

LAYOUT E FLEXBOX: INTRODUÇÃO AO POSICIONAMENTO DE ELEMENTOS NA PAGINA

(AULA 23 - PARTE2)

CURSO BÁSICO DE PROGRAMAÇÃO COM JAVASCRIPT

MAYARA MARQUES

mmrosatab@gmail.com

SUMÁRIO

- Display Flex
 - Display: flex e flex-direction
 - Alinhamento (justify-content, align-items)
- Mão na massa

FLEXBOX

O Flexbox foi criado com a motivação de tornar a construção de layouts web mais fácil e rápida. Com ele, o alinhamento, a distribuição de espaço e redimensionamento dos itens de dentro um container se torna mais simples.

A ideia principal do Flexbox é que elementos filhos de um container possam se ajustar automaticamente ao espaço disponível por meio de definições de propriedades.

FLEXBOX

O Flexbox trabalha com a relação entre dois papéis principais:

Container flexível

- Define um contexto flexível.

Itens flexíveis (filhos do container)

- São posicionados e dimensionados automaticamente dentro dele.

FLEXBOX

Em vez de trabalhar com posições fixas, o Flexbox:

- Cria um eixo principal (main axis) e um eixo cruzado (cross axis).
- Distribui o espaço entre os itens automaticamente.
- Alinha os elementos de acordo com propriedades.

FLEXBOX

Display

Para definir que um container é flexível, devemos usar a valor flex na propriedade display.

```
.flex-container {  
  display: flex;  
}
```

FLEXBOX

Todos elementos que forem filhos desse container se tornarão *itens flexíveis*.

```
<div class="flex-container">  
  <div class="item">1</div>  
  <div class="item">2</div>  
  <div class="item">3</div>  
</div>
```

```
.flex-container {  
  display: flex;  
}  
  
.item {  
  width: 100px;  
  height: 100px;  
  background-color: yellow;  
  border: 2px solid black;  
  margin: 1px;  
}
```

FLEXBOX

JS

The screenshot shows a web browser with three yellow boxes in a row. The browser's address bar shows the URL `127.0.0.1:5500`. The developer tools are open, showing the **Elements** panel with the following HTML structure:

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <div class="flex-container">
      <div class="item">1</div>
      <div class="item">2</div>
      <div class="item">3</div>
    </div>
    <script src="./script.js"></script>
    <!-- Code injected by live-server -->
  </body>
</html>
```

The **Styles** panel shows the following CSS rules:

```
element.style {
}

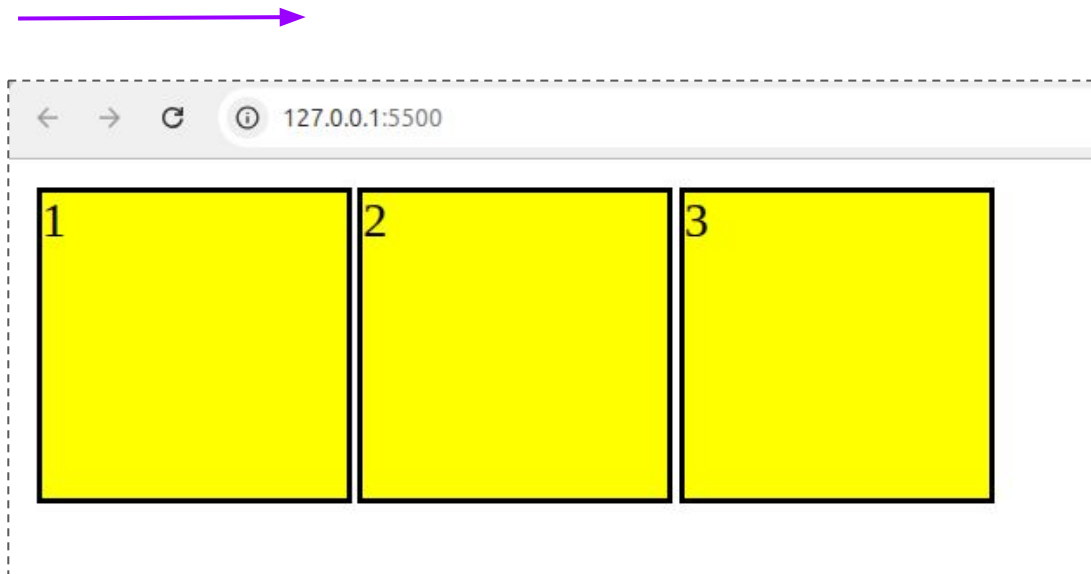
.flex-container {
  display: flex;
}

div {
  display: block;
  unicode-bidi: isolate;
}
```

A box model diagram is shown in the bottom right of the Styles panel, illustrating the layout of the selected element. The diagram shows a central content box (590x106) surrounded by padding (0), a border (0), and a margin (0).

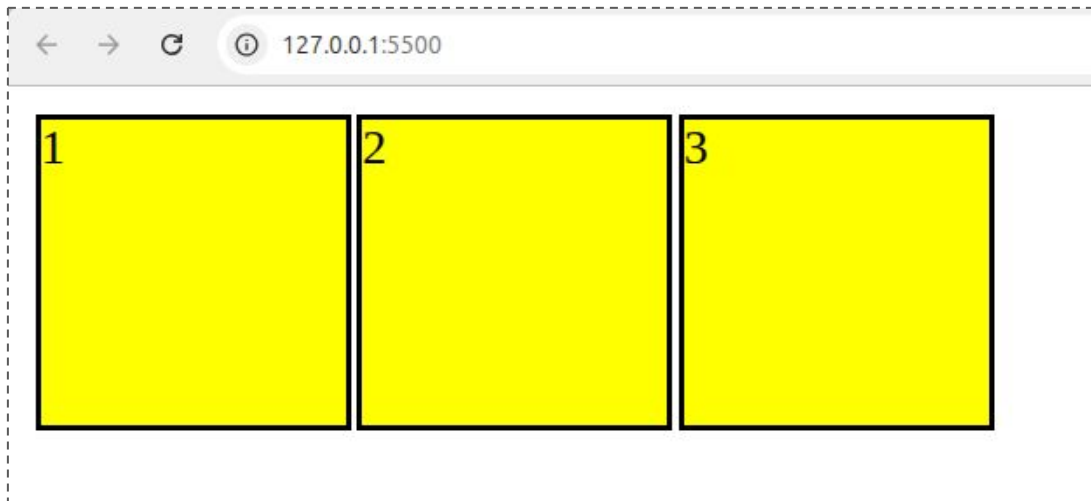
FLEXBOX

O flexbox trabalha com dois eixos, um eixo principal (main axis) e um eixo cruzado (cross axis). Quando definimos a propriedade `display: flex` sem mencionar nada a respeito de quem será eixo principal, o eixo principal, será o **horizontal ou linha**. Em outras palavras, isso significa que os itens filhos do container serão alinhados em **horizontal**.



FLEXBOX

Isso será o mesmo que definir flex-direction: row.



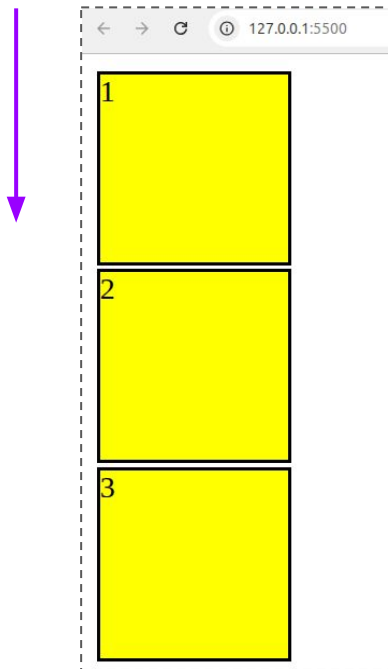
```
.flex-container {  
  display: flex;  
  flex-direction: row;  
}
```

equivale à

```
.flex-container {  
  display: flex;  
  flex-direction: row;  
}
```

FLEXBOX

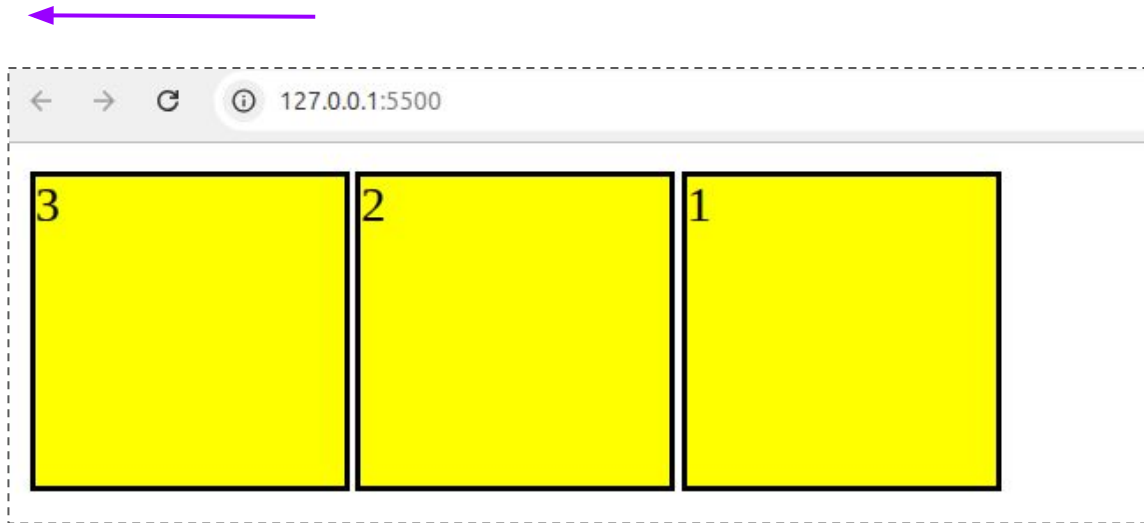
Para alterar o eixo principal para vertical, deve-se definir `flex-direction: column`.



```
.flex-container {  
  display: flex;  
  flex-direction: column;  
}
```

FLEXBOX

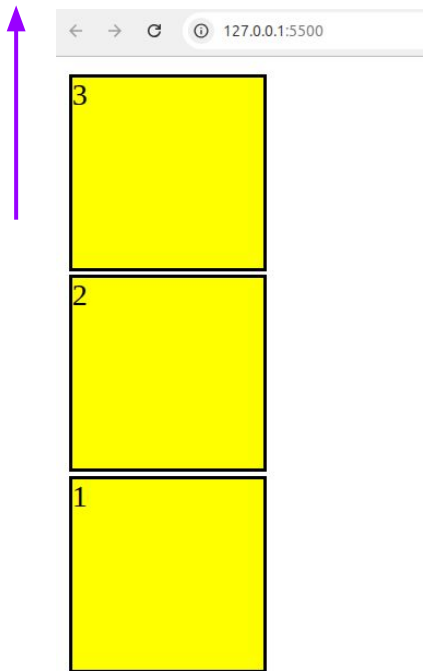
Ao definir `flex-direction: row-reverse`, os itens serão organizados em ordem contrária.



```
.flex-container {  
  position: absolute;  
  left: 8px;  
  
  display: flex;  
  flex-direction: row-reverse;  
}
```

FLEXBOX

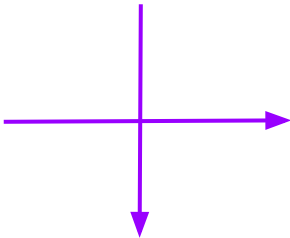
Ao definir `flex-direction: column-reverse`, os itens serão organizados em ordem contrária em relação a vertical.



```
.flex-container {  
  position: absolute;  
  left: 8px;  
  
  display: flex;  
  flex-direction: column-reverse;  
}
```

FLEXBOX

A respeito da organização e espaçamento dos itens, existem as propriedades **justify-content** e **align-items**. A **justify-content** está associada ao eixo principal que for definido e a **align-items** ao eixo cruzado.



Se o eixo principal for o horizontal, **justify-content** irá se referir a organização do itens nesse eixo e o **align-items** irá se referir ao eixo vertical, eixo contrário.

FLEXBOX

A propriedade **justify-content** é uma das ferramentas mais importantes para a distribuição e o alinhamento dos itens dentro de um contêiner, especificamente ao longo do eixo principal (Main Axis).

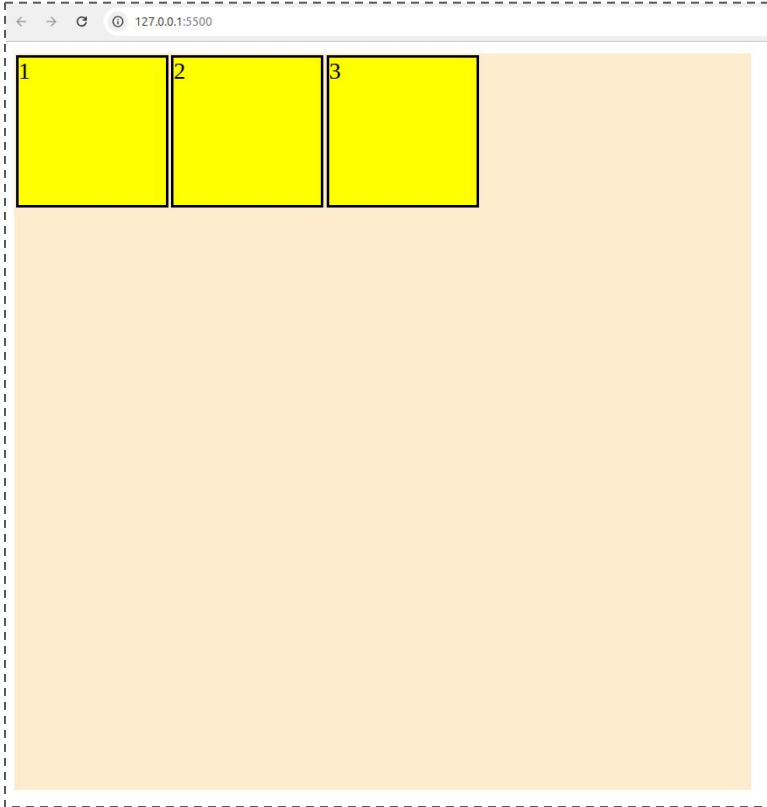
Esta propriedade possui vários valores que podem ser utilizados. Veremos o **flex-start**, **flex-end**, **center**, **space-around** e **space-between**.

FLEXBOX

flex-start: Os itens são alinhados ao início do eixo principal (por padrão, à esquerda).

```
.flex-container {  
  position: absolute;  
  top: 8px;  
  left: 8px;  
  
  width: 500px;  
  height: 500px;  
  
  background-color: blanchedalmond;  
  
  display: flex;  
  flex-direction: row;  
  justify-content: flex-start;  
}
```


FLEXBOX



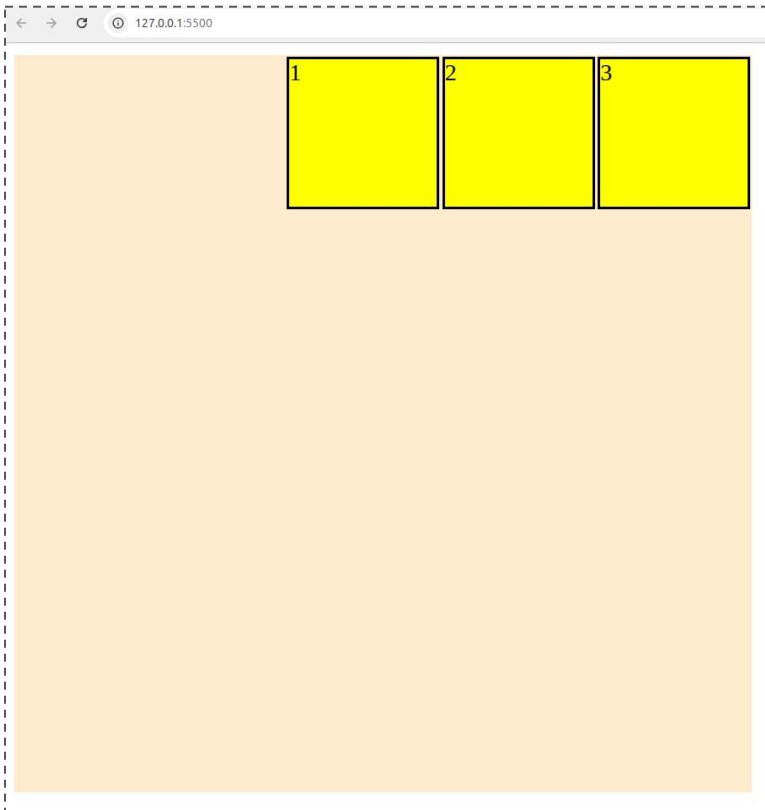
FLEXBOX

flex-end: Os itens são alinhados ao final do eixo principal (por padrão, à direita).

```
.flex-container {  
  position: absolute;  
  top: 8px;  
  left: 8px;  
  
  width: 500px;  
  height: 500px;  
  
  background-color: blanchedalmond;  
  
  display: flex;  
  flex-direction: row;  
  justify-content: flex-end;  
}
```

FLEXBOX

JS

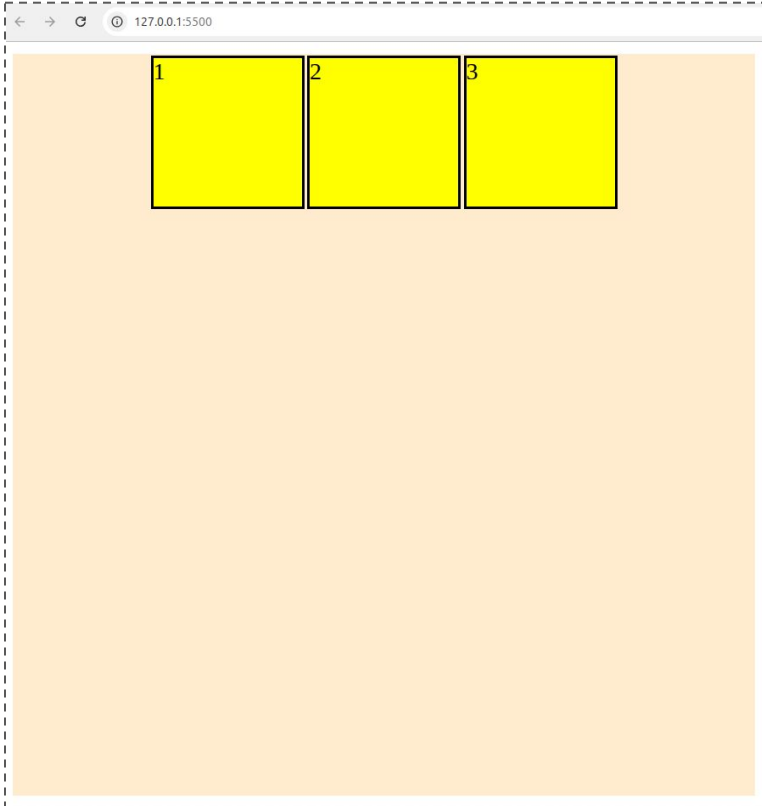


FLEXBOX

center: Os itens são centralizados ao longo do eixo principal.

```
.flex-container {  
  position: absolute;  
  top: 8px;  
  left: 8px;  
  
  width: 500px;  
  height: 500px;  
  
  background-color: blanchedalmond;  
  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
}
```

FLEXBOX

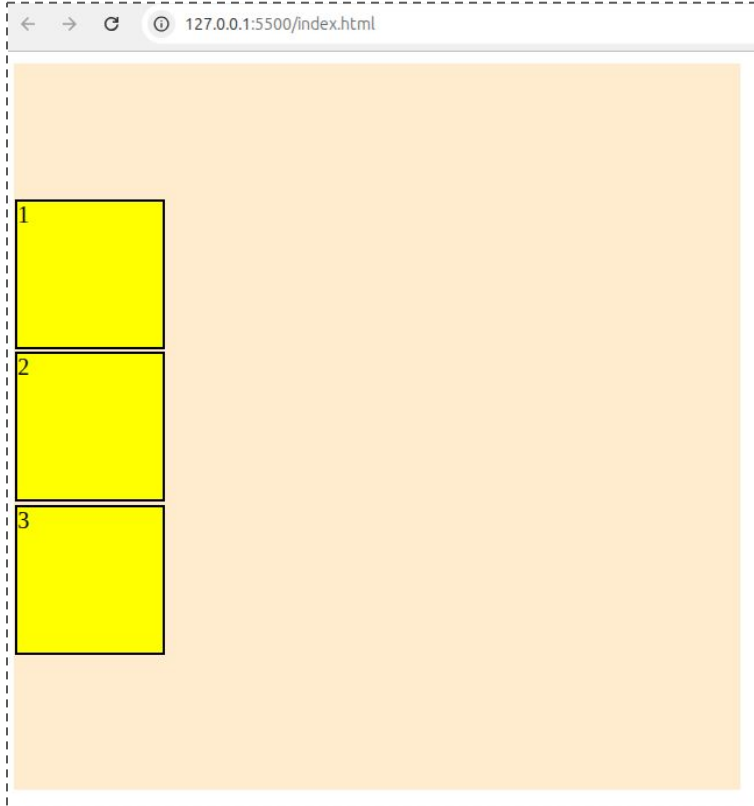


FLEXBOX

center: Trocando o eixo principal para vertical (`flex-direction: column`) e deixando os itens centralizados ao longo desse eixo.

```
.flex-container {  
  position: absolute;  
  top: 8px;  
  left: 8px;  
  
  width: 500px;  
  height: 500px;  
  
  background-color: blanchedalmond;  
  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
}
```

FLEXBOX

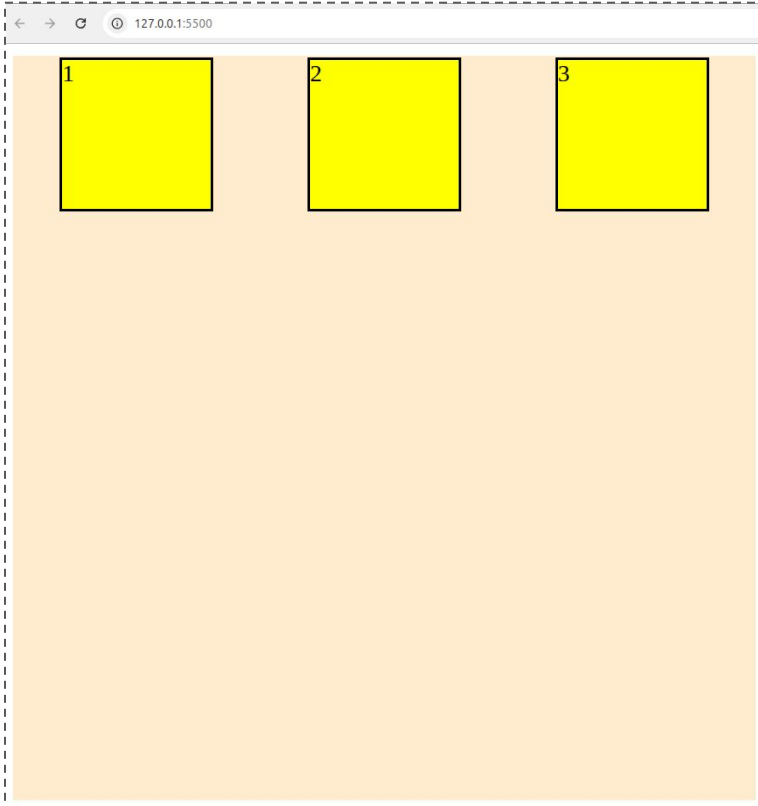


FLEXBOX

space-around: Os itens são distribuídos com espaço igual ao redor de cada um. Isso faz com que o espaço entre os itens seja duas vezes maior do que o espaço nas bordas.

```
.flex-container {  
  position: absolute;  
  top: 8px;  
  left: 8px;  
  
  width: 500px;  
  height: 500px;  
  
  background-color: blanchedalmond;  
  
  display: flex;  
  flex-direction: row;  
  justify-content: space-around;  
}
```


FLEXBOX



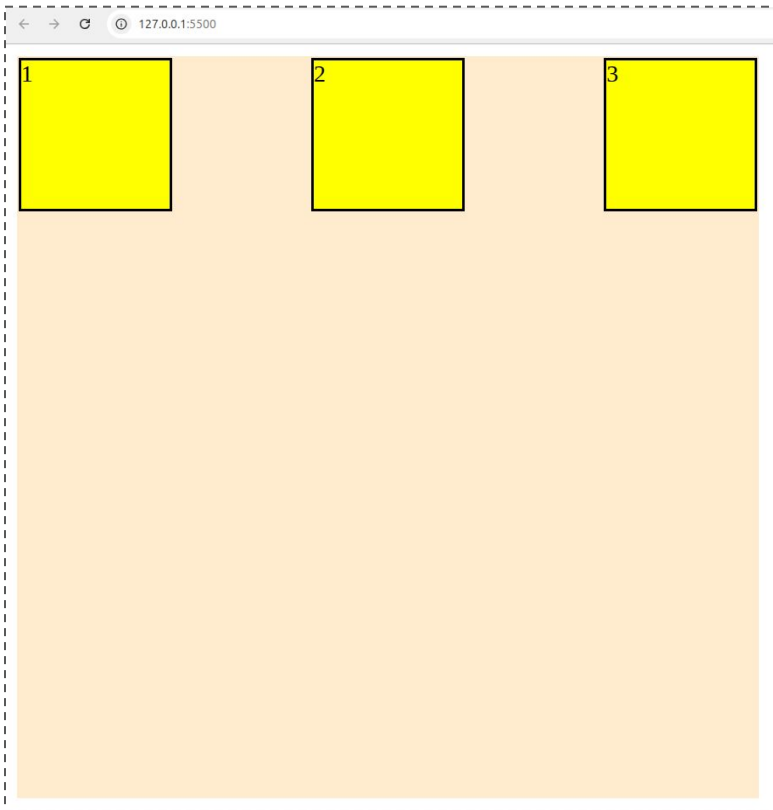
FLEXBOX

space-between: Os itens são distribuídos com espaço igual entre eles, mas sem espaço nas bordas do container.

```
.flex-container {  
  position: absolute;  
  top: 8px;  
  left: 8px;  
  
  width: 500px;  
  height: 500px;  
  
  background-color: blanchedalmond;  
  
  display: flex;  
  flex-direction: row;  
  justify-content: space-between;  
}
```

FLEXBOX

JS



FLEXBOX

A propriedade **align-items** é crucial para controlar o alinhamento dos itens dentro de um contêiner flexível ao longo do eixo cruzado (o eixo perpendicular ao eixo principal).

A propriedade **align-items** possui vários valores que podem ser utilizados. Veremos o **flex-start**, **flex-end** e **center**.

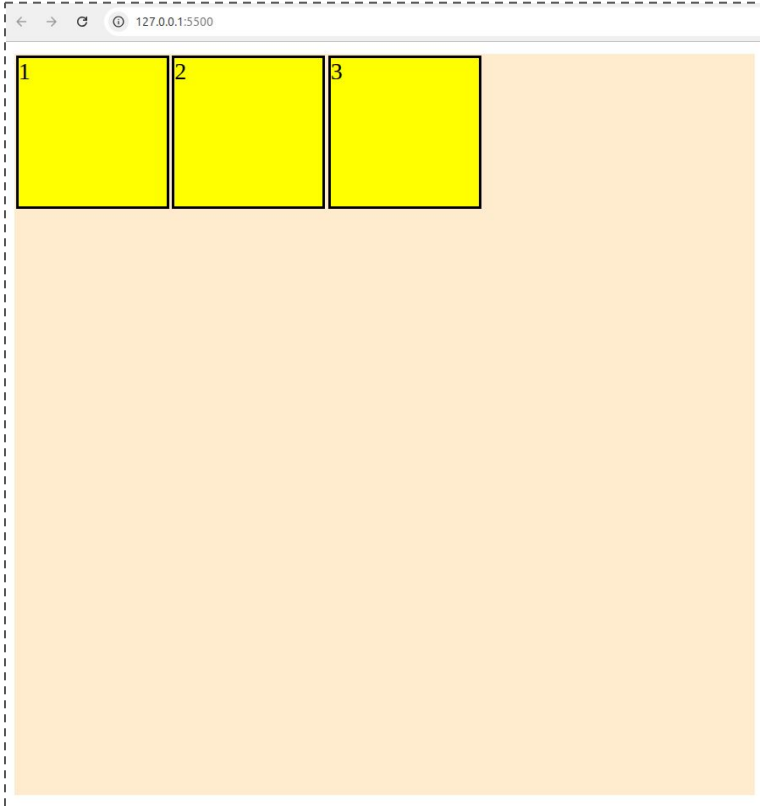
FLEXBOX

align-items

flex-start: Alinha os itens flexíveis ao início do eixo cruzado.

```
.flex-container {  
  position: absolute;  
  top: 8px;  
  left: 8px;  
  
  width: 500px;  
  height: 500px;  
  
  background-color: blanchedalmond;  
  
  display: flex;  
  flex-direction: row;  
  justify-content: flex-start;  
  align-items: flex-start;  
}
```

FLEXBOX



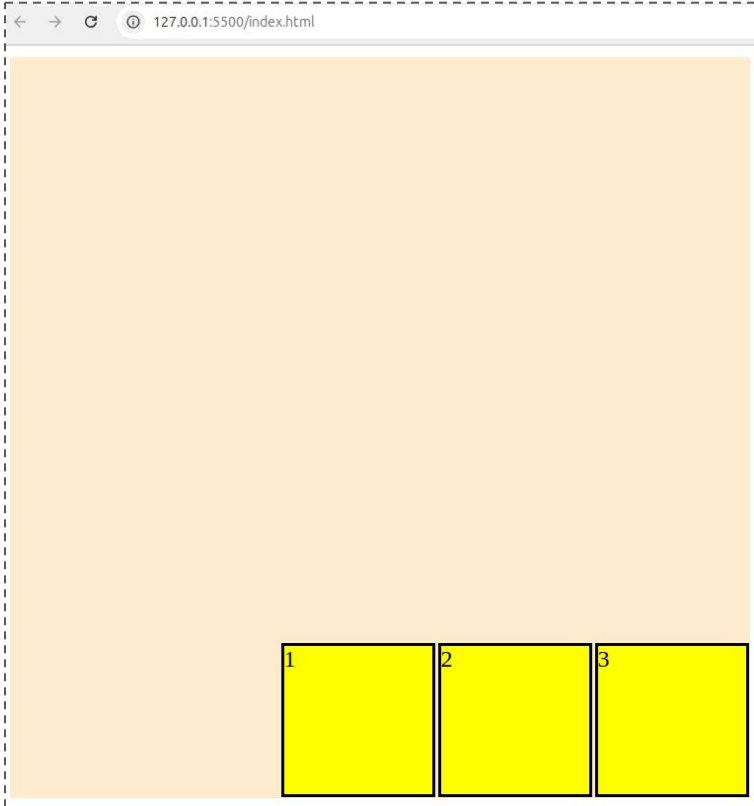
FLEXBOX

align-items

flex-end: Alinha os itens flexíveis ao fim do eixo cruzado.

```
.flex-container {  
  position: absolute;  
  top: 8px;  
  left: 8px;  
  
  width: 500px;  
  height: 500px;  
  
  background-color: blanchedalmond;  
  
  display: flex;  
  flex-direction: row;  
  justify-content: flex-end;  
  align-items: flex-end;  
}
```

FLEXBOX



FLEXBOX

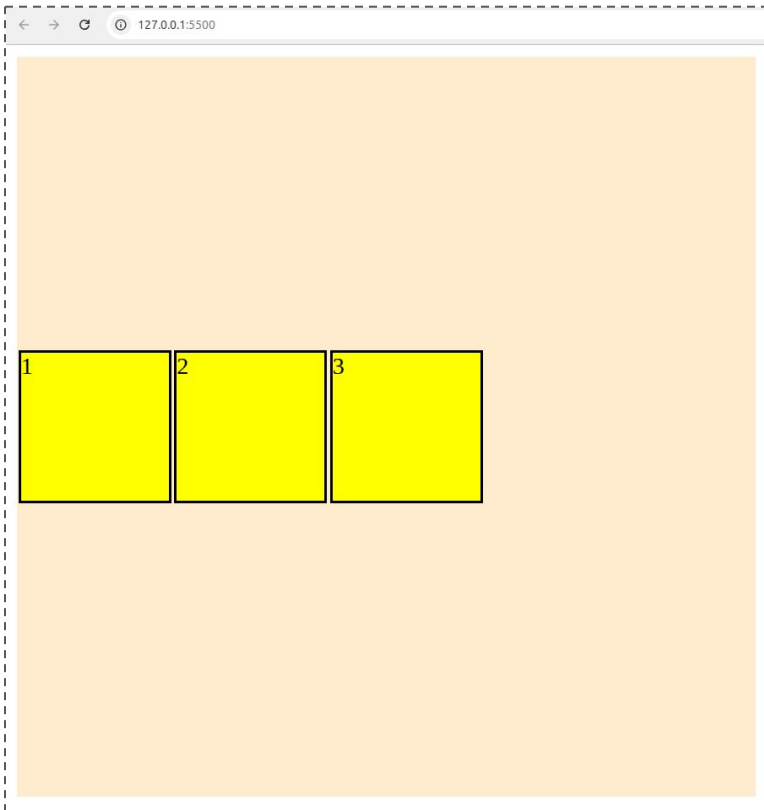
align-items

center: Centraliza os itens flexíveis ao longo do eixo cruzado.

```
.flex-container {  
  position: absolute;  
  top: 8px;  
  left: 8px;  
  
  width: 500px;  
  height: 500px;  
  
  background-color: blanchedalmond;  
  
  display: flex;  
  flex-direction: row;  
  justify-content: flex-start;  
  align-items: center;  
}
```

FLEXBOX

JS



FLEXBOX

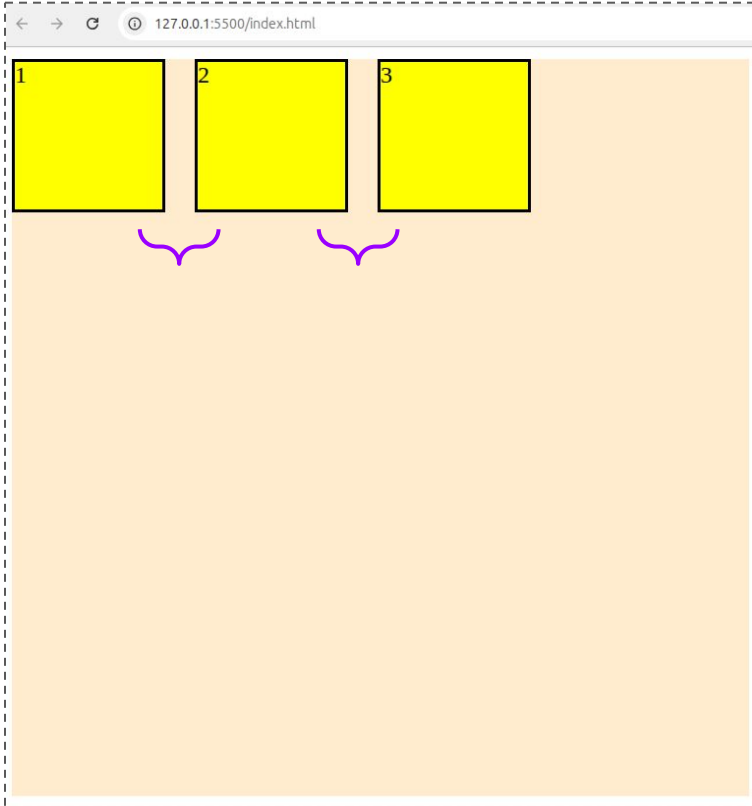
Gap

Permite definir espaçamento entre os itens flexíveis.

```
<div class="flex-container">  
  <div class="item">1</div>  
  <div class="item">2</div>  
  <div class="item">3</div>  
</div>
```

```
.flex-container {  
  display: flex;  
  gap: 20px;  
  width: 500px;  
  height: 500px;  
  
  background-color: blanchedalmond;  
}  
  
.item {  
  width: 100px;  
  height: 100px;  
  background-color: yellow;  
  border: 2px solid black;  
}
```

FLEXBOX



MÃO NA MASSA



1. Monte um layout como o abaixo.



MÃO NA MASSA



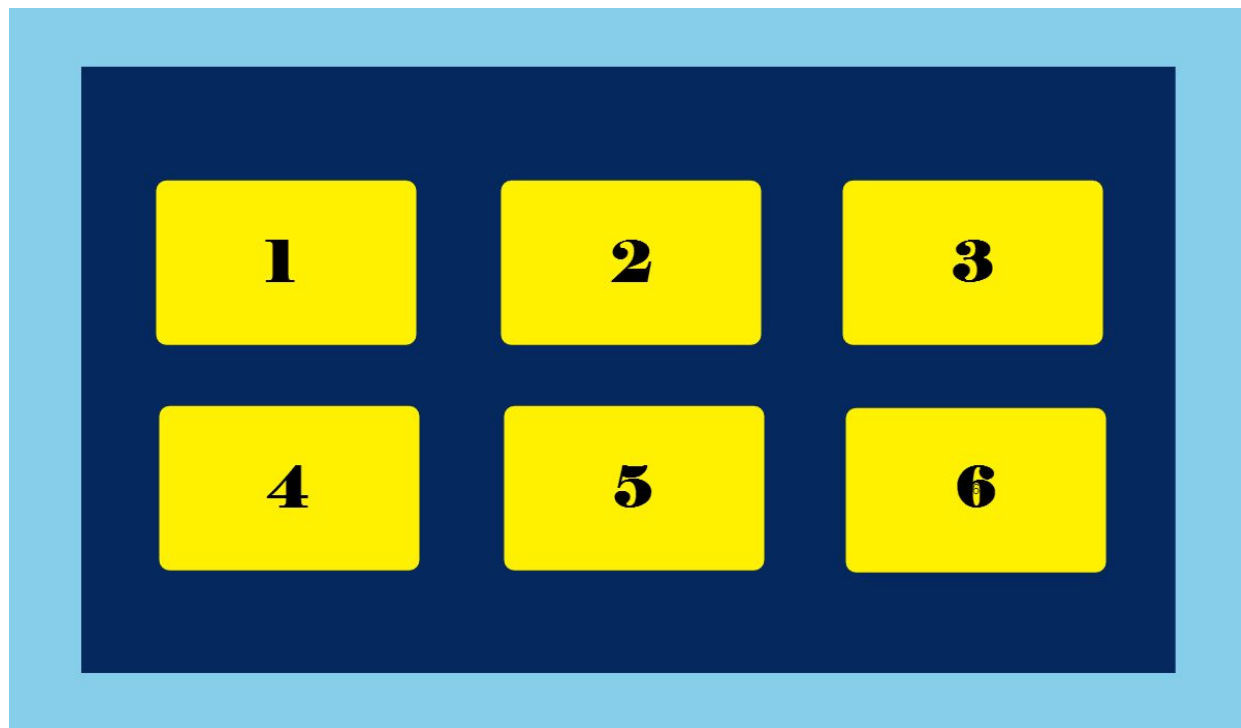
2. Monte um layout como o abaixo.



MÃO NA MASSA



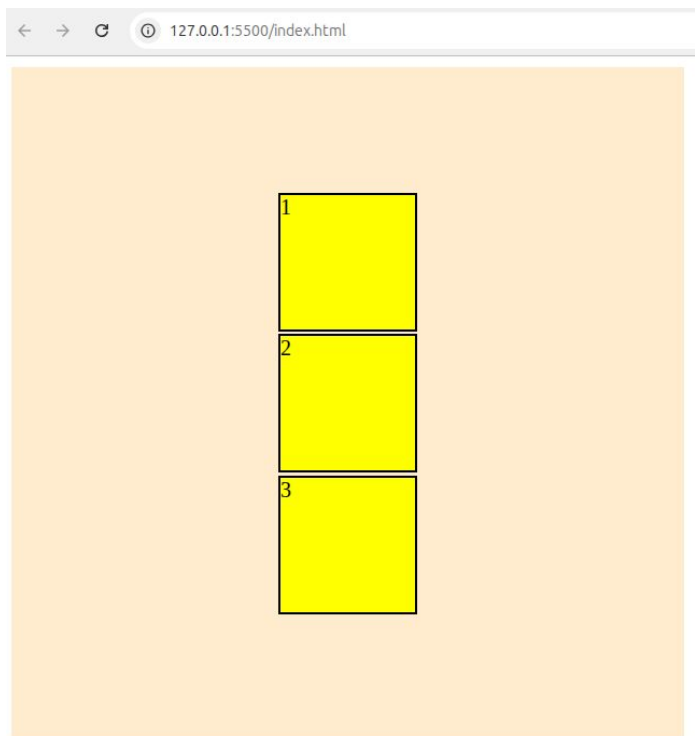
3. Monte um layout como o abaixo.



MÃO NA MASSA



4. Monte um layout como o abaixo.



REFERÊNCIAS

- CSS3 Flexbox: Funcionamento e propriedades
- Flexbox CSS: Guia Completo, Elementos e Exemplos