



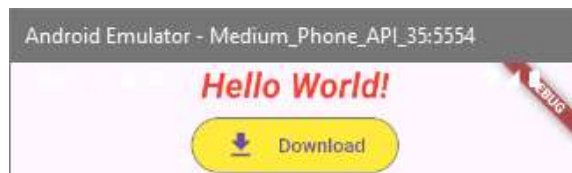
Scaffold and AppBar

- Explain what UI elements of an app are defined in the Scaffold
- Define an AppBar in your app
- Center and format its title
- Define action buttons in the AppBar
- Define and style a FloatingActionButton (fab) in your app
- Define a BottomNavigationBar in your app



Next Goal: show an AppBar on top of our app

From last lesson
(Column without
MainAxisAlignment.center):

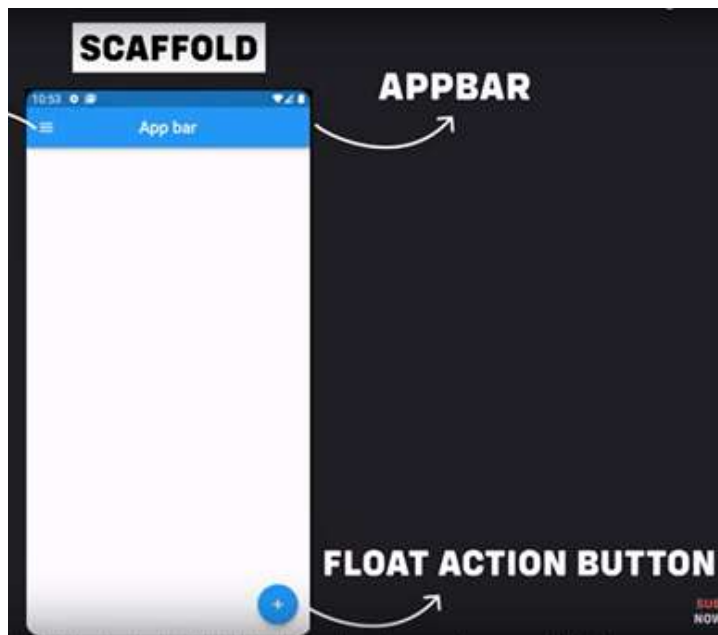


Added an AppBar
on top of the App





Children of the Scaffold (deutsch: Gerüst)



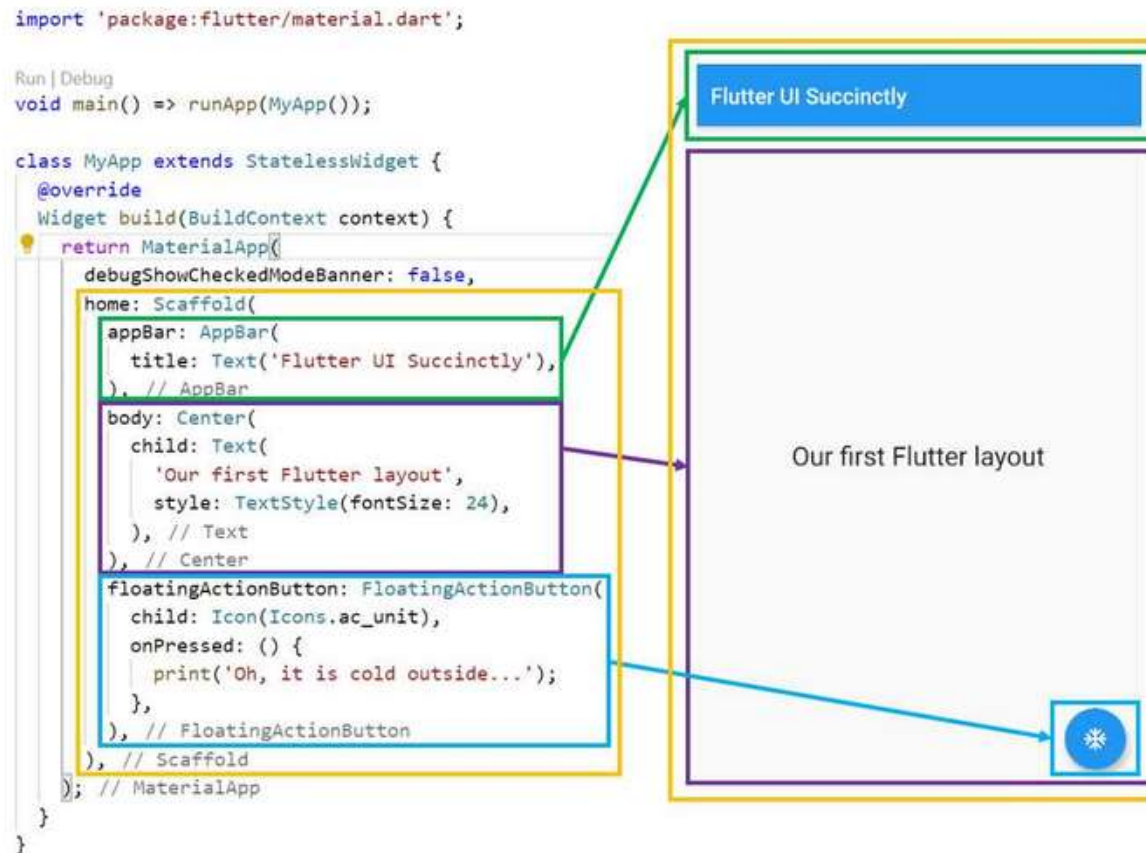
```
@override
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      body: Column(
```

```
(const) Scaffold Scaffold({
  Key? key,
  PreferredSizeWidget? appBar,
  Widget? body,
  Widget? floatingActionButton,
  FloatingActionButtonLocation? floatingActionButtonLocation,
  FloatingActionButtonAnimator? floatingActionButtonAnimator,
  List<Widget>? persistentFooterButtons,
  AlignmentDirectional persistentFooterAlignment = AlignmentDirectional.centerEnd,
  Widget? drawer,
  void Function(bool)? onDrawerChanged,
  Widget? endDrawer,
  void Function(bool)? onEndDrawerChanged,
```

Copied from <https://www.youtube.com/watch?app=desktop&v=D4nhaszNW4o> at 3:34



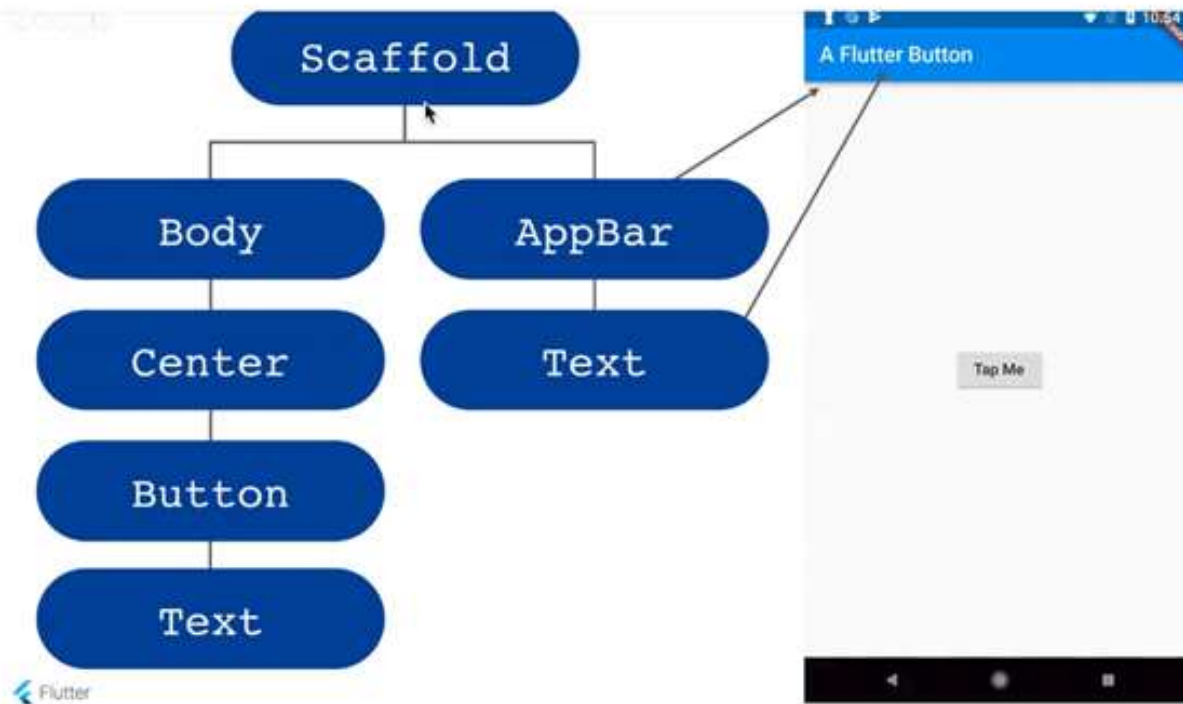
Scaffold's c-tor arguments





The widget tree below Scaffold

Scaffold widget





Exercise

Create the following AppBar for our HelloWorld App:



Hint:

You can use properties of class AppBar to show the title in the center and in white. But there are also other ways to achieve this.

Solutions:

```
appBar: AppBar(  
  title: Text(data: "Flutter Training",  
    style: TextStyle(fontSize: 22)), // Text  
  centerTitle: true,  
  foregroundColor: Colors.white,  
  backgroundColor: Colors.blue,
```

or

```
appBar: AppBar(  
  title: Center(  
    child: Text(data: "Flutter Training",  
      style: TextStyle(fontSize: 22,  
        color: Colors.white)), // TextStyle  
  ), // Center  
  backgroundColor: Colors.blue,
```

Reminder from one of the last slides:

```
return MaterialApp(  
  debugShowCheckedModeBanner: false,  
  home: Scaffold(  
    appBar: AppBar(  
      title: Text('Flutter UI Succinctly'),  
    ), // AppBar  
    body: Center(  

```



More c-tor arguments for an AppBar

```
(new) AppBar AppBar({  
  Key? key,  
  Widget? leading,  
  bool automaticallyImplyLeading = true,  
  Widget? title,  
  List<Widget>? actions,  
  Widget? flexibleSpace,  
  PreferredSizeWidget? bottom,  
  double? elevation,  
  double? scrolledUnderElevation,  
  bool Function(ScrollNotification) notificationPredicate = defaultScrollNotificationPredicate,  
  Color? shadowColor,  
  Color? surfaceTintColor,  
  ShapeBorder? shape,  
  Color? backgroundColor,  
  Color? foregroundColor,  
  IconThemeData? iconTheme,  
  IconThemeData? actionsIconTheme,  
  bool primary = true,  
  bool? centerTitle,  
  bool excludeHeaderSemantics = false,  
  double? titleSpacing,  
  double toolbarOpacity = 1.0,  
  double bottomOpacity = 1.0,  
})
```



leading and actions

```
home: Scaffold(  
  appBar: AppBar(  
    title: Text("Flutter Training", style: TextStyle(fontSize: 22)),  
    backgroundColor: Colors.blue,  
    foregroundColor: Colors.white,  
    centerTitle: true,  
    actions: [  
      IconButton(  
        onPressed: () {  
          print("settings button pressed");  
        },  
        icon: Icon(Icons.settings)) // IconButton  
    ],  
    leading: Icon(Icons.construction)), // AppBar
```

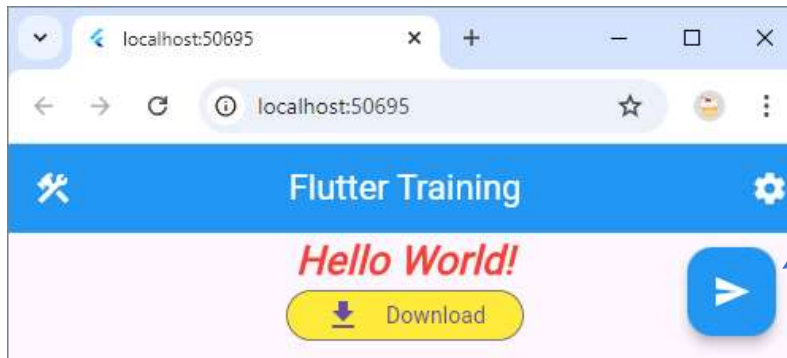


Exercise:

Implement the same or something similar by your own.
Try to define 2 children in the list behind “actions:”



Define a Floating Action Button



```
appBar: AppBar( // AppBar ...  
  floatingActionButton: FloatingActionButton(  
    foregroundColor: Colors.white,  
    backgroundColor: Colors.blue,  
    onPressed: () {},  
    child: Icon(Icons.send),  
  ), // FloatingActionButton
```

Exercise:

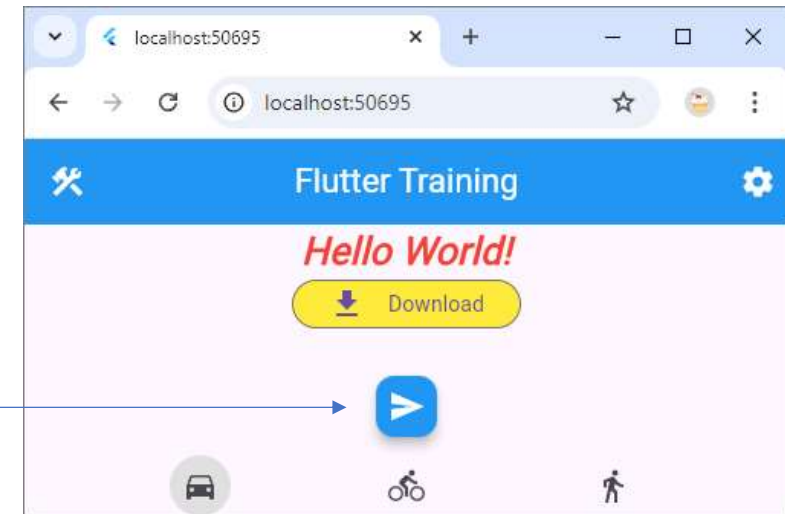
Define a FloatingActionButton in your app and test it on Chrome or Edge. Observe what happens when you resize the browser window.

Define and test the properties “**splashColor**” and “**hoverColor**” of a FloatingActionButton. Finally use the property “**mini**” of a fab (short for FloatingActionButton)



FloatingActionButtonLocation and BottomNavigationBar

```
floatingActionButton: FloatingActionButton(  
  mini: true,  
  foregroundColor: Colors.white,  
  backgroundColor: Colors.blue,  
  hoverColor: Colors.red,  
  splashColor: Colors.green,  
  onPressed: () {},  
  child: Icon(Icons.send),  
), // FloatingActionButton  
floatingActionButtonLocation: FloatingActionButtonLocation.miniCenterFloat,  
bottomNavigationBar: Row(  
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
  children: [  
    IconButton(  
      style:  
        IconButton.styleFrom(backgroundColor: Colors.grey.shade300),  
      onPressed: () {},  
      icon: Icon(Icons.directions_car)], // IconButton  
    IconButton(onPressed: () {}, icon: Icon(Icons.directions_bike)),  
    IconButton(onPressed: () {}, icon: Icon(Icons.directions_walk)),  
  ],  
), // Row
```



- centerDocked Float
- centerFloat Float
- centerTop Floating
- endContained Float
- endDocked Floating
- endFloat Floating
- endTop FloatingAc
- startDocked Float
- startFloat Float
- startTop Floating

- miniCenterDocked
- miniCenterFloat F
- miniCenterTop Flo
- miniEndDocked Flo
- miniEndFloat Floa
- miniEndTop Floati
- miniStartDocked F
- miniStartFloat Fl
- miniStartTop Floa

Possible goal in the next weeks

Define a Scaffold that
looks like WhatsApp:

