

FIAP

Mobile App Development

View, Flexbox e Style

View

View

O que é a View?

É o componente básico do React-native. É um container já declarado com layout Flexbox e que na compilação é traduzido para o container da respectiva tecnologia.



View

View Function Component Example ⓘ ↗



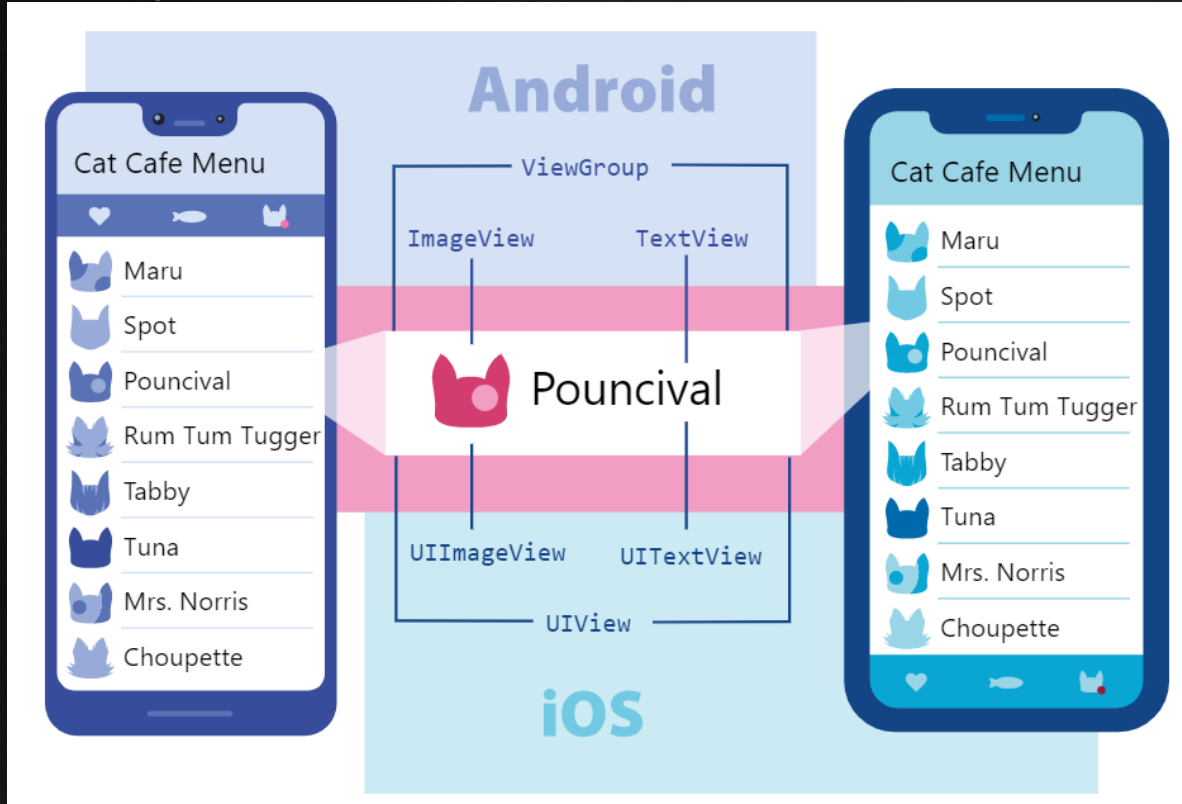
```
import React from 'react';
import {View, Text} from 'react-native';

const ViewBoxesWithColorAndText = () => {
  return (
    <View
      style={{
        flexDirection: 'row',
        height: 100,
        padding: 20,
      }}>
      <View style={{backgroundColor: 'blue', flex: 0.3}} />
      <View style={{backgroundColor: 'red', flex: 0.5}} />
      <Text>Hello World!</Text>
    </View>
  );
};

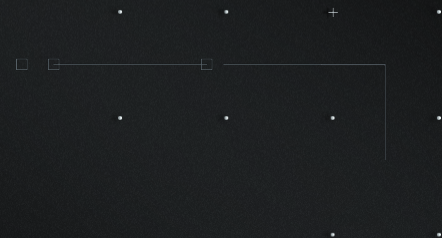
export default ViewBoxesWithColorAndText;
```



View



FlexBox



Flexbox

O que é o Flexbox?

É um modelo de **layout** criado para conseguirmos organizar componentes de maneira **dinâmica**, simples.

É muito utilizado pela sua adaptabilidade da disposição e tamanho dos componentes, tornando uma ótima escolha para designs responsivos

Flexbox

Entendendo o conceito básico

A organização do flexbox se dá pela divisão dos componentes entre containers e itens.

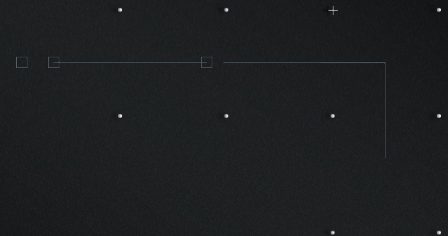
É muito utilizado pela sua adaptabilidade da disposição e tamanho dos componentes, tornando uma ótima escolha para designs responsivos



Flexbox

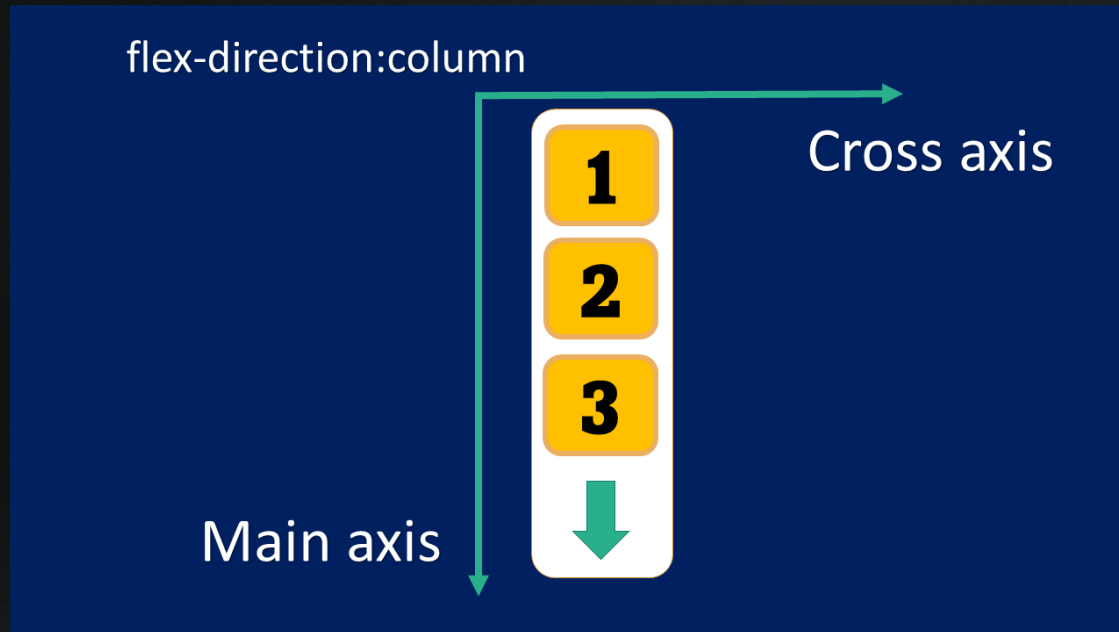
Nomenclaturas importantes

- Eixo principal
- Eixo transversal
- Item
- Container



Flexbox

Nomenclaturas importantes

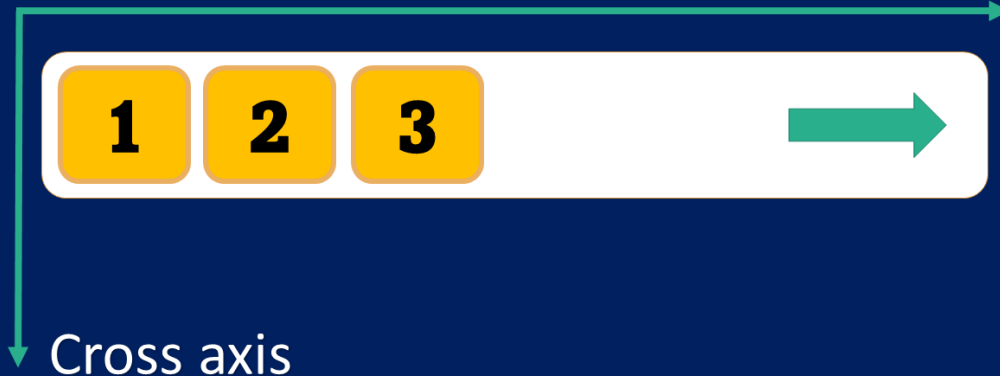


Flexbox

Nomenclaturas importantes

flex-direction: row

Main axis



View

```
import React from 'react';
import {View, Text} from 'react-native';

const ViewBoxesWithColorAndText = () => {
  return (
    <View
      style={{
        flexDirection: 'row'
        height: 100,
        padding: 20,
      }}>
      <View style={{backgroundColor: 'blue', flex: 0.3}} />
      <View style={{backgroundColor: 'red', flex: 0.5}} />
      <Text>Hello World!</Text>
    </View>
  );
};

export default ViewBoxesWithColorAndText;
```

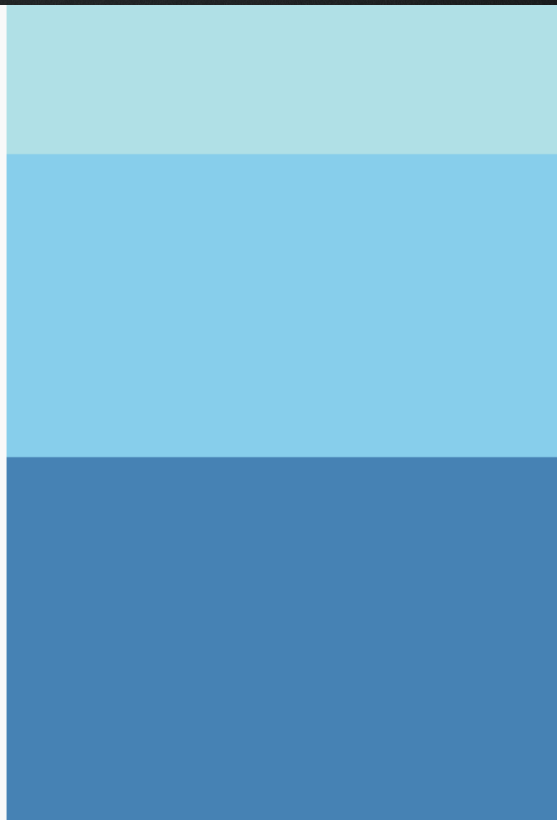


View

```
import React from 'react';
import {View} from 'react-native';

const FlexDimensionsBasics = () => {
  return (
    // Try removing the `flex: 1` on the parent View.
    // The parent will not have dimensions, so the children can't expand.
    // What if you add `height: 300` instead of `flex: 1`?
    <View style={{flex: 1}}>
      <View style={{flex: 1, backgroundColor: 'powderblue'}} />
      <View style={{flex: 2, backgroundColor: 'skyblue'}} />
      <View style={{flex: 3, backgroundColor: 'steelblue'}} />
    </View>
  );
};

export default FlexDimensionsBasics;
```



Flexbox

flexDirection

Determina a direção e sentido que os itens são inseridos

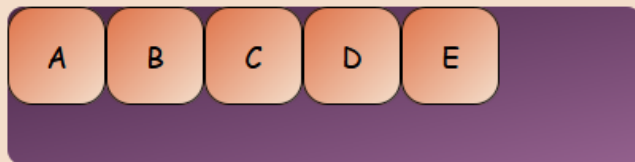
- row
- row-reverse
- **column (default do componente View)**
- column-reverse



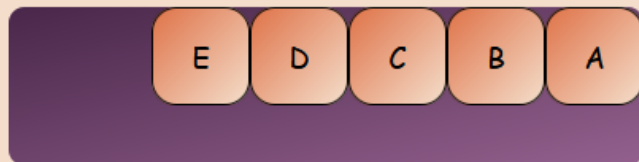
Flexbox

flex-direction

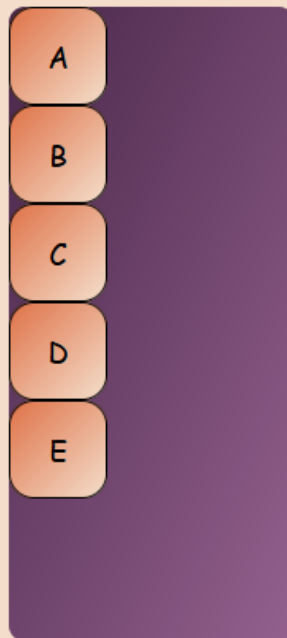
row



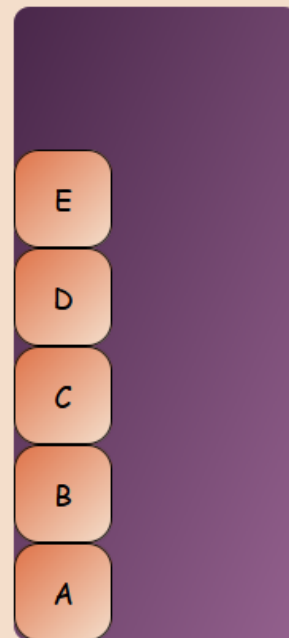
row-reverse



column



column-reverse



Flexbox

justifyContent

Determina a distribuição dos itens dentro do componente.

- **flex-start (default)**
- flex-end
- center
- space-between
- space-around
- space-evenly



Flexbox

justifyContent

flex-start

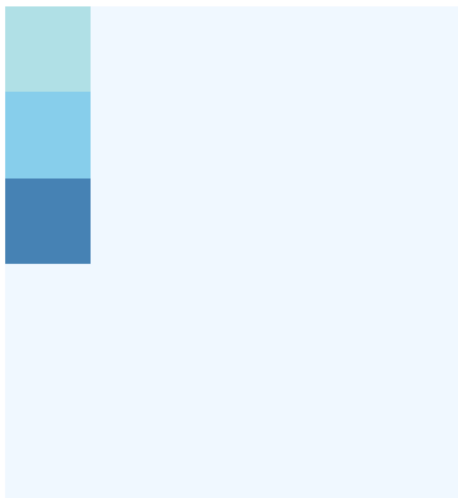
flex-end

center

space-between

space-around

space-evenly



justifyContent

flex-start

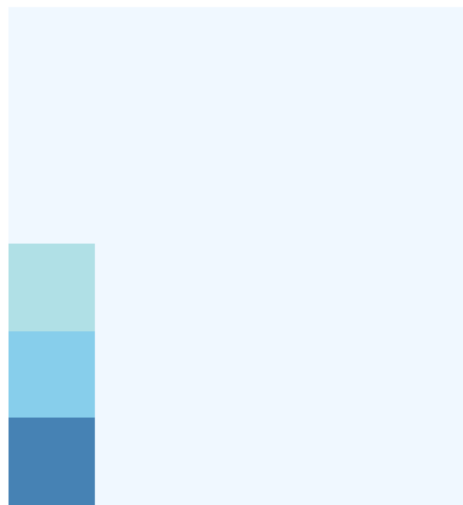
flex-end

center

space-between

space-around

space-evenly



Flexbox

justifyContent

flex-start

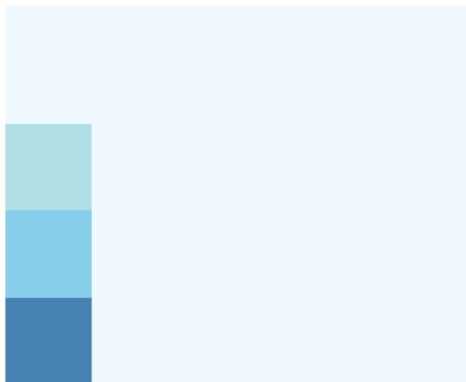
flex-end

center

space-between

space-around

space-evenly



justifyContent

flex-start

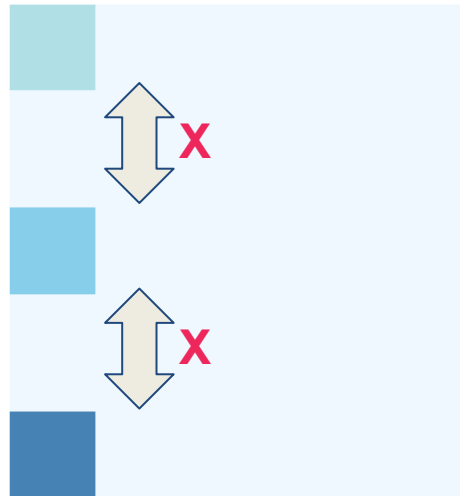
flex-end

center

space-between

space-around

space-evenly



Flexbox

justifyContent

flex-start

flex-end

center

space-between

space-around

space-evenly

| Y/2



| Y



| Y



| Y/2

justifyContent

flex-start

flex-end

center

space-between

space-around

space-evenly

| Z



| Z



| Z



| Z

Flexbox

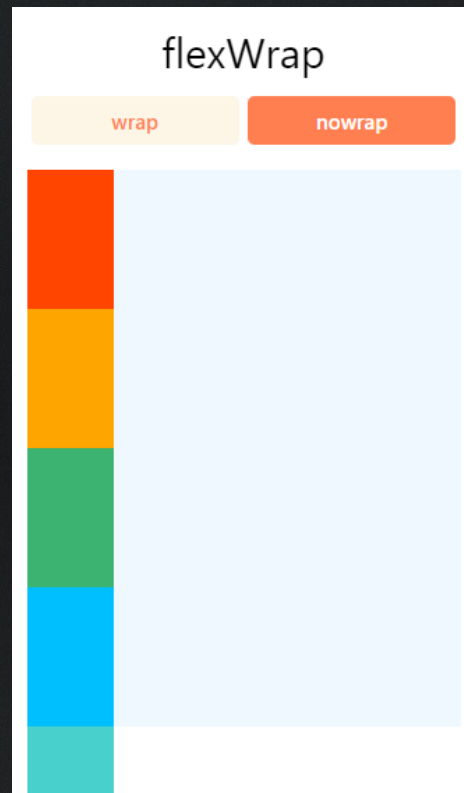
flexWrap

Determina se, ao ter mais itens que o tamanho do container, os itens irão “vazar” ou se irão formar uma nova linha ou coluna

- wrap
- nowrap (default)



Flexbox



Flexbox

alignContent

Determina a distribuição das linhas/colunas quando usamos flexWrap: wrap

- **flex-start (default)**
- flex-end
- stretch
- center
- space-between
- space-around



Flexbox

alignContent

flex-start

flex-end

stretch

center

space-between

space-around



alignContent

flex-start

flex-end

stretch

center

space-between

space-around



Flexbox

alignContent

flex-start

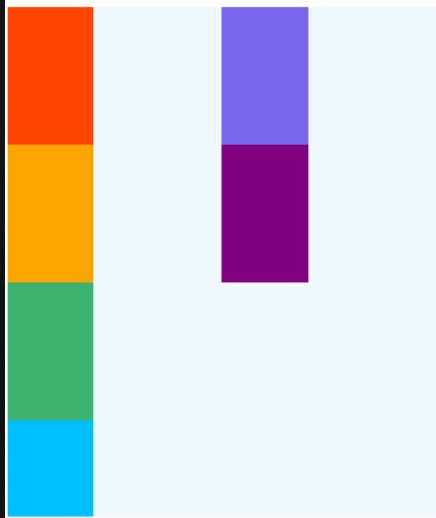
flex-end

stretch

center

space-between

space-around



alignContent

flex-start

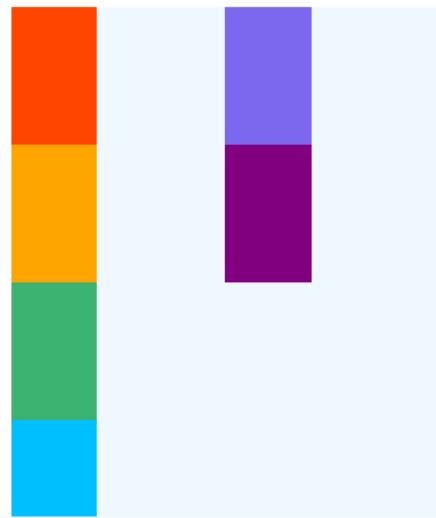
flex-end

stretch

center

space-between

space-around



Flexbox

alignContent

flex-start

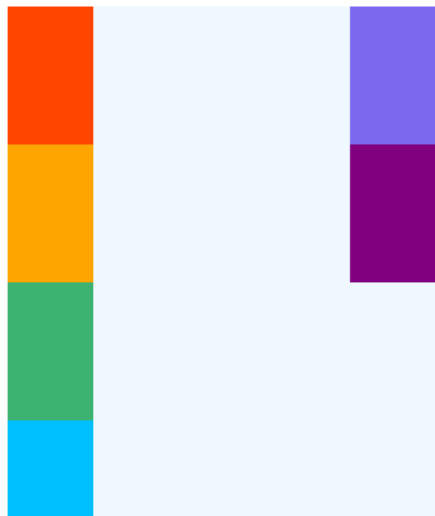
flex-end

stretch

center

space-between

space-around



alignContent

flex-start

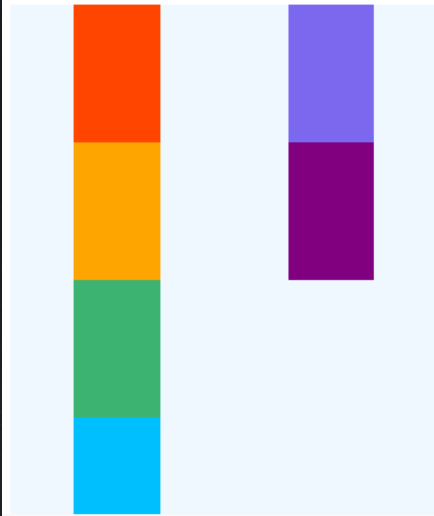
flex-end

stretch

center

space-between

space-around



Flexbox

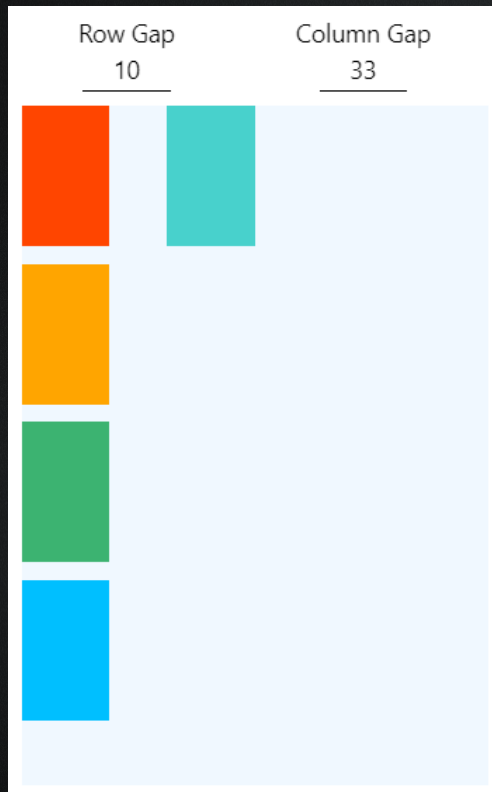
gap, rowGap, columnGap

Utilizado para determinar um valor fixo de distância entre os itens em um ou ambos os eixos

- 0 como default



Flexbox



Flexbox

width e height

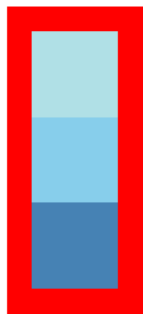
Determina altura e largura do seu item/container
O valor do container NÃO altera o valor de seus itens

- **auto (default)**
- pixels
- porcentagem



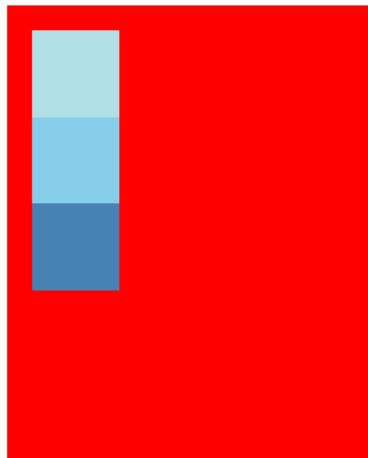
Flexbox

width auto 300 80%
height auto 200 60%



(Essa “sobra” é o padding de 15px)


width auto 300 80%
height auto 200 60%



Flexbox

width auto 300 80%

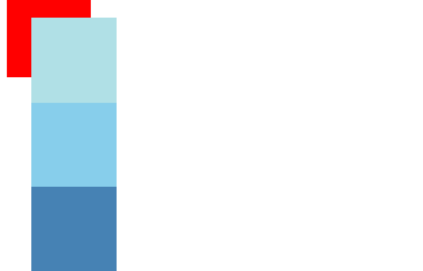
height auto 200 60%



A large red rectangular container is shown. On its left side, three smaller rectangular boxes are stacked vertically. The top box is light blue, the middle one is medium blue, and the bottom one is dark blue. The container's width is set to 300 (80%) and its height is set to 200 (60%).

width auto 50 80%


height auto 50 60%



A smaller red rectangular container is shown. On its left side, three smaller rectangular boxes are stacked vertically. The top box is light blue, the middle one is medium blue, and the bottom one is dark blue. The container's width is set to 50 (80%) and its height is set to 50 (60%).

width auto 50 80%

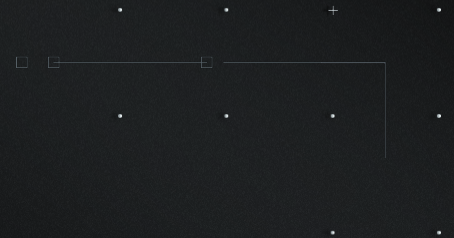
height auto 50 60%



A small red rectangular container is shown. On its left side, three smaller rectangular boxes are arranged horizontally. The leftmost box is light blue, the middle one is medium blue, and the rightmost one is dark blue. The container's width is set to 50 (80%) and its height is set to 50 (60%).

Se declaramos
flexWrap: wrap

Container



Flexbox

positioning

Determina a posição do item

- **relative (default)**
- **absolute**
- **right/left/top/bottom (default 0)**

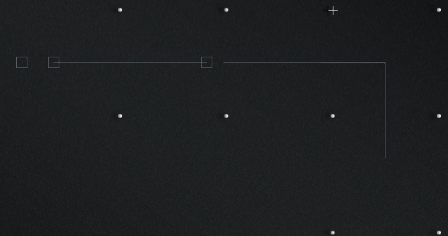


Flexbox

positioning

Para determinarmos a posição de um item, utilizamos **top** ou **bottom**, e **left** ou **right**.


Não utilizamos posições que atuem no mesmo eixo juntos



Flexbox

position


relative absolute



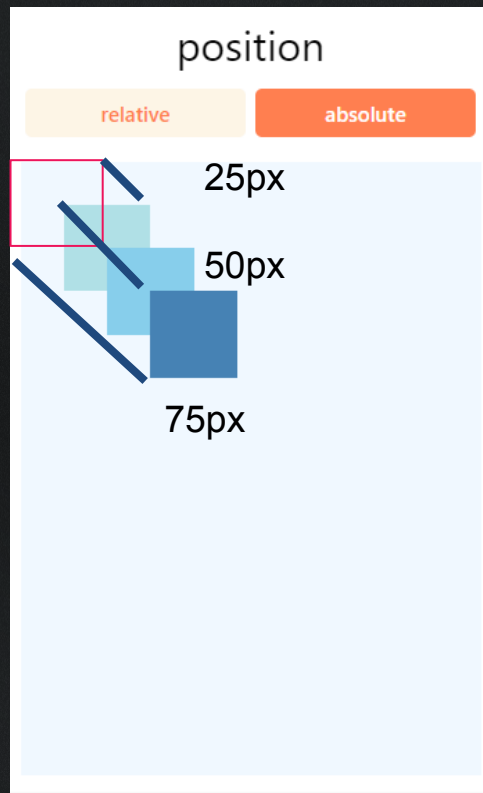
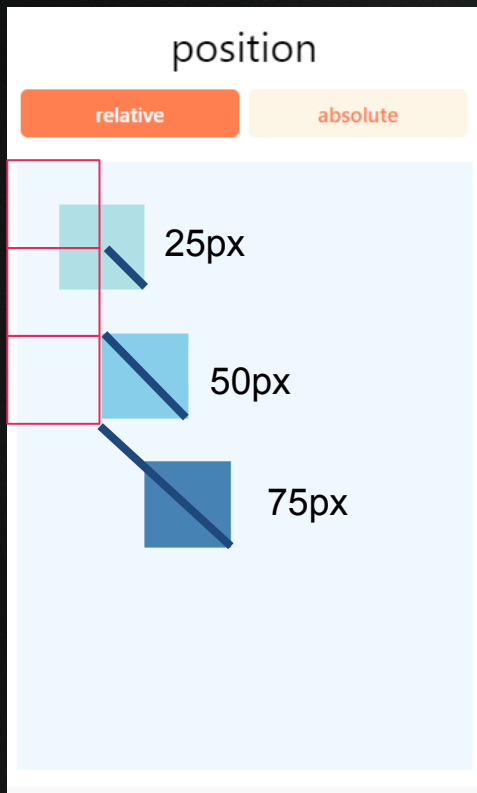
width: 50px
height: 50px
left: 25px
top: 25px
left: 50px
top: 50px
left: 75px
top: 75px

position

relative absolute



Flexbox



Flexbox

alignItems

É utilizado para alinhar os itens dentro do container. Semelhante ao `justifyContent`, porém o `alignItems` atua no **eixo transversal**, enquanto o `justifyContent` atua no **eixo principal**

- **stretch (default)**
- `flex-start`
- `flex-end`
- `center`
- `baseline`



Flexbox

alignItems

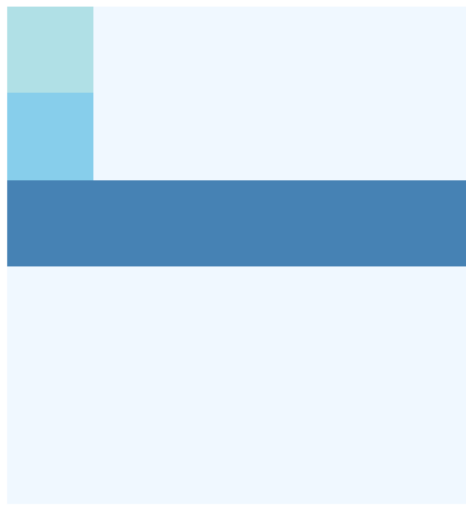
stretch

flex-start

flex-end

center

baseline



alignItems

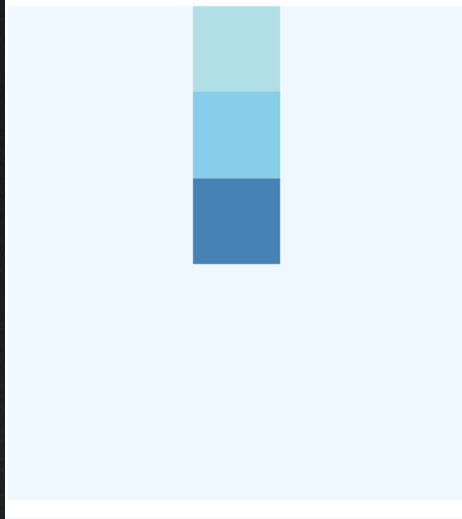
stretch

flex-start

flex-end

center

baseline



Flexbox

alignItems

stretch

flex-start

flex-end

center

baseline



alignItems

stretch

flex-start

flex-end

center

baseline



Flexbox

alignItems

O stretch só funciona quando não há um tamanho definido no eixo transversal. No caso anterior, 2 itens tinham width: 50px, apenas o item sem width que foi modificado



Flexbox

alignSelf

É semelhante ao `alignItems`, porém é aplicado no **item** e não no container. Se declarado, sobrescreve o `alignItems`

- **stretch (default)**
- `flex-start`
- `flex-end`
- `center`
- `baseline`



Flexbox

alignSelf

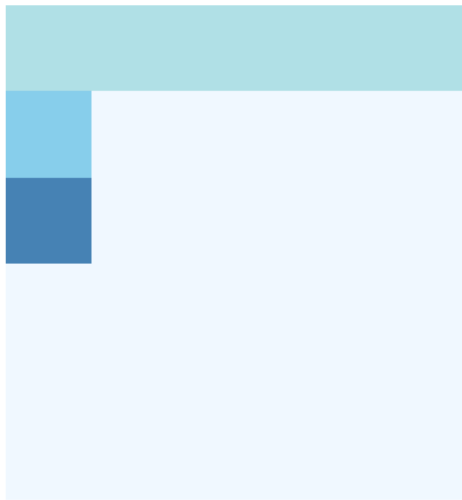
stretch

flex-start

flex-end

center

baseline



alignSelf

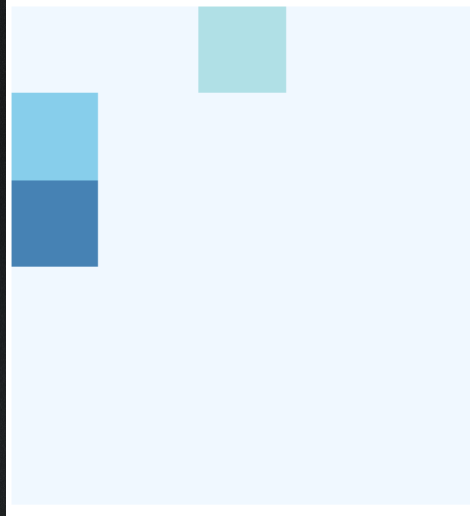
stretch

flex-start

flex-end

center

baseline



Flexbox

alignSelf

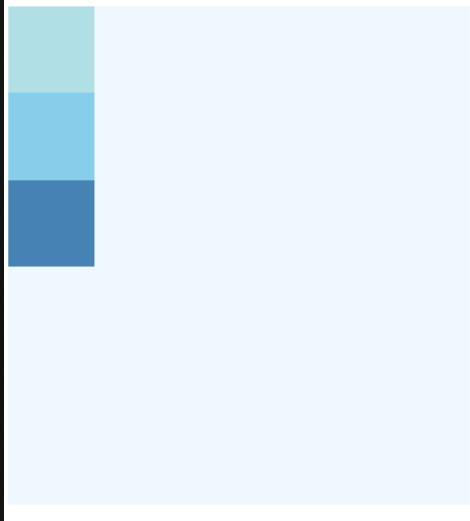
stretch

flex-start

flex-end

center

baseline



alignSelf

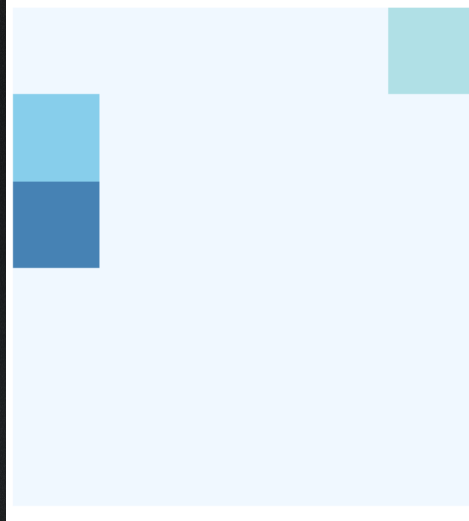
stretch

flex-start

flex-end

center

baseline



Flexbox

flexBasis, flexShrink, flexGrow

São parâmetros que definem como os itens irão se comportar para dividir o espaço do container entre si

- flexBasis: Define o valor “default” do item no eixo principal. Similar a height ou width
- flexShrink: Define o quanto um item irá diminuir para pode ocupar o espaço
- flexGrow: Define o quanto um item irá aumentar para ocupar o espaço disponível

Flexbox

flexBasis, flexShrink, flexGrow

FlexShrink é utilizado quando o espaço do container não é suficiente para todos os itens. Os itens são redimensionados para um tamanho menor, de acordo com o peso dado a cada um deles.



Flexbox

flexBasis, flexShrink, flexGrow

FlexGrow é utilizado quando o espaço do container é maior que o necessário para todos os itens. Os itens são redimensionados para um tamanho maior, de acordo com o peso dado a cada um deles.



Flexbox

flexBasis, flexShrink, flexGrow

flexBasis: O flexBasis default é o tamanho do item no eixo principal.

flexShrink: Valores positivos (float), com o default sendo 0

flexGrow: Valores positivos (float), com o default sendo 1



Flexbox

flex

flex é uma abreviação dos 3 parâmetros anteriores. Quando declaramos **flex**, estamos declarando flex-grow, flex-shrink, flex-basis nessa ordem

exemplos:

flex: 1 1 auto



Flexbox

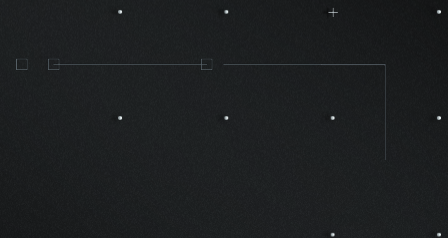
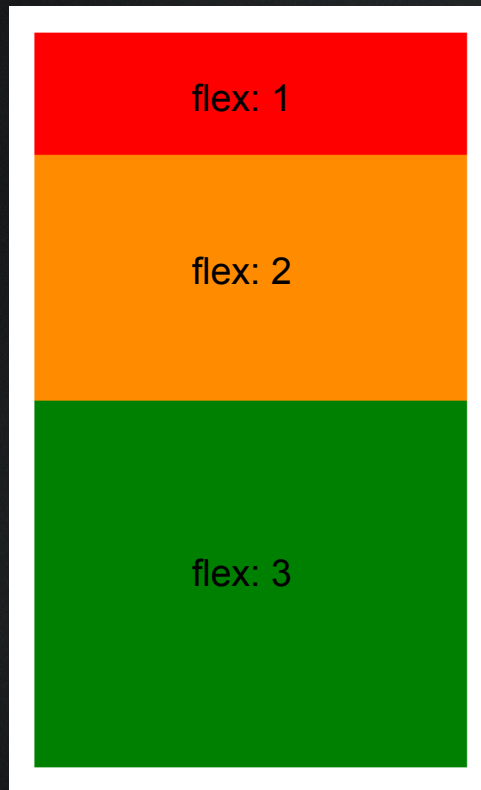
flex (apenas 1 valor)

Quando utilizamos flex com apenas 1 valor, é o mesmo que declararmos **flex: 1 0 auto**

É muito comum utilizarmos apenas **flex: x**, para dividirmos o espaço entre os itens, já que é bem mais incomum uma necessidade de utilizar flexBaseline ou flexShrink



Flexbox



View

```
import React from 'react';
import {View, Text} from 'react-native';

const ViewBoxesWithColorAndText = () => {
  return (
    <View
      style={{
        flexDirection: 'row',
        height: 100,
        padding: 20,
      }}>
      <View style={{backgroundColor: 'blue', flex: 0.3}} />
      <View style={{backgroundColor: 'red', flex: 0.5}} />
      <Text>Hello World!</Text>
    </View>
  );
};

export default ViewBoxesWithColorAndText;
```



Style e Styles Sheet

Style e Stylesheet

O que é ao style?

É uma prop recebida pelos componentes para aplicar style CSS nos componentes react

- backgroundColor
- color
- width/height
- atributos flexbox
- Entre outros (alguns específicos para certos componentes)

Style e Stylesheet

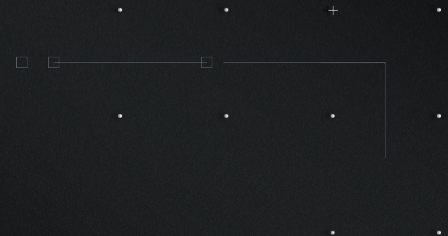
Text

- fontFamily
- fontSize
- fontStyle
- fontWeight
- textAlign
- textDecorationLine

Style e Stylesheet

Image

- backgroundColor
- borderColor
- borderRadius
- opacity
- overflow
- resizeMode



Style e Stylesheet

Stylesheet

É um objeto especial, que encapsula e valida o estilo a ser aplicado no componente. Temos 2 principais métodos, o create e o compose.

- Create: cria um stylesheet a partir de um objeto
- Compose: cria um stylesheet a partir de 2 stylesheets, ou partes deles

Style e Stylesheet

Stylesheet

É um objeto especial, que encapsula e valida o estilo a ser aplicado no componente. Temos 2 principais métodos, o create e o compose.

- Create: cria um stylesheet a partir de um objeto
- Compose: cria um stylesheet a partir de 2
stylesheets, ou partes deles. Se houver
repetição, o 1o stylesheet é sobrescrito pelo 2o

Style e Stylesheet

```
const page = StyleSheet.create({
  container: {
    flex: 1,
    padding: 24,
    backgroundColor: '#fff',
  },
  text: {
    fontSize: 30,
    color: '#000',
  },
});

const lists = StyleSheet.create({
  listContainer: {
    flex: 1,
    backgroundColor: '#61dafb',
  },
  listItem: {
    fontStyle: 'italic',
    fontWeight: 'bold',
  },
});

const container = StyleSheet.compose(page.container, lists.listContainer);
const text = StyleSheet.compose(page.text, lists.listItem);
```

Style Text

Style page.text

Style lists.listItem

Links

https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox

<https://reactnative.dev/docs/flexbox>

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

<https://reactnative.dev/docs/styleSheet>

Dúvidas, anseios, desabafos?

FIAP