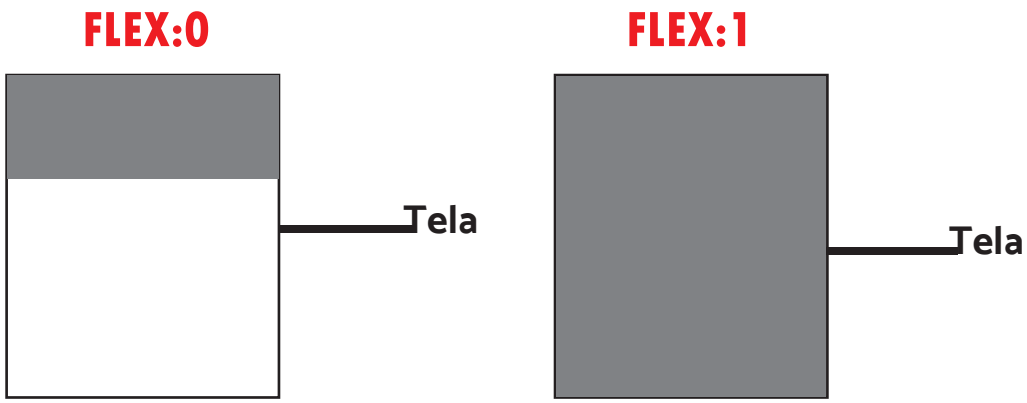
LAYOUT COM FLEXBOX

Um componente pode especificar o layout de seus filhos usando o algoritmo Flexbox. O Flexbox é projetado para fornecer um layout consistente em diferentes tamanhos de tela.

.

PROPIEDADE FLEX

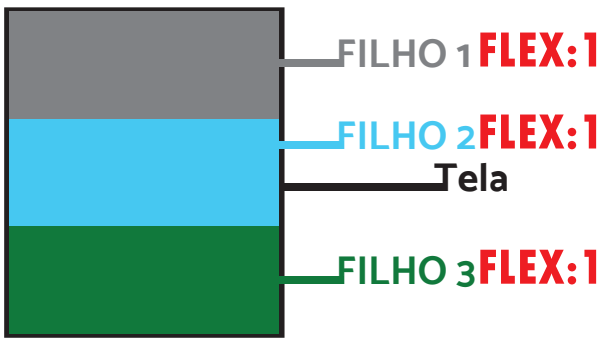
Vamos imaginar uma view e, dentro dela, criaremos um quadrado cinza. Nele, vamos colocar a propriedade flex com os valores 0 e 1 em duas situações diferentes.



Quem é o componente pai e quem é o filho ?

A propriedade flex: 1 faz com que o elemento cinza cresça para ocupar toda a área disponível no seu contêiner pai.

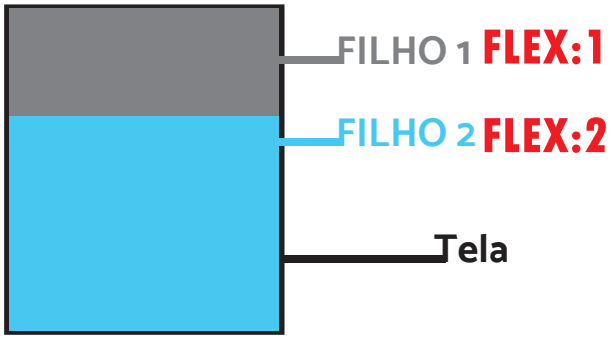
E ao colocar vários containers filhos com propriedade **FLEX:1 ?**



Se utilizarmos mais de um **FLEX:1** os elementos vão tentar ocupar o mesmo tamanho do seu pai. tipo 33% 33% 33% !

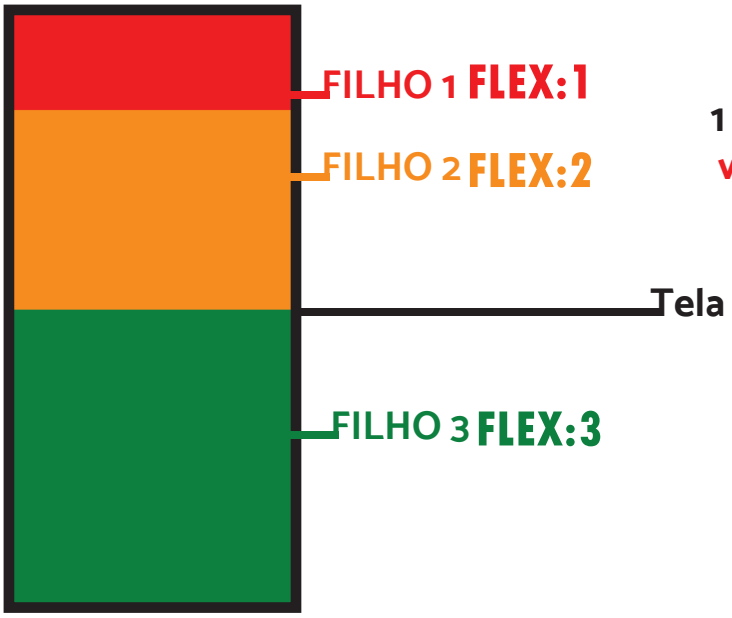
LAYOUT COM FLEXBOX

Fique atento nos próximos exemplos. Iremos mostrar o que acontece quando utilizamos o flex com um valor maior em um elemento e, em outro, um valor menor.



Se utilizarmos um valor maior para **FLEX:2** o nosso componente vai tentar calcular uma área que fique 75% maior que a do componente que está com um valor menor para o **FLEX:1**

No exemplo a seguir, as visualizações vermelha, laranja e verde são todas filhas do contêiner que foi definido com flex: 1 (tela). A visualização vermelha usa flex: 1, a visualização laranja usa flex: 2 e a visualização verde usa flex: 3.



1 + 2 + 3 = 6, o que significa que a visualização vermelha obterá 1/6 do espaço, a laranja 2/6 do espaço e a verde 3/6 do espaço.

.Agora que entendemos como os componentes se comportam com a propriedade flex, vamos entender mais sobre os métodos que podem ser utilizados quando o flex está ativo,

Isso significa que podemos mudar a forma como os contêineres filhos irão se organizar dentro do contêiner pai.

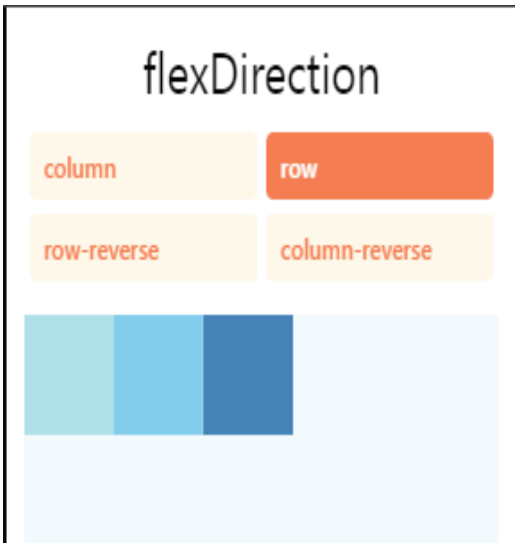
PROPIEDADES COM FLEXBOX

flexDirectioncontrola a direção na qual os filhos de um nó são dispostos. Isso também é chamado de eixo principal. O eixo cruzado é o eixo perpendicular ao eixo principal, ou o eixo no qual as linhas de encapsulamento são dispostas.

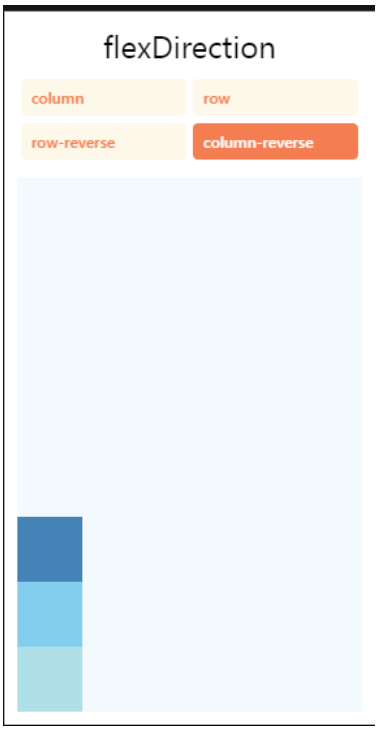
flexDirection-
controla a direção na qual os filhos de um nó são dispostos. Isso também é chamado de eixo principal. O eixo cruzado é o eixo perpendicular ao eixo principal, ou o eixo no qual as linhas de encapsulamento são dispostas.

column(valor padrão)
Alinhar filhos de cima para baixo.

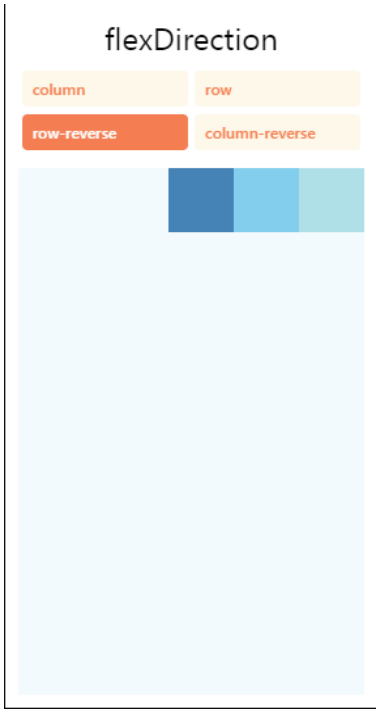
row
Alinhe os filhos da esquerda para a direita.



column-reverse
Alinhe os filhos de baixo para cima.



row-reverse
Alinhe os filhos da direita para a esquerda.



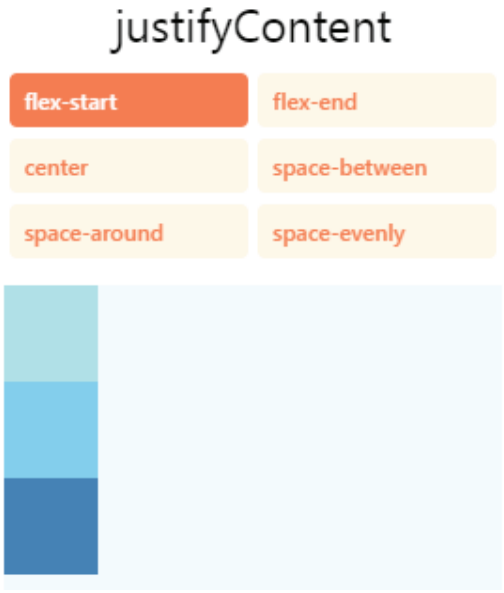
PROPIEDADES COM FLEXBOX

justifyContent-

descreve como alinhar os filhos dentro do eixo principal do contêiner deles. Por exemplo, você pode usar essa propriedade para centralizar um filho horizontalmente dentro de um contêiner com flexDirectionset to row.

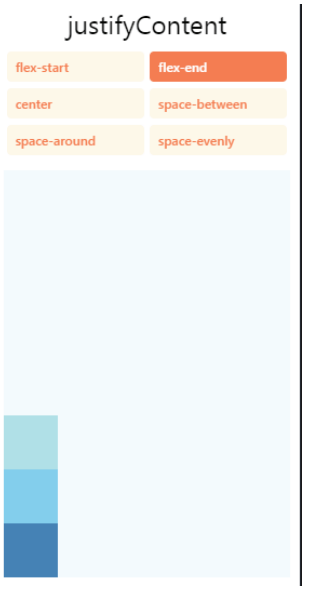
flex-start(valor padrão)

Alinha os filhos de um contêiner ao início do eixo principal do contêiner.



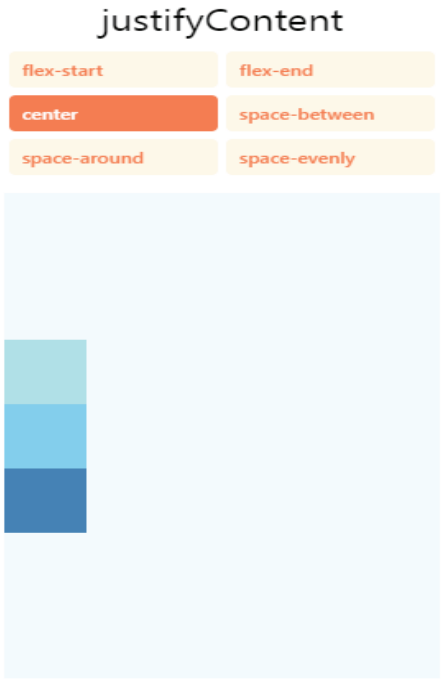
flex-end

Alinha os filhos de um contêiner ao final do eixo principal do contêiner.



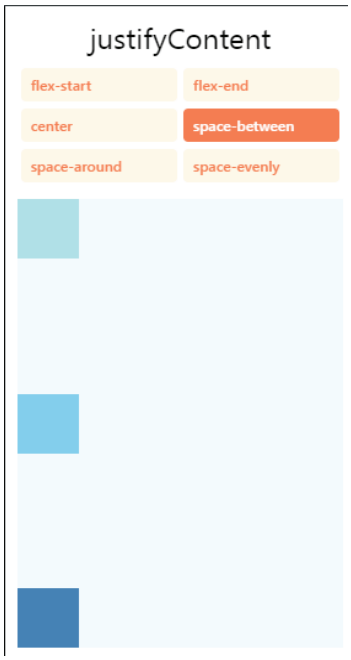
center:

Alinhe os filhos de um contêiner no centro do eixo principal do contêiner.



space-between:

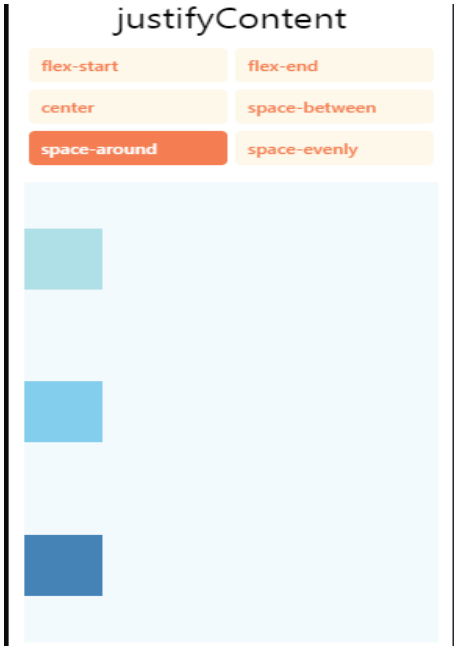
Distribua uniformemente os filhos no eixo principal do contêiner, distribuindo o espaço restante entre os filhos.



justifyContent-

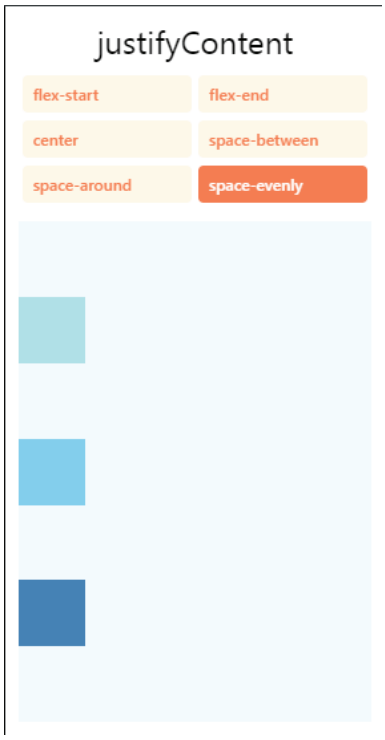
space-around:

cria um espaço uniforme entre os filhos no eixo principal do contêiner, distribuindo o espaço restante ao redor dos filhos. Comparado a space-between, usar space-around resultará em espaço sendo distribuído para o início do primeiro filho e o fim do último filho.



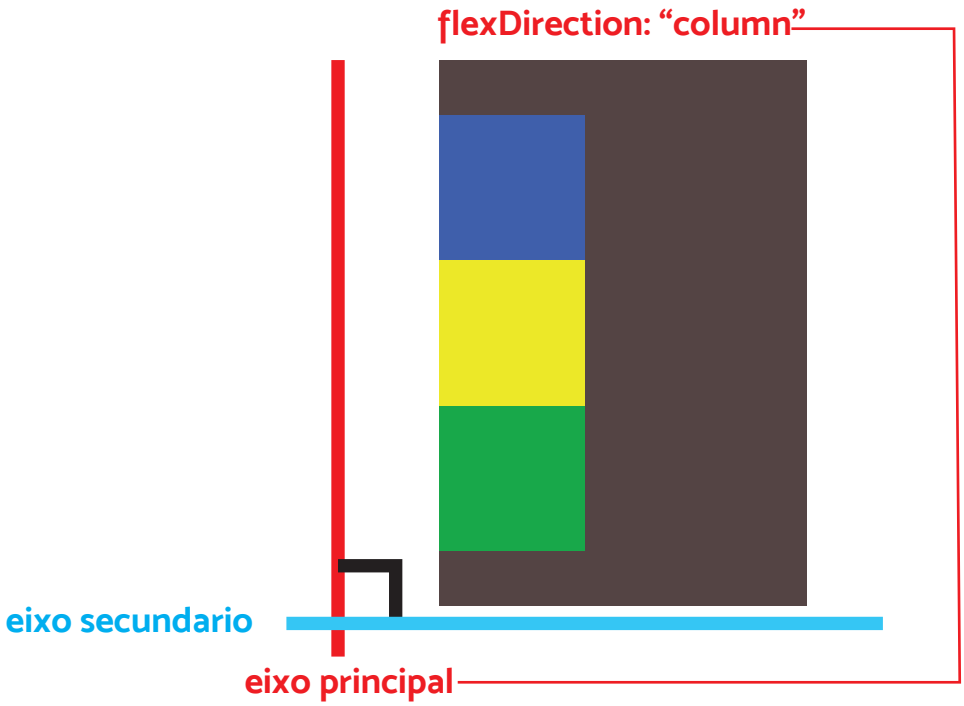
space-evenly:

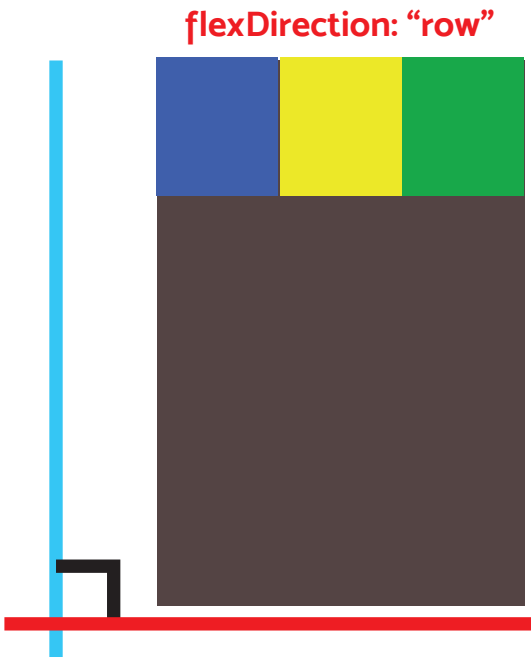
Distribui uniformemente os filhos dentro do contêiner de alinhamento ao longo do eixo principal. O espaçamento entre cada de item é exatamente o mesmo.



justifyContent-

Estamos falando muito sobre o eixo principal com justify-content e estamos alinhando nossos itens a ele. Mas quem seria o outro eixo, o eixo secundário? Nosso eixo secundário, que na documentação sempre se refere como eixo perpendicular, é o eixo que não está sendo utilizado no flex-direction ou o eixo que forma um ângulo de 90° com o eixo principal. Veja nos exemplos





Consegue identificar agora quem é o eixo principal e o eixo secundário?

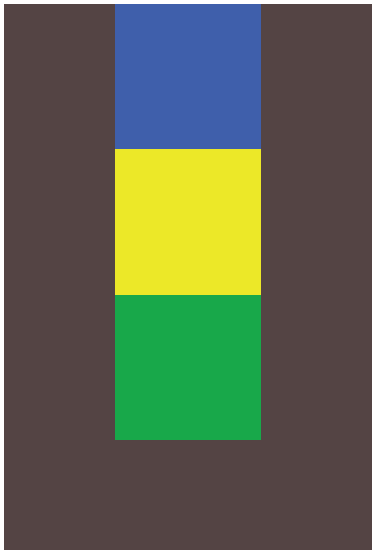
É muito importante sabermos sobre isso porque o próximo método que vamos ver é o align-items, e ele trabalha justamente com o eixo perpendicular ao eixo principal.

alignItems-

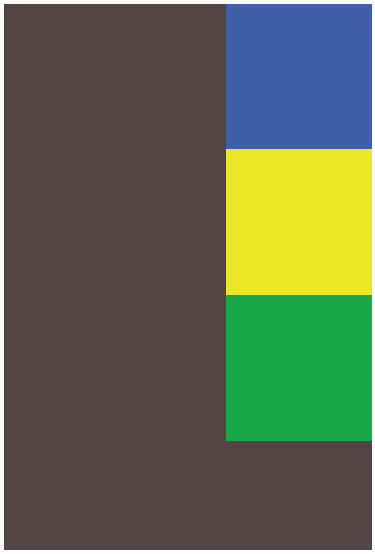
alignItems descreve como alinhar filhos ao longo do eixo perpendicular de seu contêiner. É muito semelhante a, justifyContentmas em vez de aplicar ao eixo principal, alignItems aplica-se ao eixo perpendicular.
Levaremos em consideração que o
flexDirection será definido como padrão “column”.

alignItems:’start’

alignItems:’center’



alignItems:’end’



Se trocássemos para flexDirection: ’row’, como ficariam alinhados os quadrados dentro de cada tela? Desenhe 3 em cada tela.

alignItems:’start’

alignItems:’center’

alignItems:’end’

Agora vamos codar tudo isso isso.