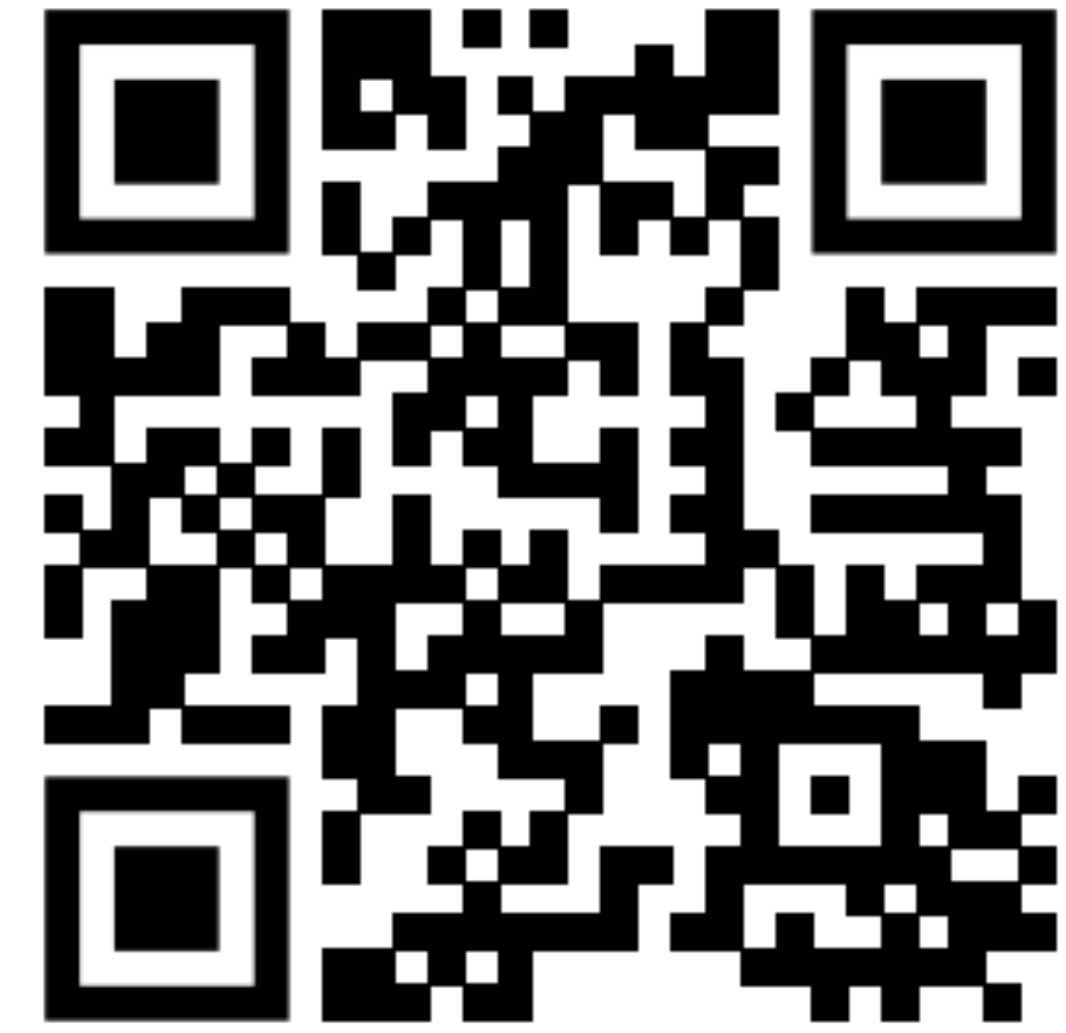




Виджеты. “Три кита Flutter’a”

Павел Хотин, Flutter разработчик

Окей, летс гоу



<https://github.com/Paulik8/lessons>

Зачем нам это всё

- › UI во Flutter'е и близко не похож на вёрстку в корневой мобильной разработке (iOS/Android)
- › Виджеты бывают разных типов, и понимать их возможности - критически важно
- › Только понимая, как устроен UI под капотом, получится делать сложные интерфейсы
- › Flutter из коробки имеет очень удобные инструменты для работы с UI и данными, важно уметь их правильно использовать

О чём поговорим

00 | Что такое виджет

01 | StatelessWidget

02 | StatefulWidget

03 | InheritedWidget

00

Что такое виджет

Что такое виджет

Get started

Samples & tutorials

Development

▼ User interface

Introduction to Widgets

▶ Building layouts

Adding interactivity

Assets and images

Navigation & routing

▶ Animations

▶ Advanced UI

Widget catalog

▶ Data & backend

▶ Accessibility & internationalization

▶ Platform integration

▶ Packages & plugins

▶ Tools & techniques

Widget catalog

[Docs](#) > [Development](#) > [UI](#) > Widgets

Create beautiful apps faster with Flutter’s collection of visual, structural, platform, and interactive widgets. In addition to browsing widgets by category, you can also see all the widgets in the [widget index](#).

Accessibility

Make your app accessible.

Visit

Animation and Motion

Bring animations to your app.

Visit

Assets, Images, and Icons

Manage assets, display images, and show icons.

Visit

Async

Async patterns to your Flutter application.

Basics

Widgets you absolutely need to know before building your first

Cupertino (iOS-style widgets)

Beautiful and high-fidelity widgets

Что такое виджет



Что такое виджет

Неизменяемая конфигурация

```
@immutable  
abstract class Widget extends DiagnosticableTree {  
  /// Initializes [key] for subclasses.  
  const Widget({ this.key });
```

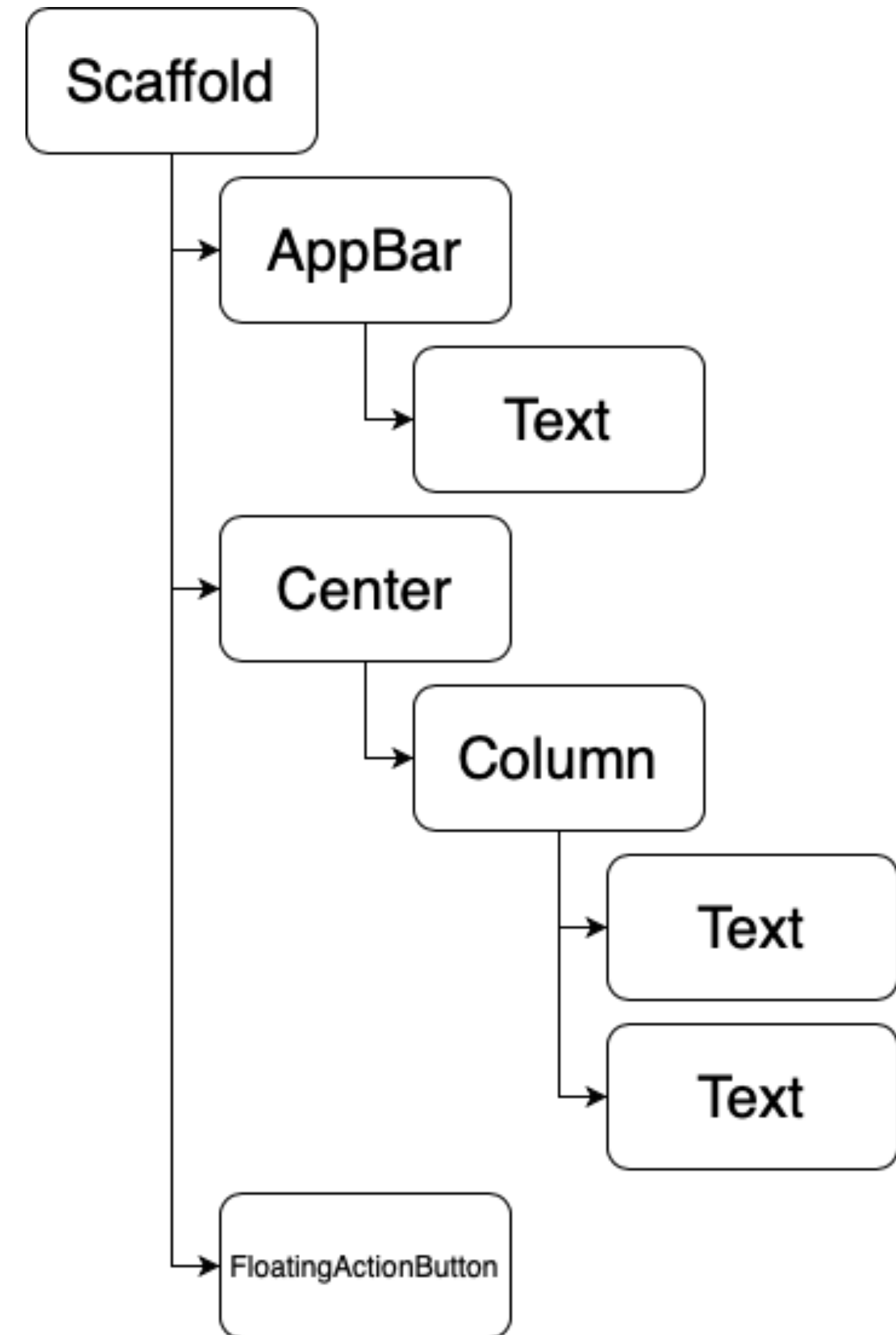

Что такое виджет

Древовидная,
иерархическая
структура

```
@override
Widget build(BuildContext context) => Scaffold(
  appBar: AppBar(
    title: Text(title),
  ), // AppBar
  body: Center(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        const Text('Tapped count: '),
        Text('$_counter'),
      ],
    ), // Column
  ), // Center
); // Scaffold
```

Что такое виджет

Древовидная,
иерархическая
структура



Что такое виджет

При помощи виджетов описывается интерфейс (UI) Flutter-приложения

Виджеты — высокоуровневые сущности, с которыми мы работаем подавляющую часть времени

Виджеты — глупые. А вот то, что с ними происходит под капотом — поумнее

01

StatelessWidget

StatelessWidget

- | Та самая легковесная конфигурация
- | Описывает UI, наполняет его данными
- | Не управляет состоянием, может быть перерисован только “по команде” родителя

ОСНОВНЫЕ ВИДЖЕТЫ

Center

Text

SizedBox

Padding

ОСНОВНЫЕ ВИДЖЕТЫ

Container

AnimatedContainer

Дополнительное поле - duration

Основные виджеты

Row

Column

Flex

Основные поля:

- `mainAxisAlignment`
- `crossAxisAlignment`
- `mainAxisSize`



<https://medium.com/flutter-community/flutter-layout-cheat-sheet-5363348d037e>

ОСНОВНЫЕ ВИДЖЕТЫ

Expanded

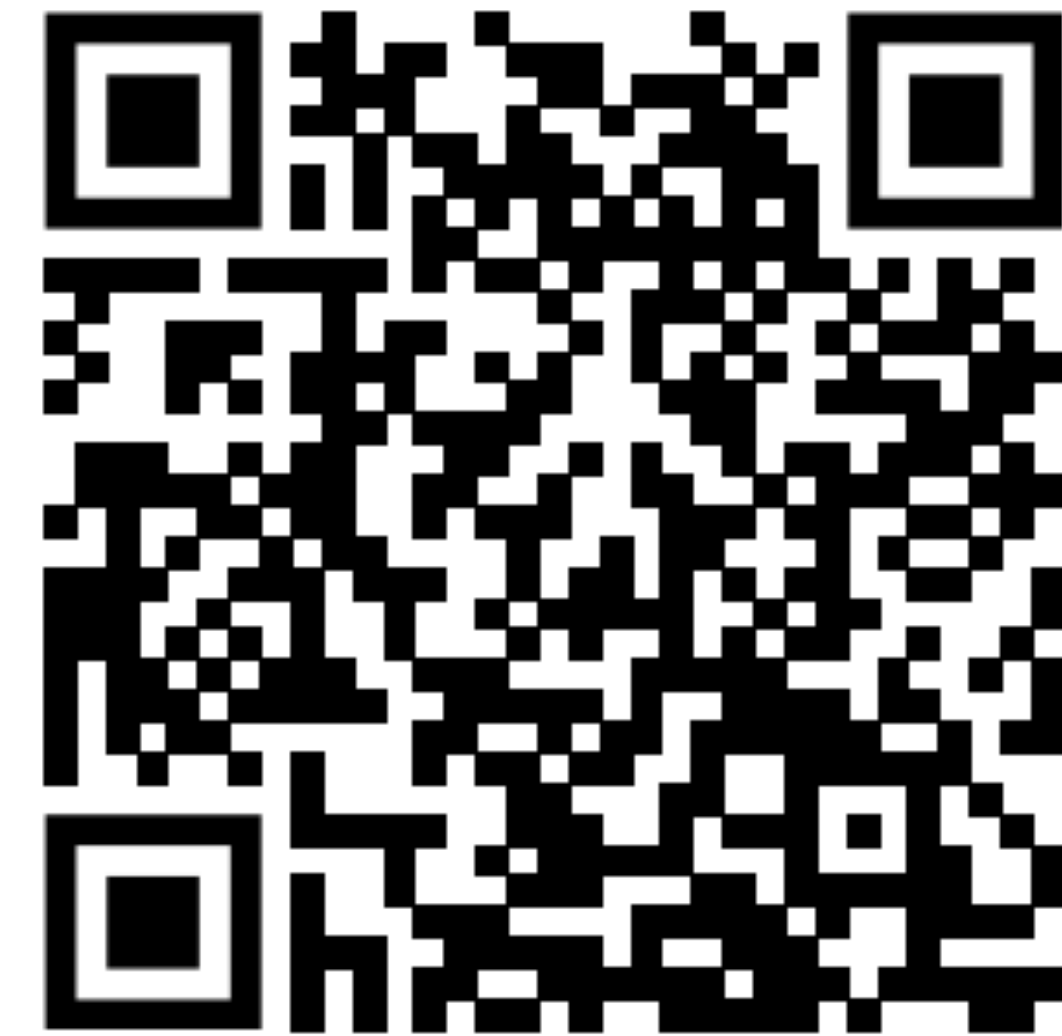
Flexible

Spacer

Основное поле - flex

Stack

Основное поле - alignment



<https://docs.flutter.dev/development/ui/layout/constraints>

02

StatefulWidget

StatefulWidget

- Сам по себе виджет — всё ещё легковесная конфигурация, иммутабельная, ничего не может
- State* — тяжеловес, может мутировать. Это отдельный класс, экземпляр которого НЕ пересоздаётся при перерисовках
- Из стейта может управлять перерисовками при помощи *setState*
- У *State* есть собственный жизненный цикл, на его события можно реагировать

State Lifecycle

initState — единожды при инициализации

didChangeDependencies — единожды после инициализации и далее при уведомлениях от виджетов вверху по дереву, от которых зависим

didUpdateWidget — каждый раз при обновлении конфигурации виджета

setState — вызывается для перерисовки

build — каждый раз при перерисовке

dispose — при удалении из дерева

Посмотрим в коде

03

InheritedWidget

InheritedWidget

- Предоставляет данные, ничего не рисует на экране

- Решает проблему передачи данных через конструкторы — сильно расцепляет и упрощает код

- Может быть доступен для виджетов внизу по дереву через контекст при обращении методом
`dependOnInheritedWidgetOfExactType<T>(BuildContext context)`

- Может уведомлять виджеты внизу по дереву об изменениях в собственных данных, тем самым провоцируя их перерисовку

Посмотрим в коде

Требования к проектам

Обязательно:

- Список

- Несколько экранов с навигацией 1.0

- Ввод текста

- Какие-то анимации

- В приложении не должно быть сильных просадок fps

Хорошо, если есть:

- Темная тема



Q&A

Павел Хотин



khotinp@yandex-team.ru



@Paulik8