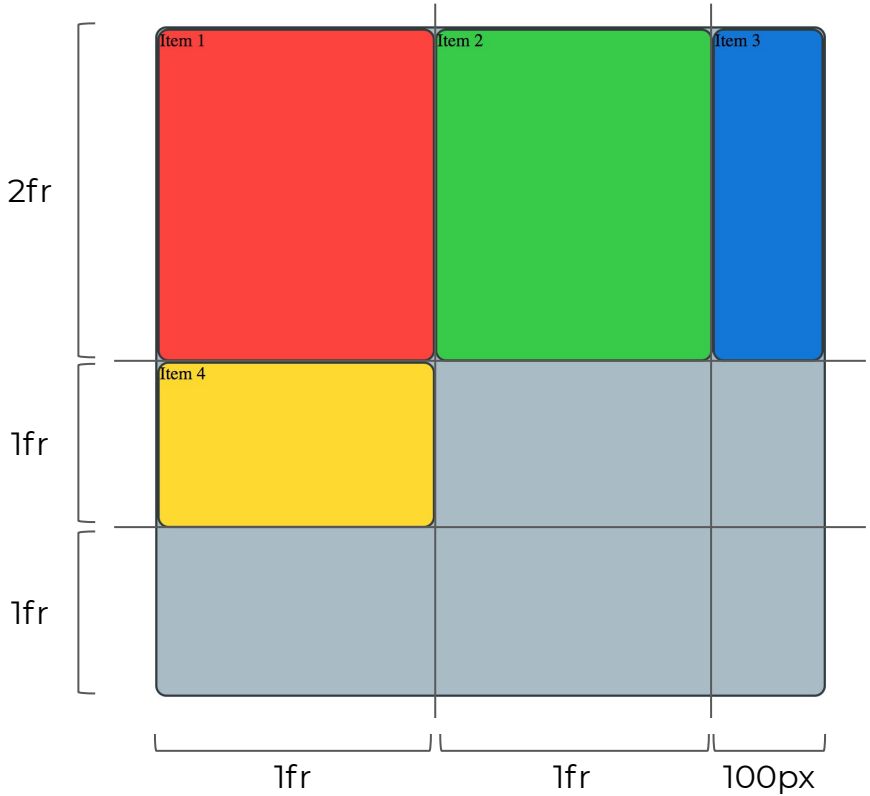
FUTURE4

align-items

- Define a **posição** dos elementos no eixo das **colunas**, dentro de cada célula

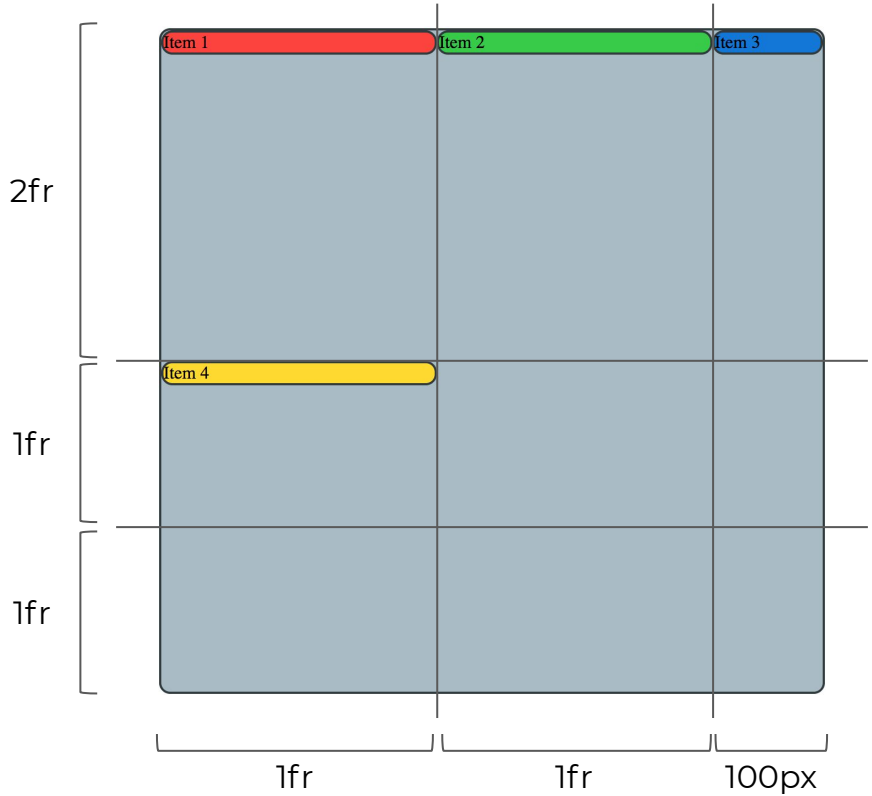
- Valores **possíveis**:
  - ***stretch***
  - *start*
  - *end*
  - *center*

align-items: *stretch*



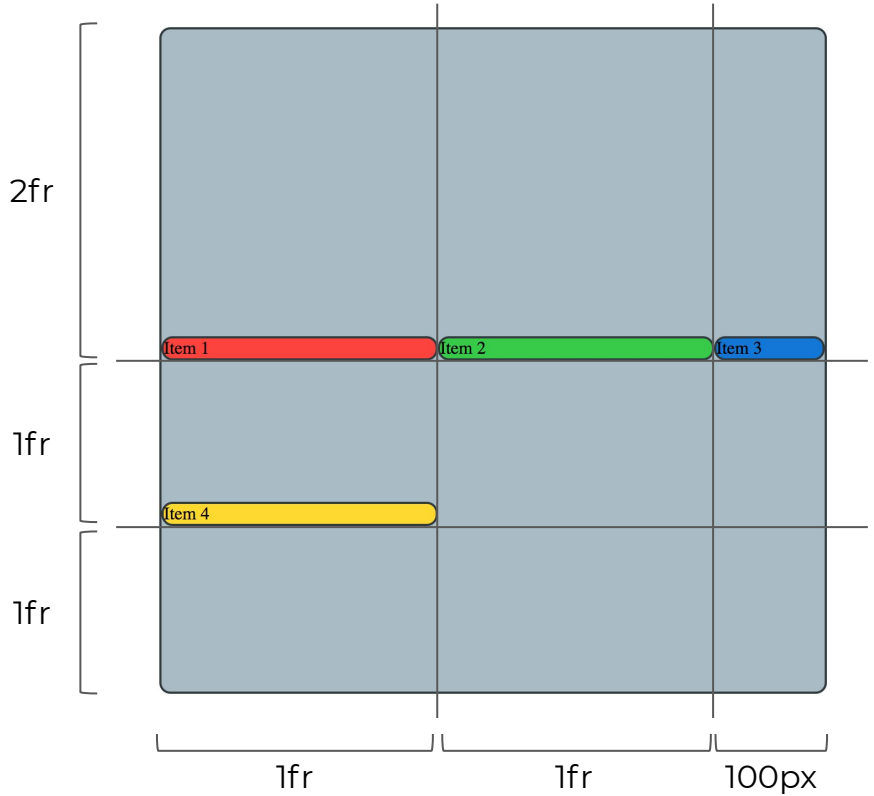
FUTURE4

align-items: *start*



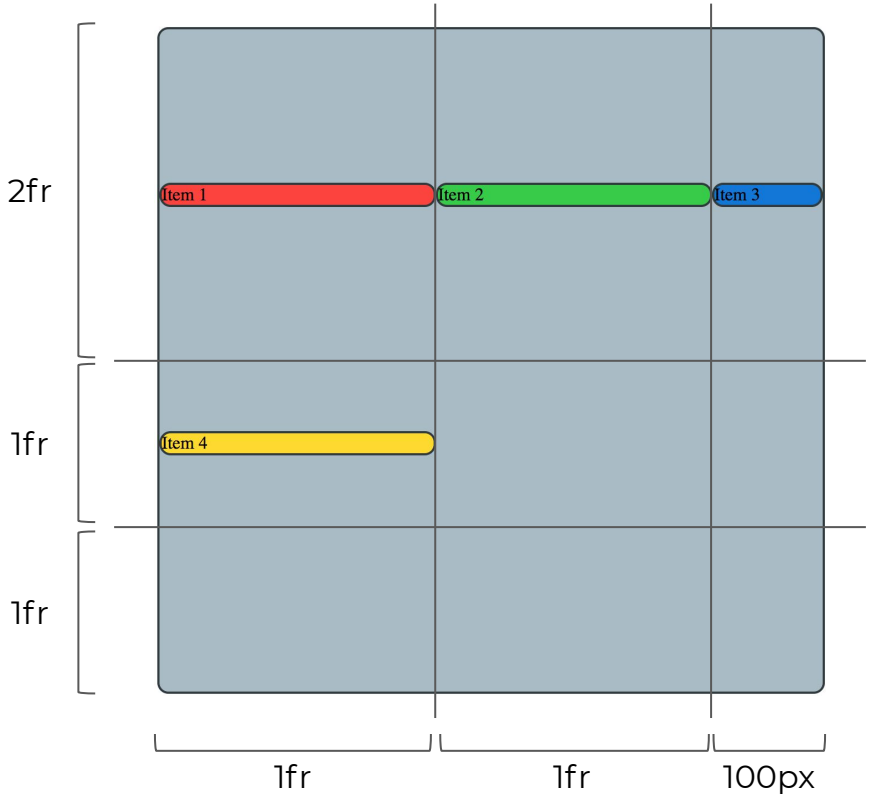
FUTURE4

align-items: *end*



FUTURE4

align-items: *center*



FUTURE4

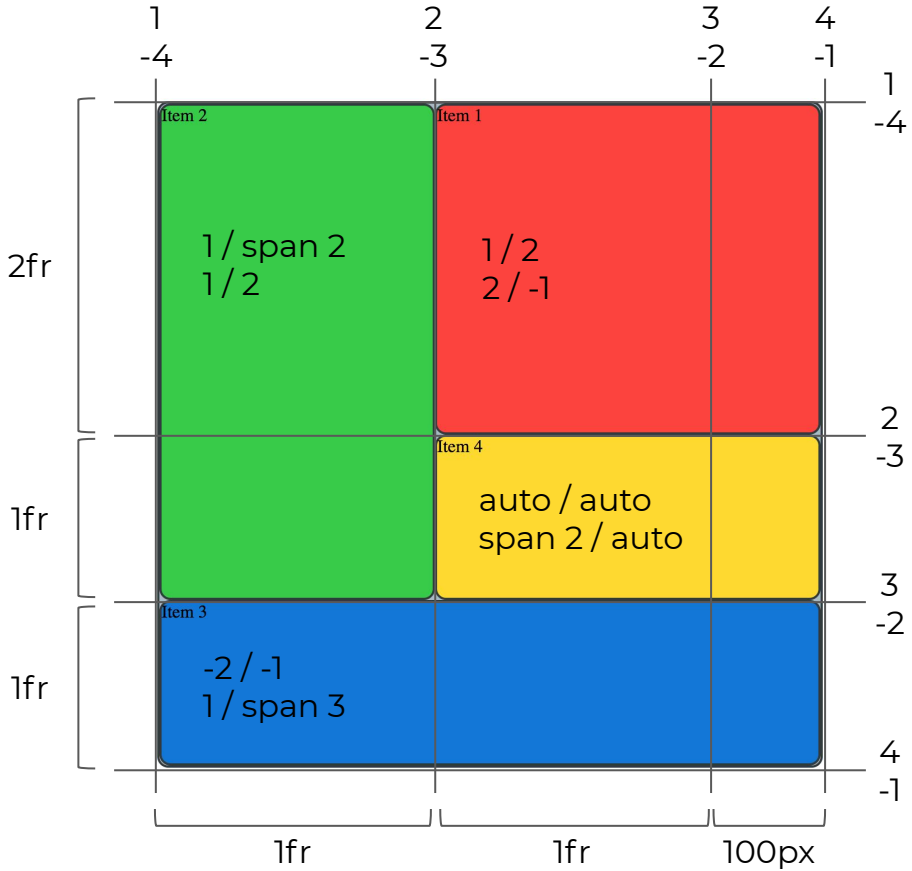
Propriedades do **item**

FUTURE4

grid-[row/column]-[start/end]

- Determina em qual linha/coluna cada item **começa** e **termina**
- Recebe o número de uma divisória
- Pode receber o valor **span**, que define um número de células a serem ocupadas

grid-[row/column]-[start/end]



Legenda:
row-start / row-end
col-start / col-end

FUTURE4

[justify/align]-self

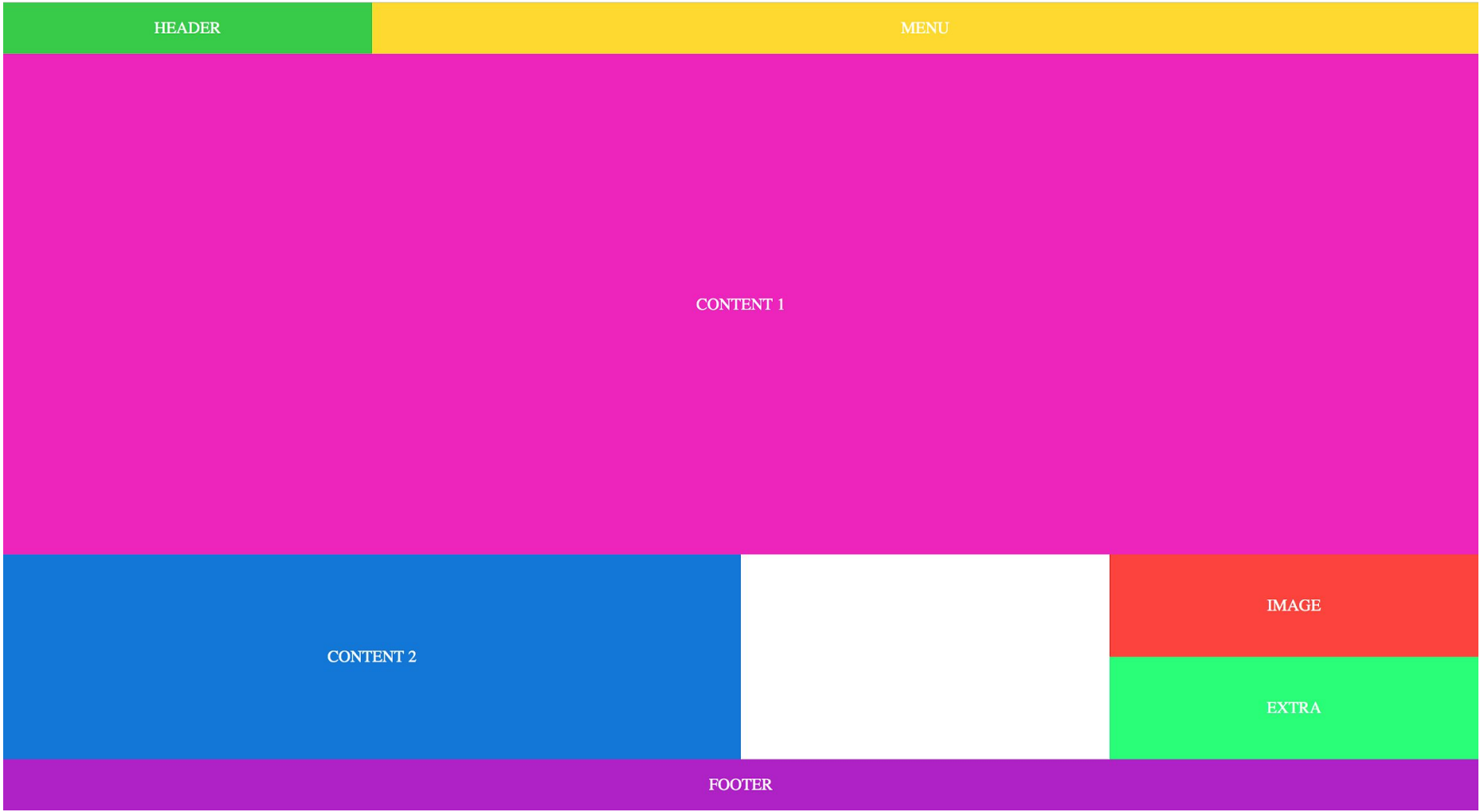
- Comportamento **igual** às propriedades **justify-items** e **align-items** do container

- Diz respeito somente ao item e **sobrepõe** as do container

- Valor **padrão** é *auto*, que segue as propriedades do container

Pausa

Coding together


HEADER
MENU
CONTENT 1
CONTENT 2
IMAGE
EXTRA
FOOTER

Grid vs. Flexbox

- Ferramentas possuem qualidades **distintas**

- Devemos usar **flexbox** quando:
  - Layout unidimensional (uma direção)
  - Elementos flexíveis (tamanhos e posições adaptáveis)
- Devemos usar **grid** quando:
  - Layouts bidimensionais (duas direções)
  - Elementos de tamanho e posição determinados

Dúvidas?

FUTURE4

Obrigado!

FUTURE4