

INSTITUTO FEDERAL
Mato Grosso do Sul

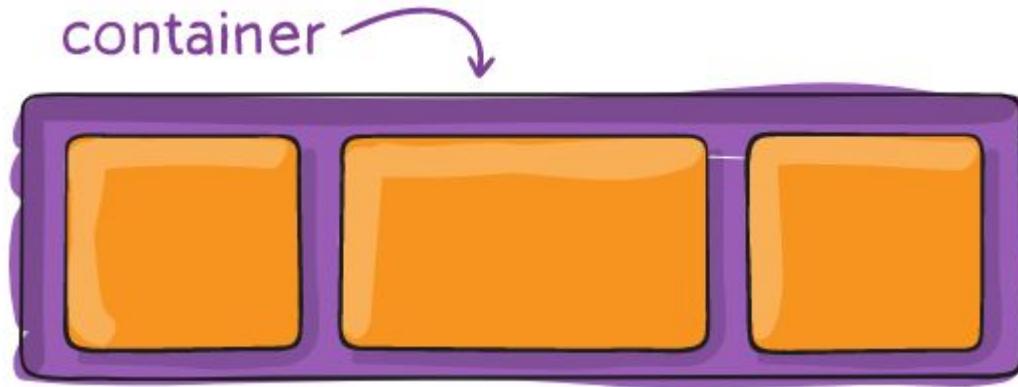
DESENVOLVIMENTO WEB 1:
CSS flexbox

O QUE É CSS FLEXBOX?

- O flexbox (Flexible Box) nos permite organizar, alinhar e distribuir itens dentro de um container. Com ele fica mais simples definir o tamanho e o alinhamento vertical e horizontal de itens.
- Flexbox lida com o layout numa dimensão de cada vez - quer seja como uma linha ou como uma coluna.

COMO FUNCIONA CSS FLEXBOX?

- O flexbox é um módulo inteiro e não uma propriedade única, envolve muitas coisas, incluindo todo o seu conjunto de propriedades.
- Existem propriedades para trabalhar com o elemento que possui itens (container ou elemento pai) e propriedades para os itens (elementos filhos).
- Algumas propriedades devem ser definidas no container (elemento pai, conhecido como "flex container"), enquanto outras devem ser definidas nos filhos (chamados "flex items").



**Propriedades definidas no elemento pai
(flex container)**

CONTAINER

- O primeiro passo para começar a usar Flexbox é criar um flex container. elementos-filhos de um flex container são chamados flex items e são estilizados dentro do flex container usando as propriedades Flexbox.
- Um flex container é criado ao usar a propriedade display de um elemento com valor **flex** ou **inline-flex**.



CONTAINER

pagina.html

```
<div class="container">
    <div class="item"> 1 </div>
    <div class="item"> 2 </div>
    <div class="item"> 3 </div>
    <div class="item"> 4 </div>
</div>
```

pagina.css

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

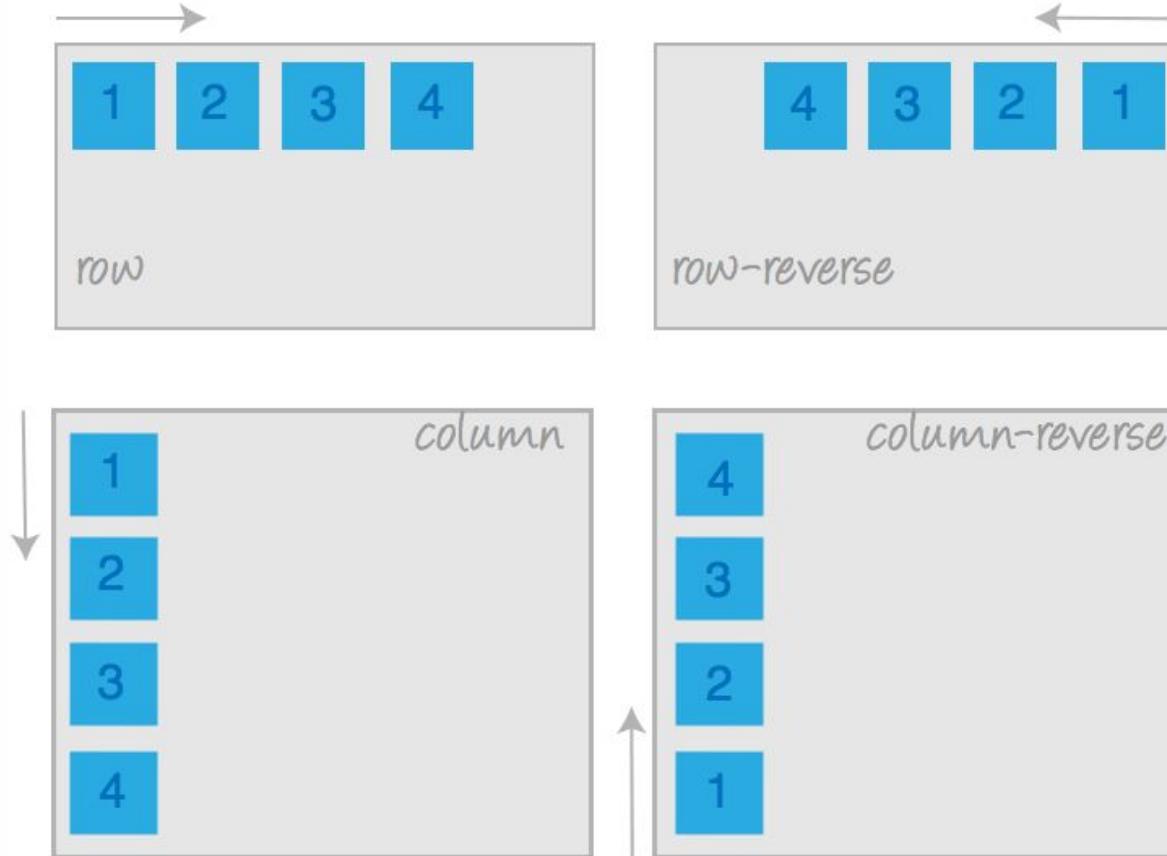
PROPRIEDADES DE FLEX CONTAINER

- **flex-direction:** row | row-reverse | column | column-reverse
 - indica a direção dos itens, definindo o que vamos chamar de eixo principal (main-axis).
- **flex-wrap:** nowrap | wrap | wrap-reverse
 - controla se o flex container é single-line ou mult-line e a direção do eixo transversal (cross-axis), que determina a direção em que novas linhas são empilhadas (stacked).
- **flex-flow:** <'flex-direction'> <'flex-wrap'>

FLEX-DIRECTION

flex-direction: row | row-reverse | column | column-reverse

- A propriedade flex-direction especifica **como os flex items são dispostos em um flex container**, definindo o que vamos chamar de **eixo principal** (main-axis).
- Valores possíveis:
 - **row** (padrão): da esquerda para direita (segue o padrão de leitura)
 - **row-reverse**: inverso de row
 - **column**: de cima para baixo
 - **column-reverse**: inverso de column

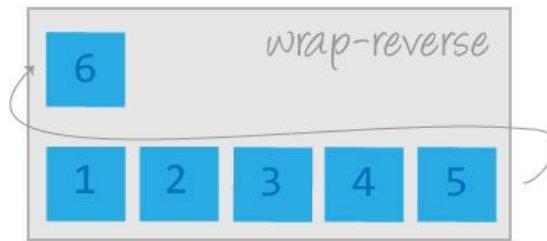
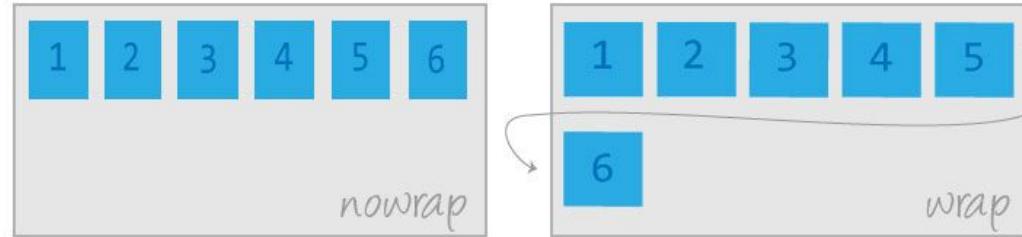


Flex-Direction

FLEX-WRAP

`flex-wrap: nowrap | wrap | wrap-reverse`

- Esta propriedade permite **controlar o comportamento dos itens, quando eles ultrapassam o espaço na linha do container.**
- **nowrap** (padrão): todos os itens ficam em uma única linha, mesmo que isso “estoure” (overflow) tamanho do container
- **wrap**: os itens flexíveis serão quebrados em várias linhas, de cima para baixo.
- **wrap-reverse**: os itens flexíveis serão agrupados em várias linhas de baixo para cima.



Flex-Wrap

FLEX-FLOW

`flex-flow: <flex-direction> <flex-wrap>`

- A propriedade flex-flow é **um propriedade de abreviação** (shorthand) de usar as propriedades flex-direction e flex-wrap que, juntas, definem os Eixos de um flex container.
- O valor inicial de flex-flow é a concatenação dos valores iniciais de flex-direction e flex-wrap:



English

```
flex-flow: row wrap;  
writing-mode: horizontal-tb;
```



Japanese

```
flex-flow: row wrap;  
writing-mode: vertical-rl;
```



PROPRIEDADES ALINHAMENTO DOS ITEMS

- **justify-content:** flex-start | flex-end | center | space-between | space-around
 - define o alinhamento dos itens ao longo do eixo principal
- **align-items:** flex-start | flex-end | center | baseline | stretch
 - define o alinhamento dos itens perpendicularmente em relação ao eixo principal. Pense nele como um justify-content, mas que alinhará os itens no outro eixo.
- **align-content:** flex-start | flex-end | center | space-between | space-around | stretch
 - alinha as linhas do container. Por alinhar as linhas, esta propriedade só tem efeito quando há mais de uma linha.

JUSTIFY-CONTENT

`justify-content: flex-start | flex-end | center | space-between | space-around`

Alinha os itens flex no container de acordo com a direção. A propriedade só funciona se os itens atuais não ocuparem todo o container.

- **flex-start** (padrão): os itens ficam junto no começo da linha
- **flex-end**: os itens ficam juntos no final da linha
- **center**: os itens ficam centralizados na linha
- **space-between**: os itens são distribuídos igualmente no espaço disponível. O primeiro item fica no começo da linha e o último fica no final.
- **space-around**: os itens são distribuídos igualmente no espaço disponível ao redor deles.



flex-start



flex-end



center



space-between



space-around



Justify-Content

ALIGN-ITEMS

align-items: flex-start | flex-end | center | baseline | stretch

Define o alinhamento dos itens perpendicularmente em relação ao eixo principal. O mesmo que justify-content, mas que alinhará os itens no outro eixo.

- **stretch** (padrão): estica os elementos para preencherem o container.
- **flex-start**: os itens ficam junto no começo do eixo perpendicular
- **flex-end**: os itens ficam juntos no final do eixo perpendicular
- **center**: os itens ficam centralizados no eixo perpendicular
- **baseline**: parecido com o center, mas usando a base da linha como referência.



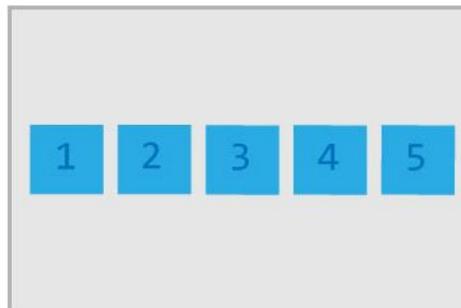
flex-start



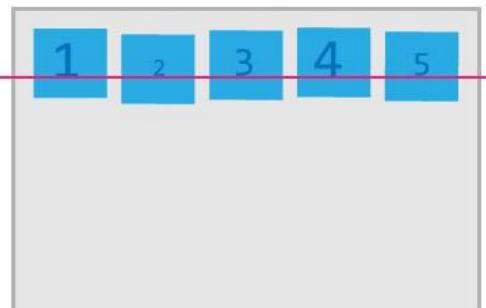
flex-end



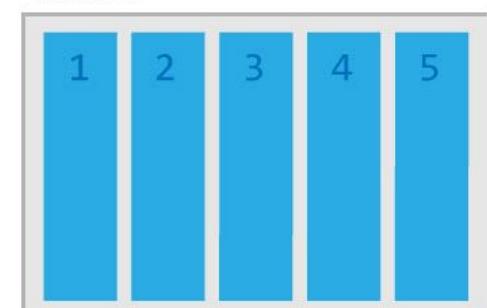
center



baseline



stretch



Align-Items

ALIGN-CONTENT

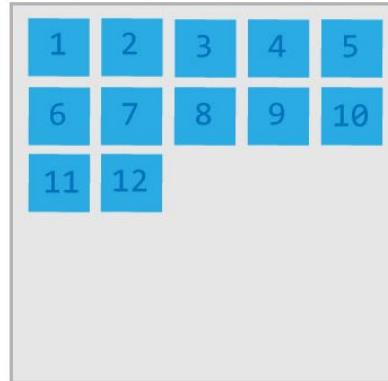
align-content: flex-start | flex-end | center | space-between | space-around | stretch

Alinha as linhas do container. Por alinhar as linhas, esta propriedade só tem efeito quando há mais de uma linha.

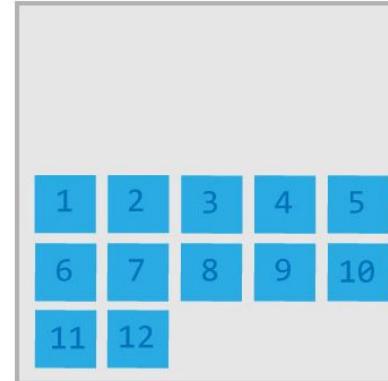
- **stretch** (padrão): estica as linhas para preencherem o espaço restante.
- **flex-start**: as linhas ficam juntas no começo do container
- **flex-end**: as linhas ficam juntas no final do container
- **center**: as linhas ficam centralizadas no container
- **space-between**: as linhas são distribuídas igualmente. A primeira linha fica no começo do container e a última fica no final.
- **space-around**: as linhas são distribuídas igualmente no espaço disponível ao redor delas.



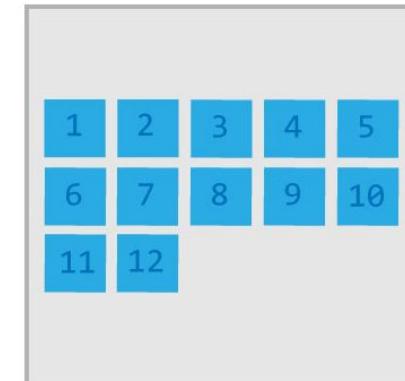
flex-start



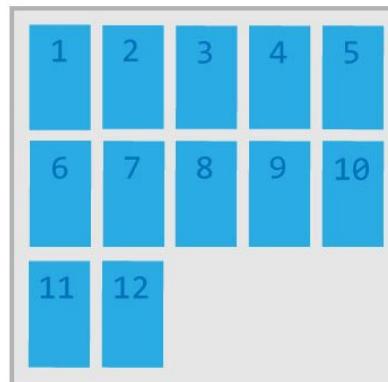
flex-end



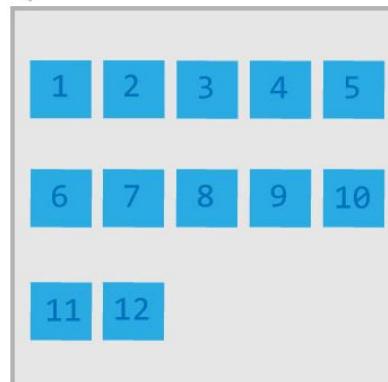
center



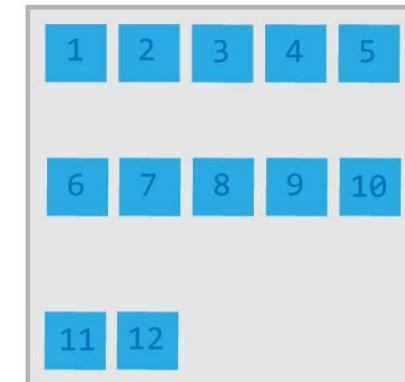
stretch



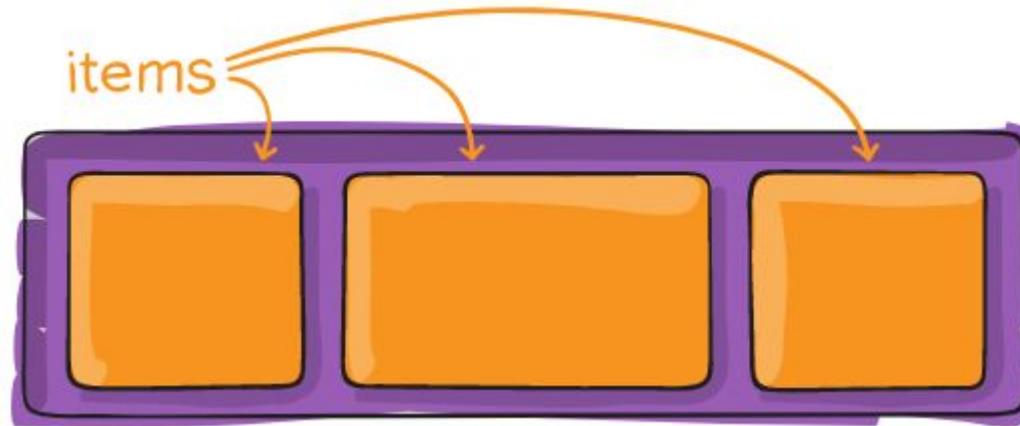
space-around



space-between



Align-Content



**Propriedades definidas nos elementos filhos
(flex items)**



ITEM

PAGINA.HTML

```
<div class="container">
    <div class="item"> 1 </div>
    <div class="item selected"> 2 </div>
    <div class="item"> 3 </div>
    <div class="item"> 4 </div>
</div>
```

ESTILO.CSS

```
.container{
    display: flex;
    justify-content: space-evenly;
    align-items: center;
}
.item.selected{
    order: 3;
}
```

PROPRIEDADES DE FLEX ITEM

- **order: <integer>**
 - controla a ordem em que flex items aparecem.
- **align-self: auto | flex-start | flex-end | center | baseline | stretch**
 - permite sobrescrever esse alinhamento padrão para items individuais.
- **flex-basis: <number>**
 - define o tamanho padrão de um elemento antes do espaço disponível ser distribuído
- **flex-grow: <number>**
 - especifica o fator de crescimento de um flex item
- **flex-shrink: <number>**
 - especifica o fator de encolhimento de um flex item

ORDER

order: <integer>

- Flex items são, por padrão, mostrados na mesma ordem em que aparecem no código-fonte do documento. **A propriedade order pode ser usada para alterar essa ordem.**
- A propriedade order **controla a ordem em que flex items aparecem num flex container** ao colocá-los em grupos ordinais, atribuindo-se um valor <integer> que define a qual grupo ordinal o flex item pertence — items com o mesmo grupo ordinal são mostrados na ordem em que aparece na camada de conteúdo.



(default)



order: 1



order: 3



Order

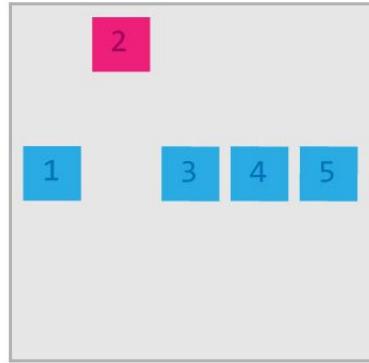
ALIGN-SELF

align-self: auto | flex-start | flex-end | center | baseline | stretch

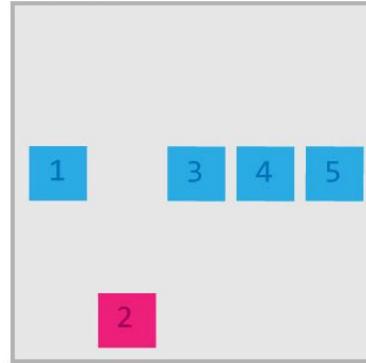
- align-items especifica o alinhamento padrão de todos os flex items do container; a propriedade align-self permite sobrescrever esse alinhamento padrão para items individuais.



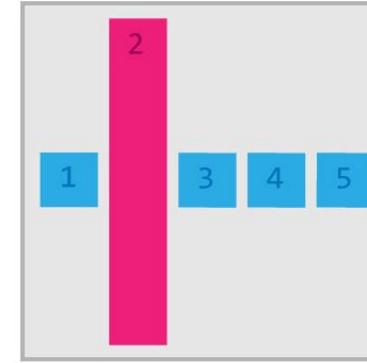
flex-start



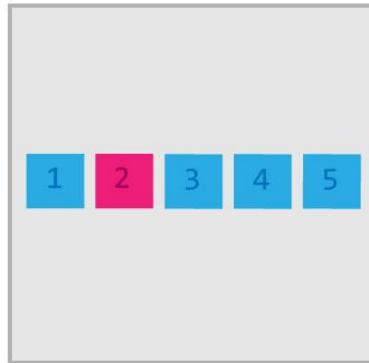
flex-end



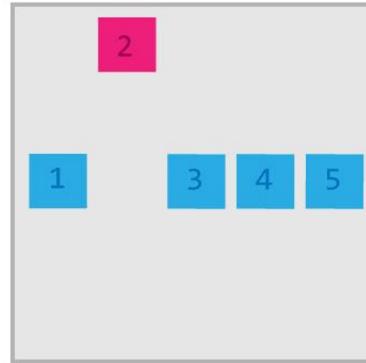
stretch



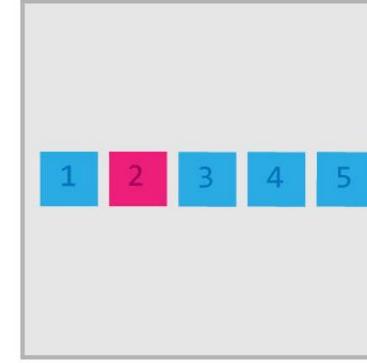
center



baseline



auto



Align-Self

with `.container { align-items: center; }`

FLEX-BASIS

`flex-basis: auto | <'width'>`

- Define o **tamanho padrão de um elemento antes do espaço disponível ser distribuído**. O valor pode ser em px, %, em, etc.
- Se o valor for auto, ele irá olhar para o valor de width e height do item.
- Se o valor for content, ele irá olhar para o tamanho do conteúdo do item.

FLEX-GROW

flex-grow: <number>

- A propriedade flex-grow **especifica o fator de crescimento de um flex item**. O fator de crescimento é um <number> que determina o quanto o flex item vai crescer em relação ao restante dos flex items no flex container quando espaço em branco positivo é distribuído.
 - O valor inicial de flex-grow é 0 e números negativos são inválidos.



(default)



flex-grow: 1 (all)



flex-grow: 2 (on second item)



Flex-Grow

FLEX-SHRINK

flex-shrink: <number>

- A propriedade flex-shrink **define a capacidade de um item flexível encolher, se necessário.** O fator é um <number> que determina o quanto o flex item vai encolher em relação ao restante dos flex items no flex container quando espaço em branco é distribuído.
- O fator de encolhimento é multiplicado pela propriedade flex-basis quando distribui espaço negativo.
 - O valor inicial de flex-grow é 0 (significando que items não têm fator de encolhimento por padrão) e números negativos são inválidos.

FLEX

`flex: <flex-grow> [flex-shrink] [flex-basis]`

- A propriedade flex é um propriedade de abreviação (shorthand) de usar as propriedades flex-grow, flex-shrink e flex-basis, nos permitindo declarar os valores em uma única propriedade.
- O segundo e terceiro parâmetros (flex-shrink e flex-basis) são opcionais.
 - O padrão é `flex: 0 1 auto;`

JOGOS CSS FLEXBOX/GRID

- <https://flexboxfroggy.com/>
- <http://www.flexboxdefense.com/>
- <https://cssgridgarden.com/>

GERADORES CSS FLEXBOX

- <https://flexbox.help/>
- <https://bennettfeely.com/flexplorer/>
- <https://the-echoplex.net/flexyboxes/>
- <https://css3gen.com/free-html5-responsive-web-template/>

CSS TOOLS

- <https://html-css-js.com/css/generator/>
- <https://webcode.tools/css-generator>
- <https://www.cssportal.com/css3-menu-generator/#>
- <https://www.cssmatic.com>
- [smashingmagazine - 50-ferramentas de css uteis](#)
- <https://dribbble.com/>
- <https://www.typewolf.com/>
- <https://www.awwwards.com/>

MATERIAIS COMPLEMENTARES

- [origamid – flexbox-guia-completo](#)
- [https://htmlreference.io/](#)
- [https://cssreference.io/](#)
- [MDN webdocs - Conceitos Basicos do Flexbox](#)
- [w3schools - flexbox](#)
- [css-tricks - a-guide-to-flexbox](#)
- [CSS Grid Layout e Flexbox - Quando Utilizar](#)

DUVIDAS E QUESTIONAMENTOS

The End