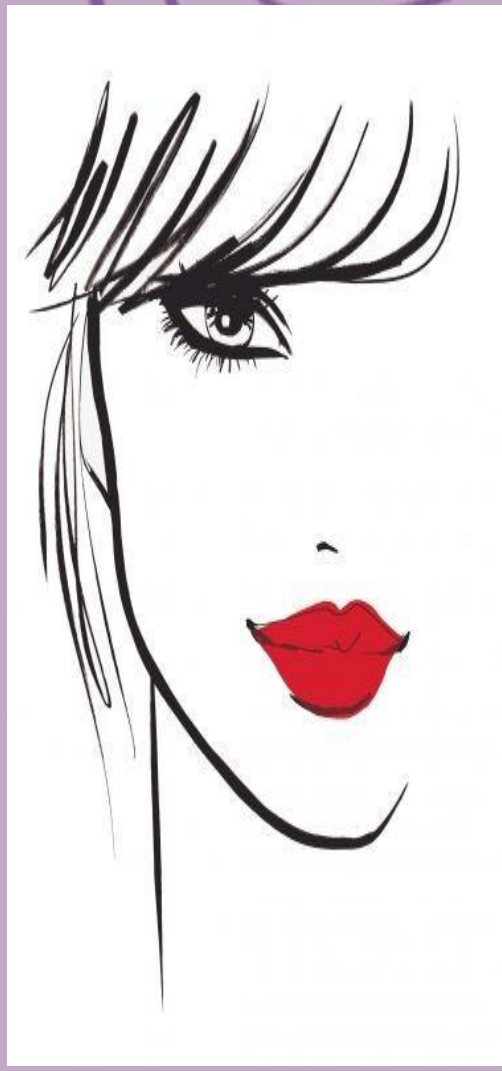
Bom dia
Descodificadas!

Flexbox

Devido as limitações de float e position principalmente em relação a resposividade, a ferramenta Flexbox (de Flexible Box) foi criada para tornar tarefas de posicionamento mais simples e funcionais.



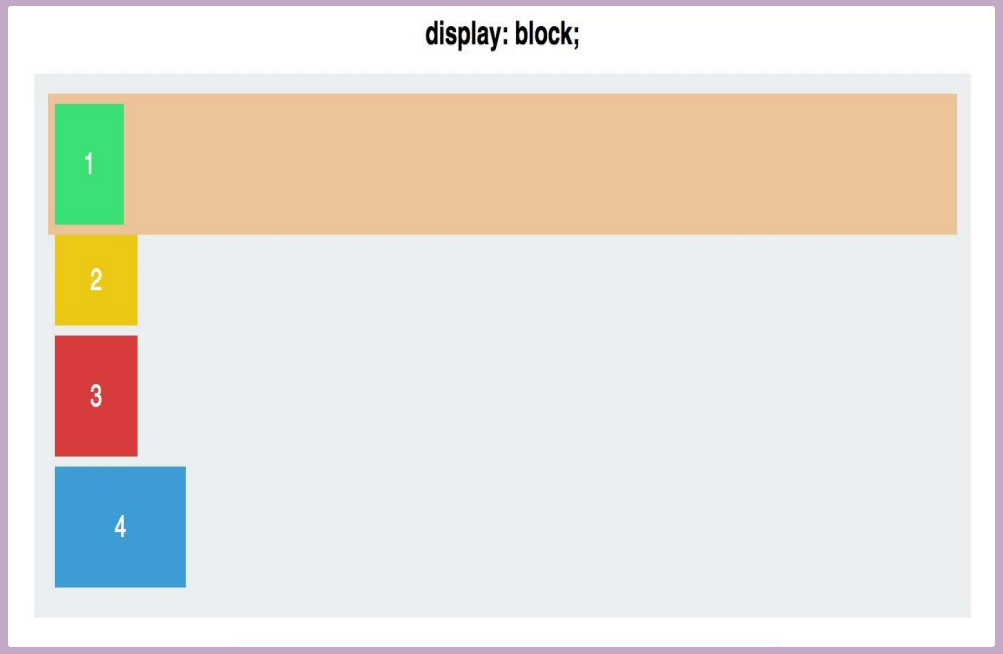
Flexbox

Algumas tarefas que consideramos básicas em um layout, como centralização vertical de um elemento-filho com relação a um elemento-pai ou fazer com que elementos-filhos ocupem a mesma quantidade de espaço, ou colunas terem o mesmo tamanho independente da quantidade de conteúdo interno, era quase impossível ou muito trabalhosa!

1ª propriedade: Display: Flex

O princípio base do Flexbox é tornar os layouts flexíveis e intuitivos.

Para conseguir isso, ele permite que os containers decidam por si mesmos como distribuir seus filhos de forma uniforme — incluindo seu tamanho e espaço entre eles.



Temos quatro divs coloridas de vários tamanhos, colocamos dentro de uma div container cinza.

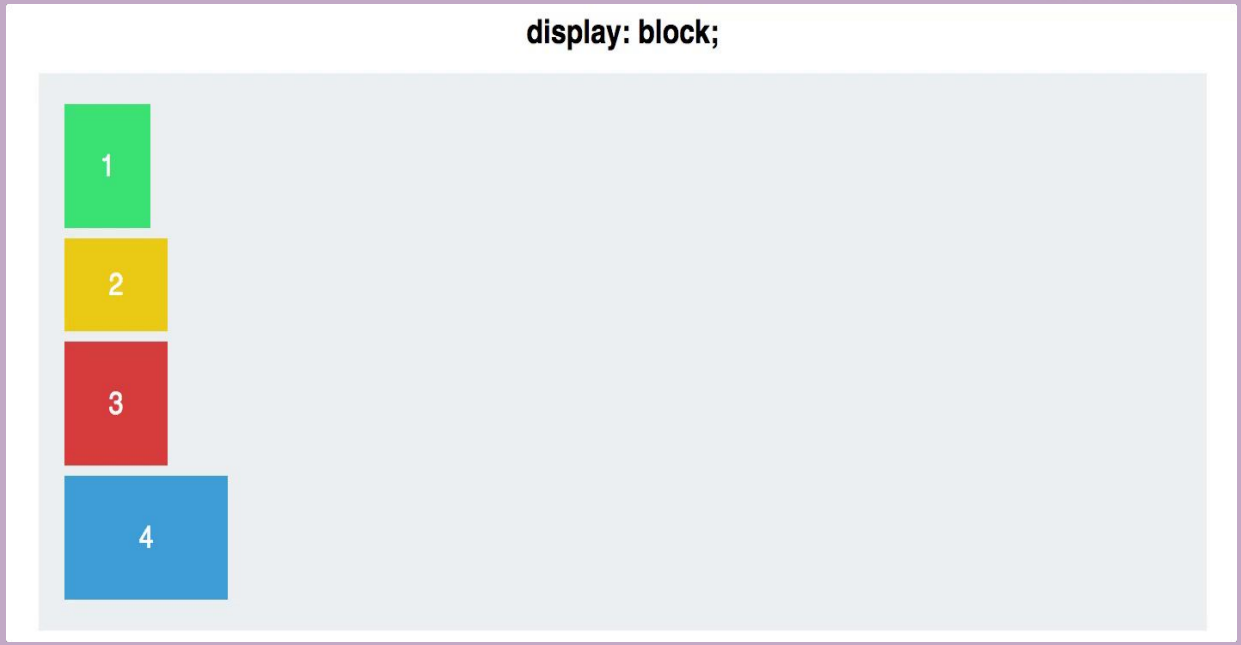
A partir de agora, cada div foi setada como padrão para display: block.

Assim cada quadrado ocupa toda a largura de sua linha.

1ª propriedade: Display: Flex

Para começar com o Flexbox, precisamos colocar o container em um flex container.

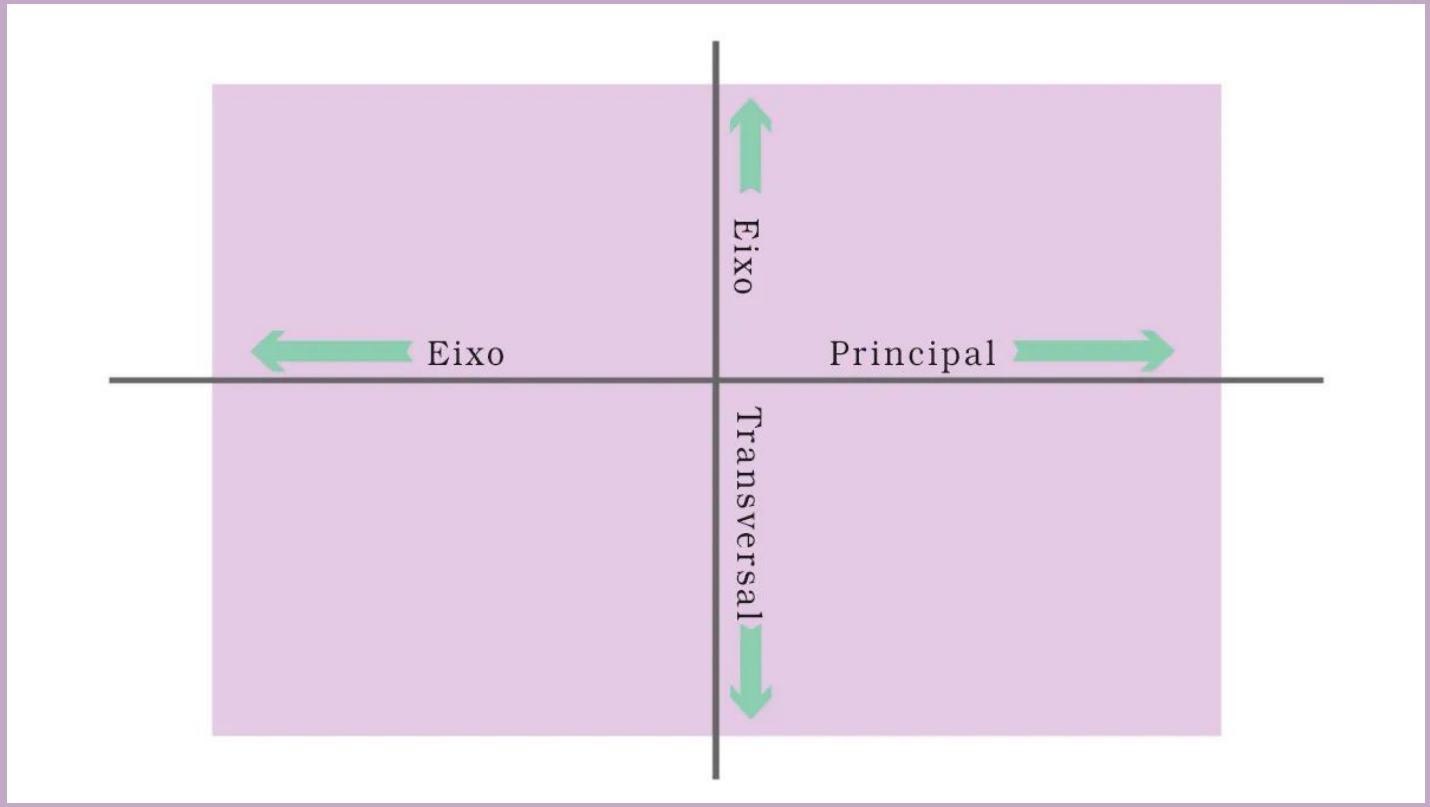
```
#container {
        display: flex;
}
```



Agora podemos começar à posicioná-los nesse contexto, com muito menos dificuldade do que com CSS tradicional.

2ª propriedade: Flex Direction

Um Flexbox container tem dois eixos: um eixo principal e um eixo transversal, que por padrão se parece assim:



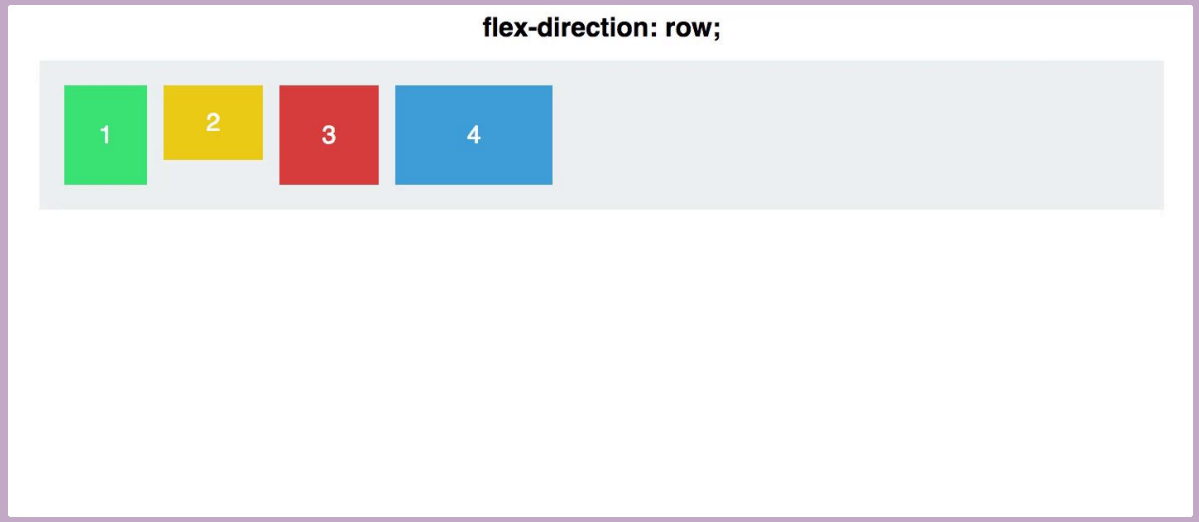
2ª propriedade: Flex Direction

Por padrão, os itens são dispostos ao longo do eixo principal, da esquerda para a direita.

Essa é a razão pela qual seus quadrados são por padrão colocados em uma linha horizontal quando você aplica *display: flex.*

Flex-direction, no entanto, faz rotacionar o eixo principal.

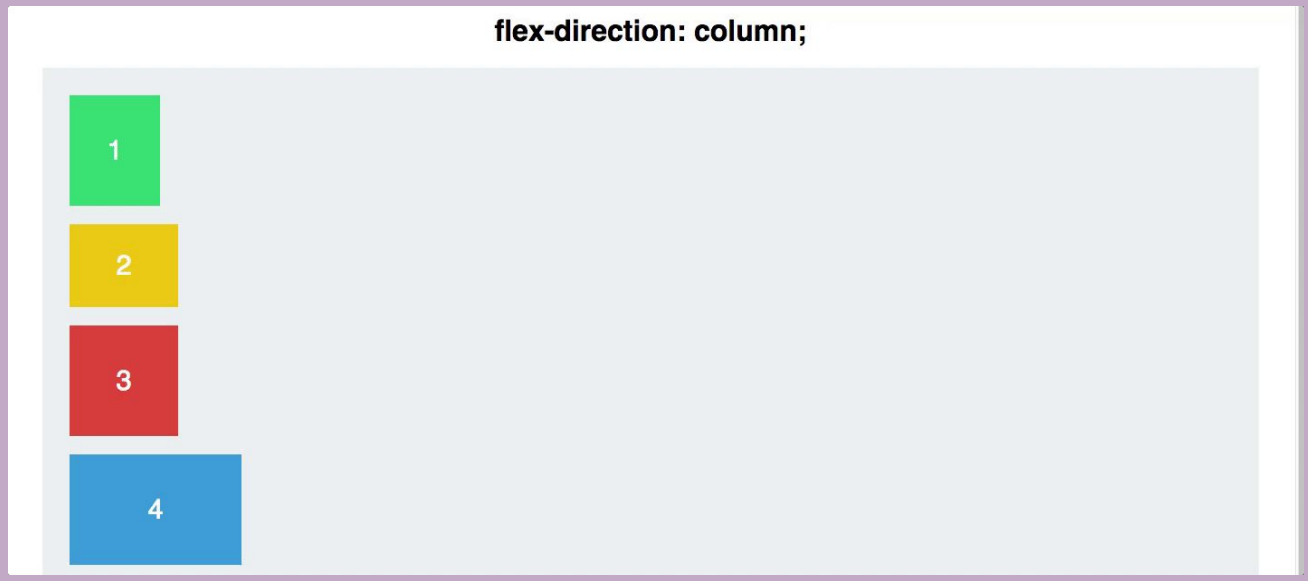
```
#container {
 display: flex;
 flex-direction: column;
}
```



2ª propriedade: Flex Direction

Há uma importante distinção a fazer aqui: *flex-direction: column* não alinha os quadrados no eixo transversal em vez do eixo principal. Faz o próprio eixo principal ir de horizontal para vertical.

Há um par de outras opções para *flex-direction*, bem como: *row-reverse* e *column-reverse.*



3ª propriedade: Justify Content

Justify-content **controla como se alinha itens no eixo principal.**

Temos cinco comandos para usar *justify-content:*

Flex-start
Flex-end
Center
Space-between
Space-around

```
#container {
 display: flex;
 flex-direction: row;
 justify-content:
flex-start;
}
```

justify-content: flex-start;

4ª propriedade: Align Items

Como o *justify-content* funciona ao longo do eixo principal, *align-items* se aplica ao eixo transversal.

Temos cinco comandos para usar *Align Items:*

```
#container {
 display: flex;
 align-items: stretch |
flex-start | flex-end |
center | baseline;
}
```

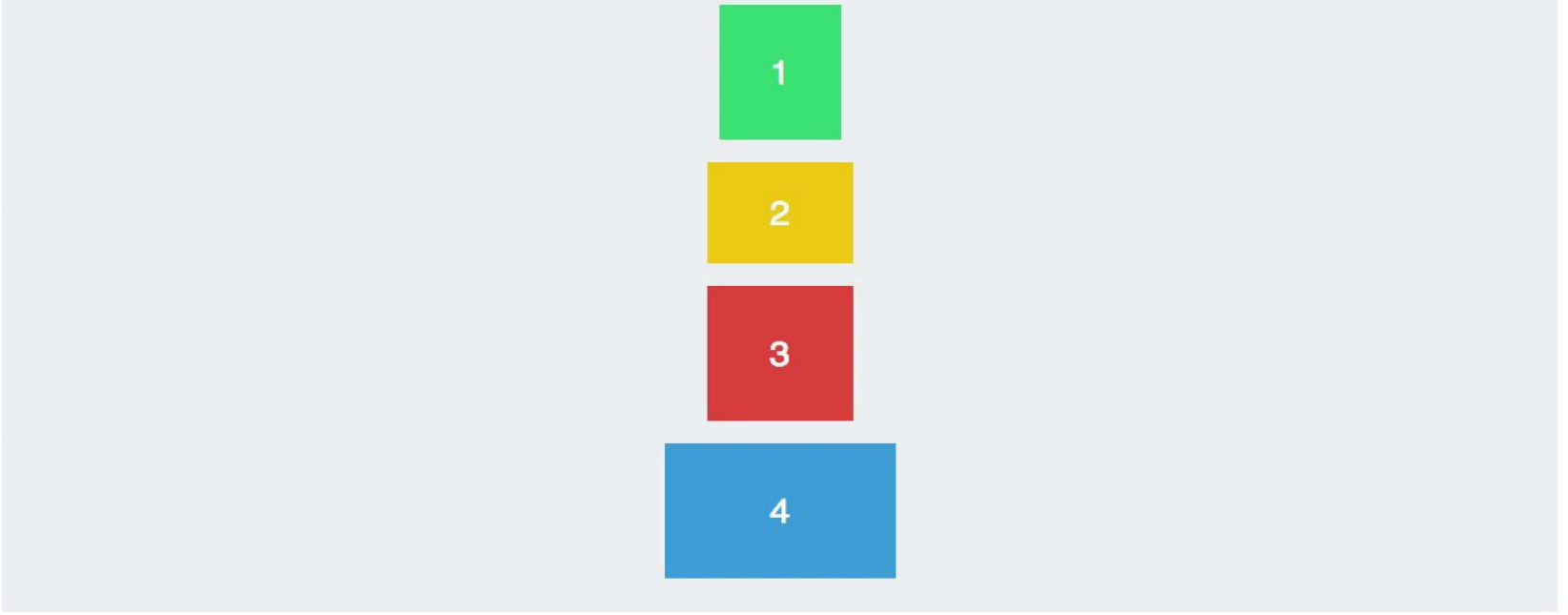
flex-start
flex-end
center
stretch
baseline

4ª propriedade: Align Items

Vamos combinar o *justify-content* e o *align-items* e ver como a centralização funciona diferente para os dos comandos do flex-direction:

flex-direction: column;

justify-content: center; align-items: center;



5ª propriedade: Align Self

Align-self permite manipular manualmente o alinhamento de um elemento em particular. É basicamente substituir o align-items para um quadrado. Todas as propriedades são as mesmas em que segue o align-items do container.

```
#container {
 align-items: flex-start;
}
.square#one {
 align-self: center;
}
// Apenas esse quadrado será centralizado.
```

Vamos aplicar o *align-self* a dois quadrados, e para o restante, aplicamos *align-items: center* e *flex-direction: row.*

Você encontrará todas as orientações e conteúdos respectivos a cada semana na nossa plataforma!

Bons estudos!!

descodificadas

descodificadas

Sigam nossas redes sociais!!!