
5 Widgets em flutter

Desenvolvimento de Aplicações para Dispositivos Móveis

Discente: Lucas da Silva Lima


Docente: Profº. Drº. Everton Coimbra

Agenda

1. Expanded;
 2. FittedBox;
 3. MaterialButton;
 4. Padding;
 5. ThemeData;
 6. Aplicação construída.
-

Expanded

Expandir um component filho, tal como: Row, Column ou Flex



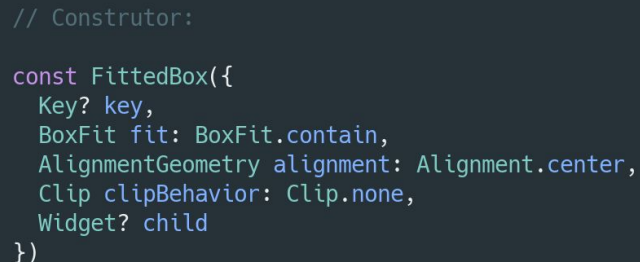
```
// Construtor:  
  
const Expanded({  
  Key? key,  
  int flex = 1,  
  required Widget child,  
})
```



```
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: Text('Expanded Row Sample'),  
    ),  
    body: Center(  
      child: Row(  
        children: <Widget>[  
          Expanded(  
            flex: 2, child: Container(color: Colors.amber, height: 100)),  
          Container(color: Colors.blue, height: 100, width: 50),  
          Expanded(  
            flex: 1, child: Container(color: Colors.amber, height: 100)),  
        ],  
      ),  
    ),  
  );  
}
```

FittedBox

Escale e posicione seus widgets filhos dentro deste de acordo com um ajuste desejado.



```
// Construtor:  
  
const FittedBox({  
  Key? key,  
  BoxFit fit: BoxFit.contain,  
  AlignmentGeometry alignment: Alignment.center,  
  Clip clipBehavior: Clip.none,  
  Widget? child  
})
```



```
@override
// ignore: non_constant_identifier_names
Widget build(BuildContext) {
  return Container(
    height: 400,
    width: 300,
    color: Colors.red,
    child: FittedBox(
      child: Image.network(
        'https://flutter.github.io/assets-for-api-docs/assets/widgets/owl-2.jpg'),
      fit: BoxFit.fill,
    ),
  );
}
```



```
@override
// ignore: non_constant_identifier_names
Widget build(BuildContext) {
  return Container(
    height: 400,
    width: 300,
    color: Colors.red,
    child: FittedBox(
      child: Image.network(
        'https://flutter.github.io/assets-for-api-docs/assets/widgets/owl-2.jpg',
        fit: BoxFit.contain,
      ),
    );
}
```

MaterialButton

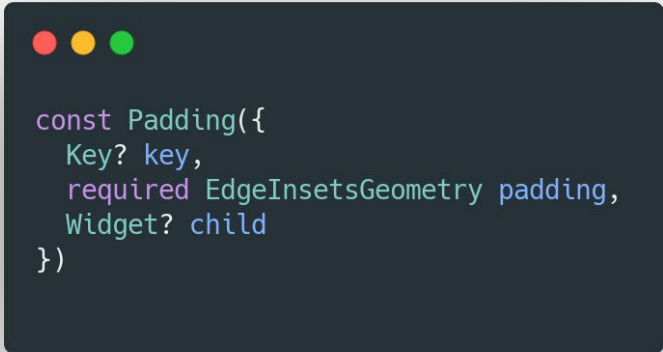
Uma classe que ajuda na construção de botões no padrão do material. Ela depende do *ButtonTheme* e do *theme* do ambiente.



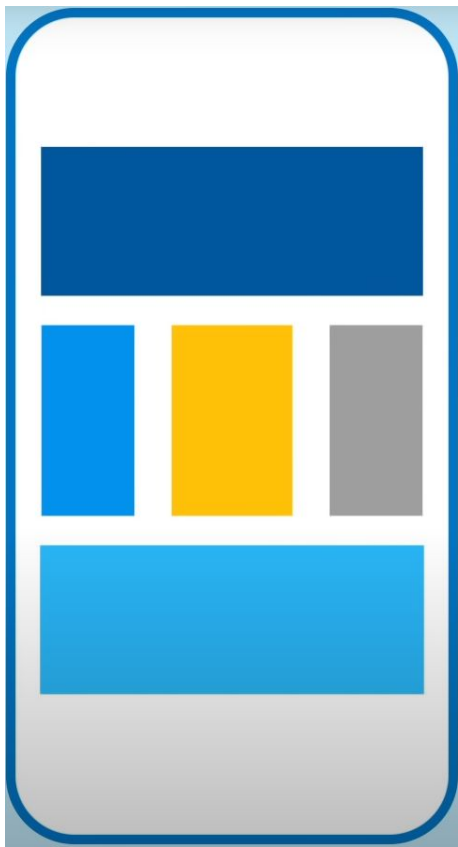
```
const MaterialButton({
  Key? key, required VoidCallback? onPressed, VoidCallback? onLongPress,
  ValueChanged<bool>? onHighlightChanged, MouseCursor? mouseCursor,
  ButtonTextTheme? textTheme, Color? textColor, Color? disabledTextColor,
  Color? color, Color? disabledColor, Color? focusColor, Color? hoverColor,
  Color? highlightColor, Color? splashColor, Brightness? colorBrightness,
  double? elevation, double? focusElevation, double? hoverElevation,
  double? highlightElevation, double? disabledElevation, EdgeInsetsGeometry? padding,
  VisualDensity? visualDensity, ShapeBorder? shape, Clip clipBehavior: Clip.none,
  FocusNode? focusNode, bool autofocus: false, MaterialTapTargetSize? materialTapTargetSize,
  Duration? animationDuration, double? minWidth, double? height, bool enableFeedback: true,
  Widget? child
})
```

Padding

Insira um widget a partir de um *padding*.



```
const Padding({  
  Key? key,  
  required EdgeInsetsGeometry padding,  
  Widget? child  
})
```



```
const Padding({  
  EdgeInsetsGeometry: EdgeInsets.all(12.0),  
  child: RectangleRandomColor()  
})
```

ThemeData

Faça seu tema, a partir do *MaterialApp*. (Também para um subgrupo de *widgets* dentro da aplicação)

Referências

- <https://api.flutter.dev/flutter/widgets/Expanded-class.html>
 - <https://api.flutter.dev/flutter/widgets/FittedBox-class.html>
 - <https://api.flutter.dev/flutter/material/MaterialButton-class.html>
 - <https://api.flutter.dev/flutter/widgets/Padding-class.html>
 - <https://api.flutter.dev/flutter/material/ThemeData-class.html>
-