

MOBILE  
PEOPLE  
TALKS

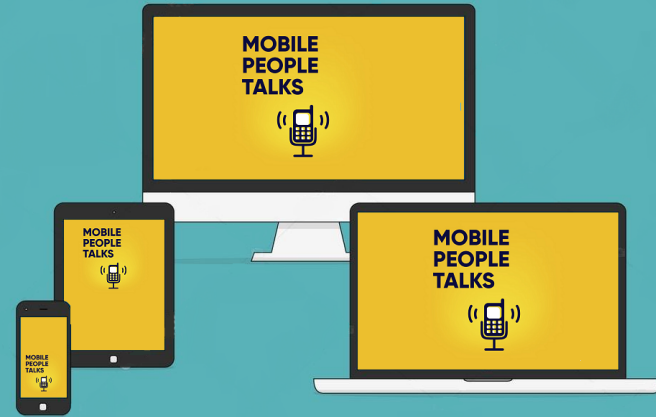


# Multiplatform Flutter

## Building Adaptive UI



Aleksandr Denisov  
Senior Software Engineer - EPAM  
Organizer - GDG Nizhny Novgorod



# Agenda

---

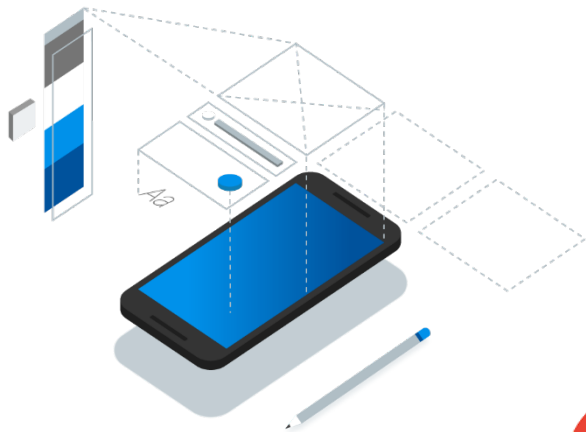
1. Flutter overview
2. Adaptive UI
3. Solutions provided by Flutter
  - a. iOS and Android
  - b. Screen orientation
  - c. Passive and Active UI
  - d. Different screen sizes and resolutions
4. Future of the Flutter



# Flutter

# What is Flutter?

---



# Why Dart?

---



# Dart



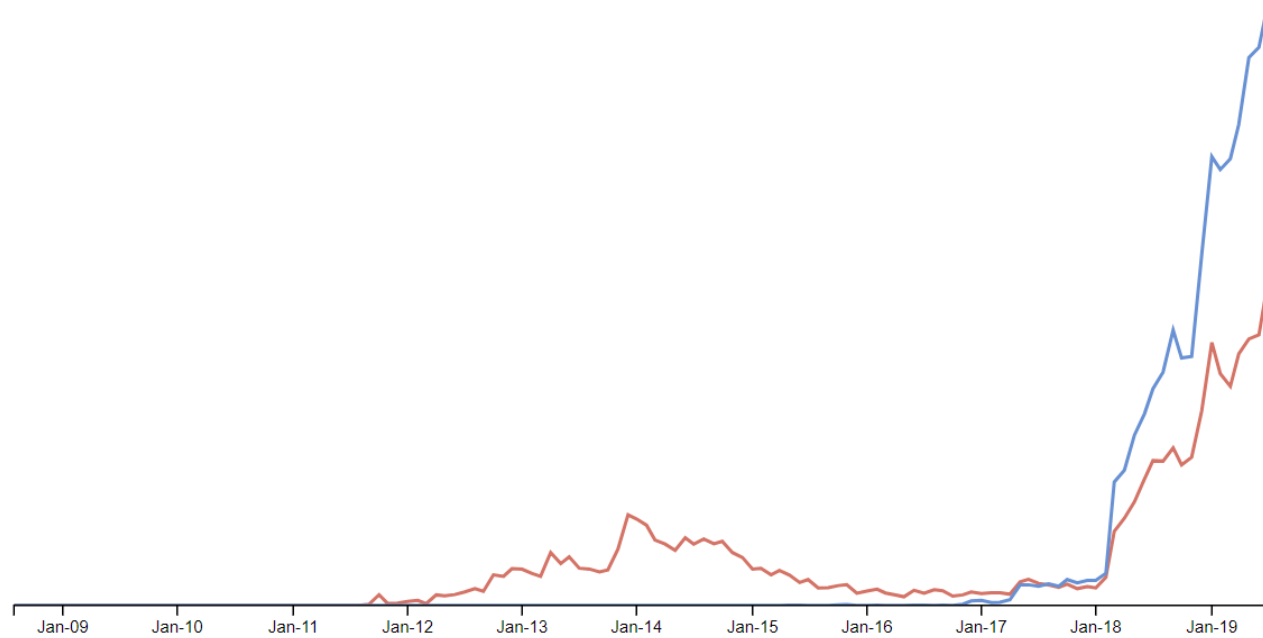


# Flutter and Dart popularity

dart × flutter ×

Questions over time ?

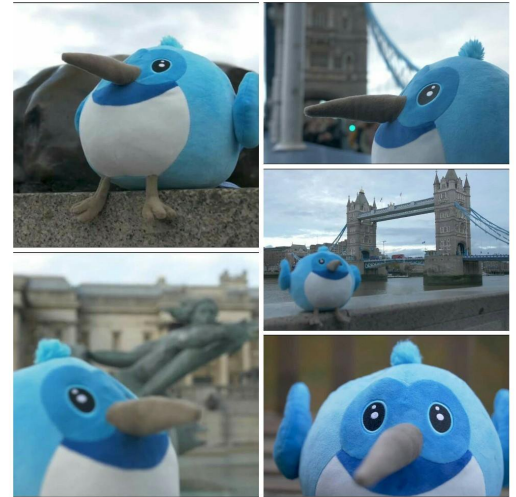
Relative:



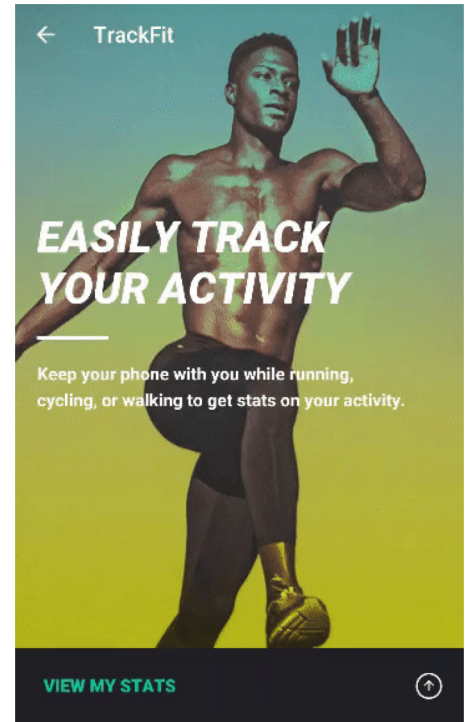
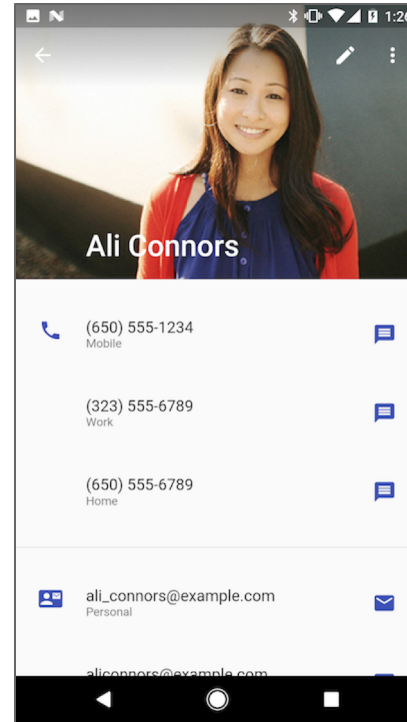
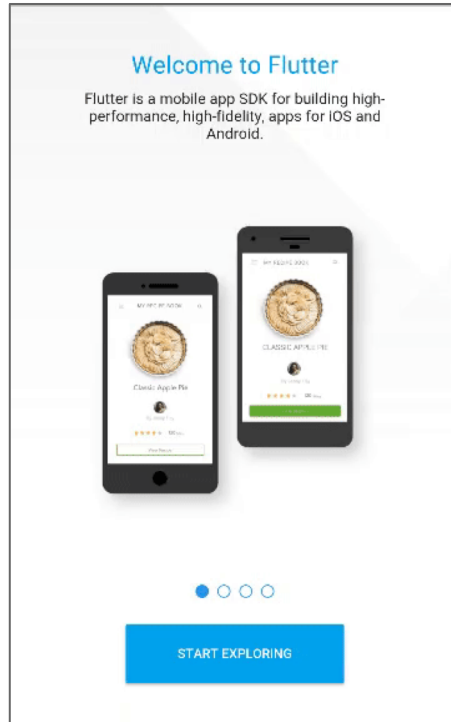
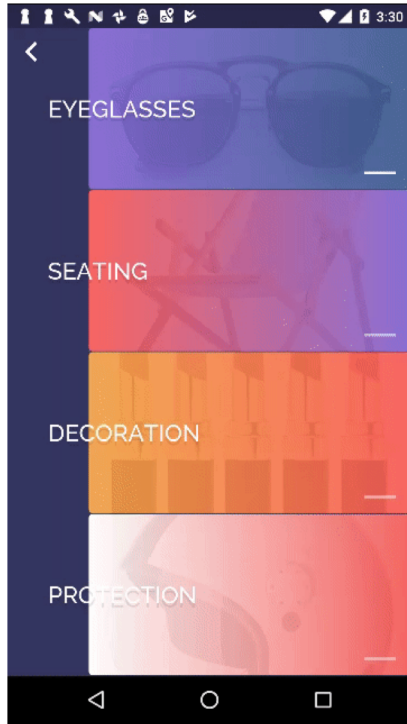
# Why Dart?

- Dart is AOT (Ahead Of Time) compiled to fast, predictable, native code (super fast startup and execution)
- Dart is JIT (Just In Time) compiled for exceptionally fast development cycles (super fast hot reload)
- A good garbage collector to clean up after creating and destroying many objects
- Single threaded to avoid locks and therefore jank
- Dart is particularly easy to learn
- Dart 2.x (currently 2.4) has a lot of nice features

Dart has a mascot, named Dash



# Great looking examples from Flutter Gallery





**FLUTTER**

YOU

~~ME~~

~~NATIVE~~

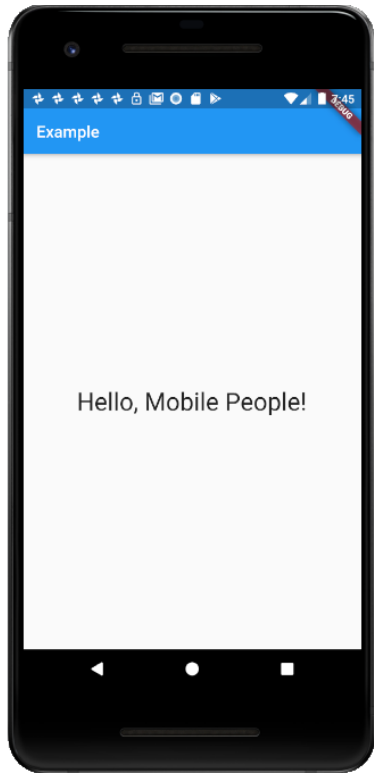
ANYTHING

# Flutter example

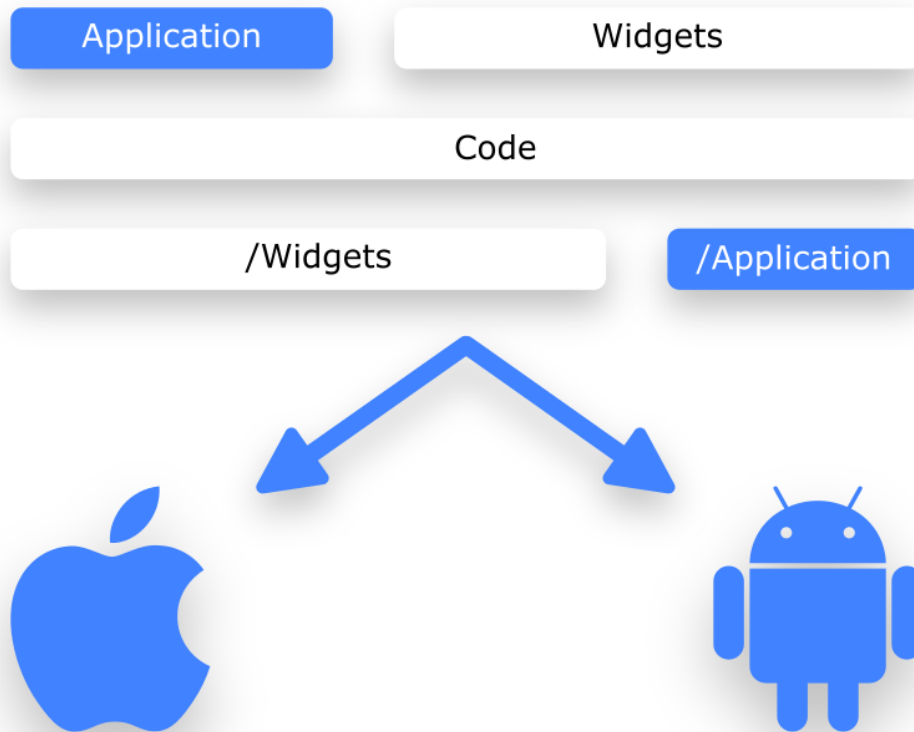
```
void main() => runApp(HelloFlutter());

class HelloFlutter extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter example',
      home: Scaffold(
        appBar: AppBar(
          title: Text('Example'),
        ),
        body: Container(
          child: Center(
            child: Text('Hello, Mobile People!'),
          ),
        ),
      ),
    );
  }
}
```

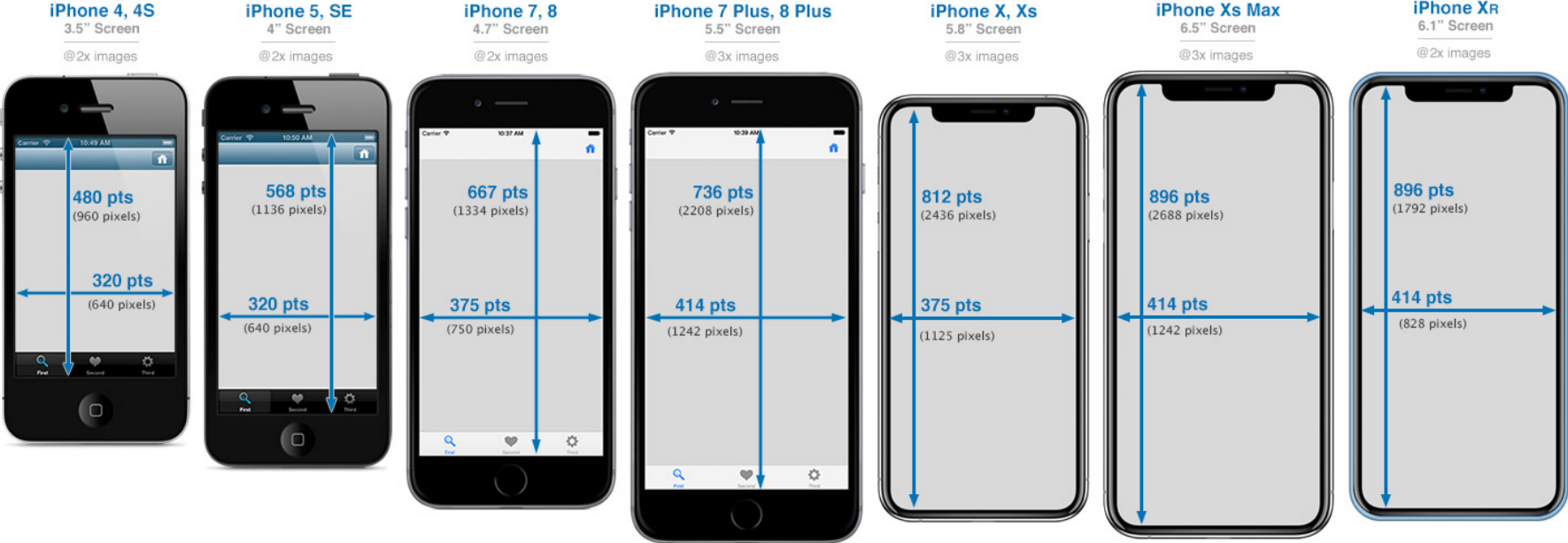


# Single Codebase



Картинку  
перерисовать

# iOS resolutions



# Android resolutions

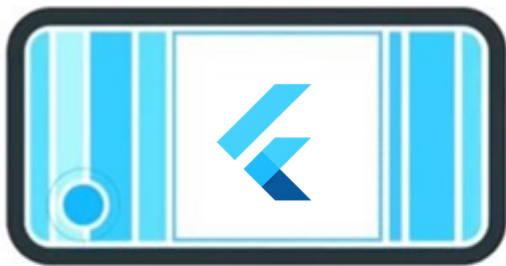
---



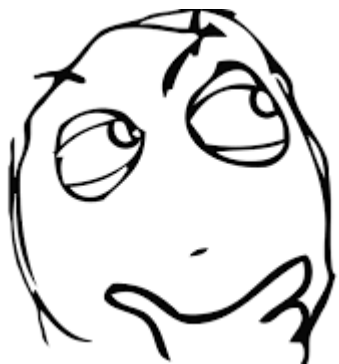


# Portrait/Landscape

---

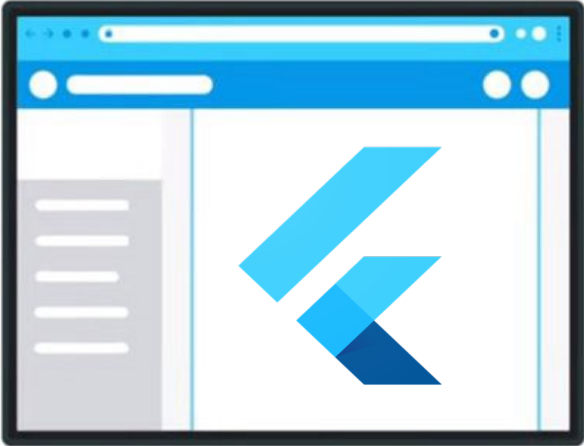


© 2014 EPAM



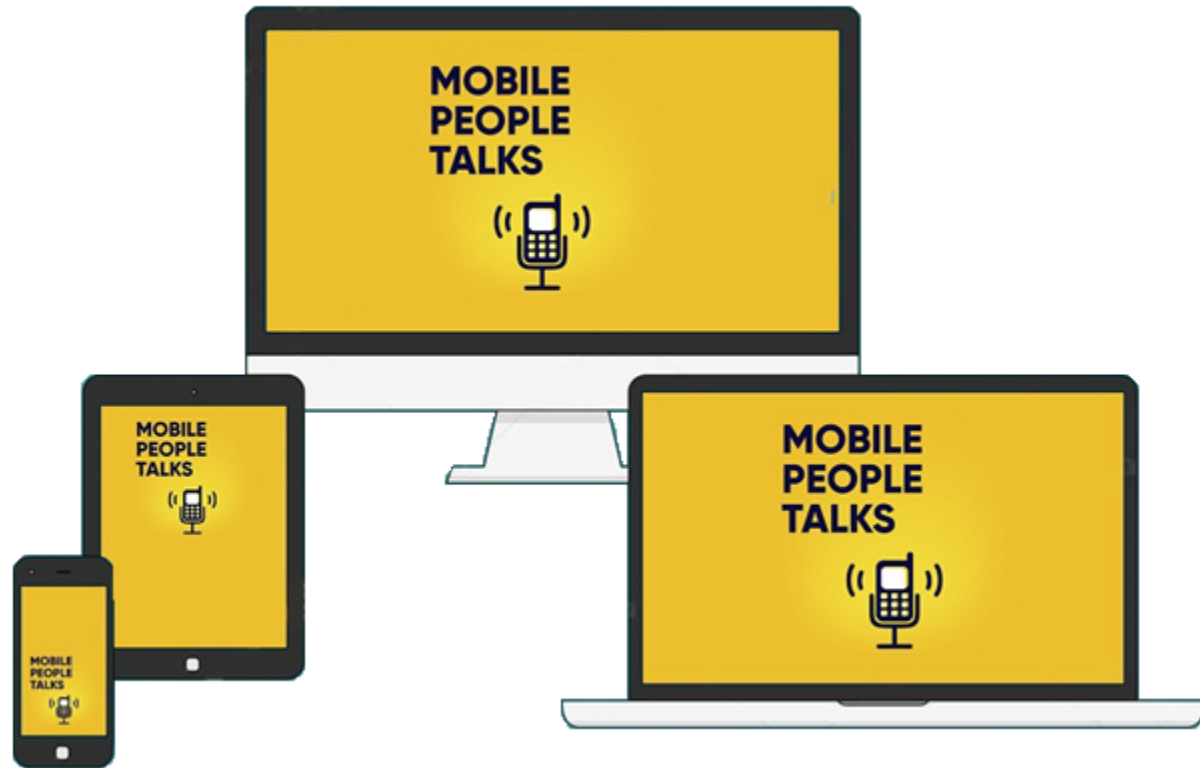
# Not mobile only

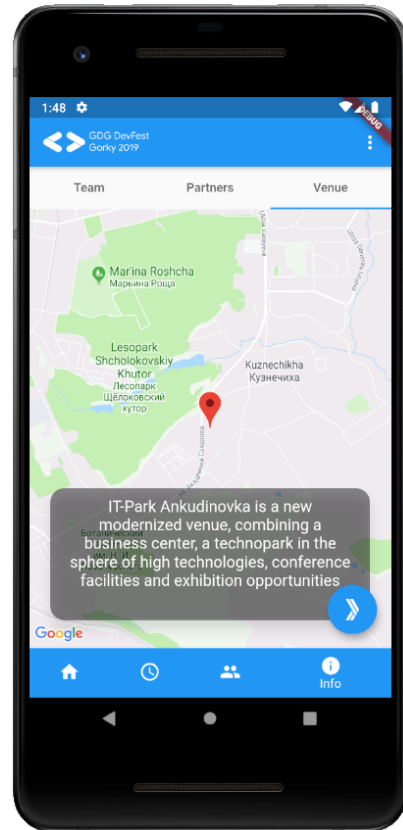
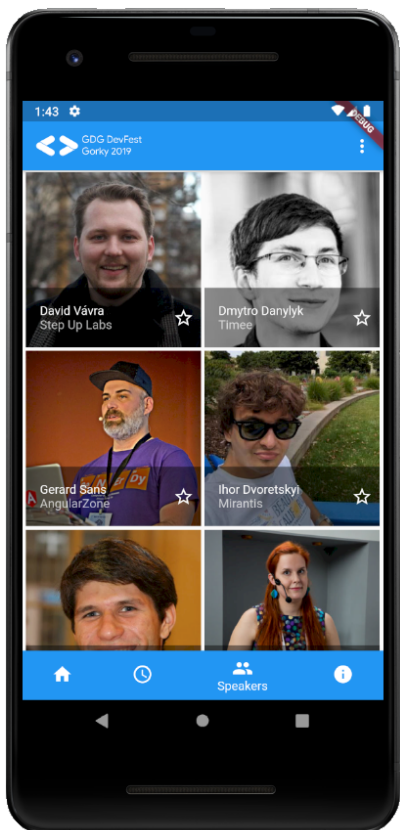
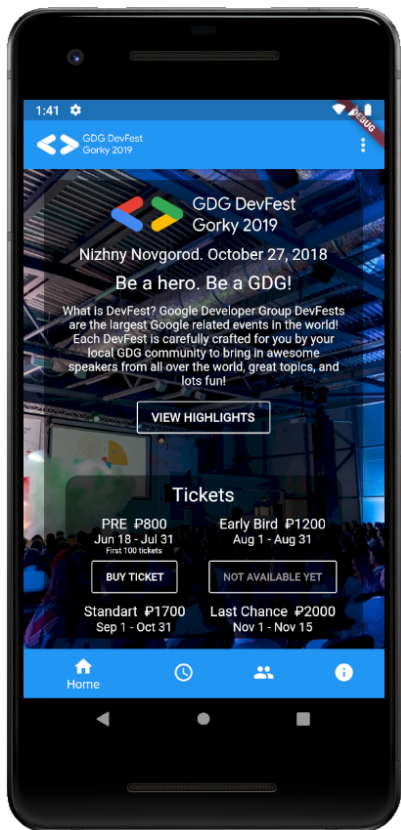
---



# Adaptive UI

---







GDG DevFest  
Gorky 2019

Nizhny Novgorod, October 27, 2018

Be a hero. Be a GDG!

What is DevFest? Google Developer Group DevFests are the largest Google related events in the world! Each DevFest is carefully crafted for you by your local GDG community to bring in awesome speakers from all over the world, great topics, and lots fun!

[VIEW HIGHLIGHTS](#)

## Tickets

Students ₺500

Jun 17 - Oct 26

Requires valid student ID

[BUY TICKET](#)

Early Bird ₺750

Aug 1 - Aug 31

[BUY TICKET](#)

Pre-Early Bird ₺500

Jun 17 - Jul 31

[BUY TICKET](#)

Regular ₺1000

Sep 1 - Oct 21

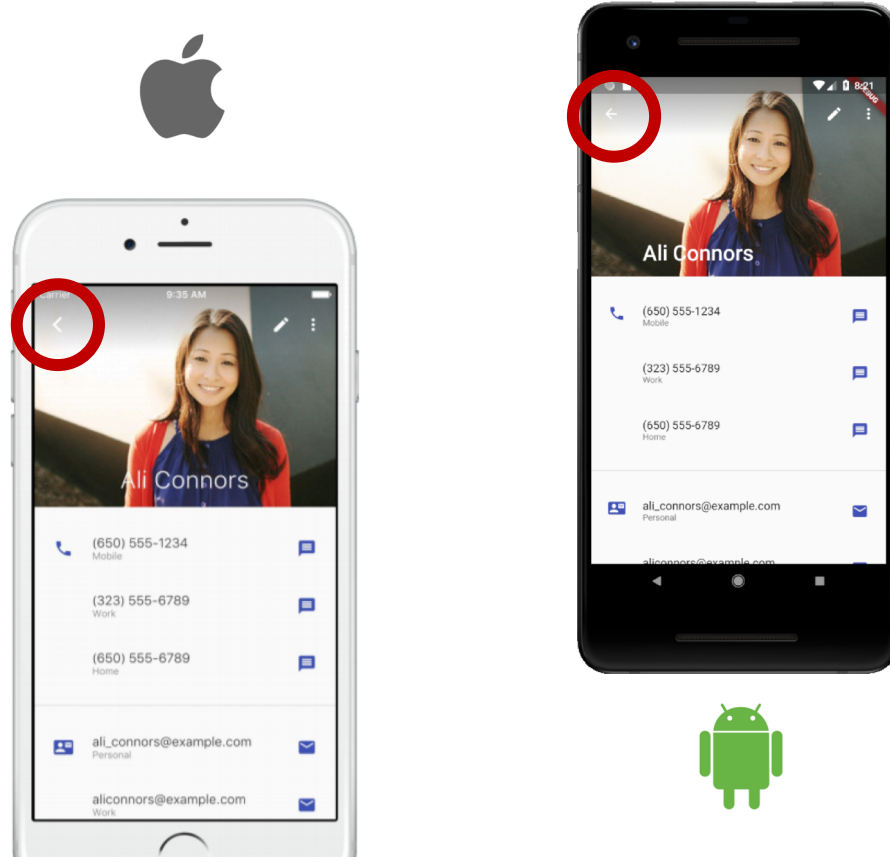
[BUY TICKET](#)



Home



# IOS and Android

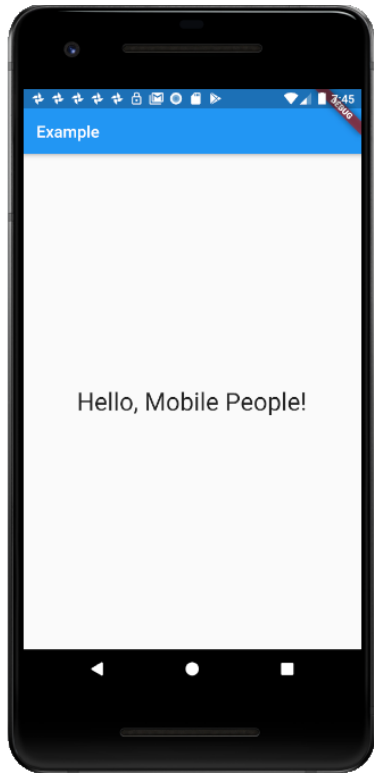


# Flutter example

```
void main() => runApp(HelloFlutter());


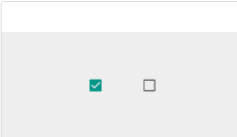
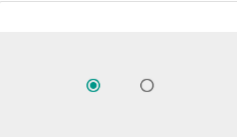
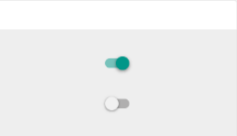


class HelloFlutter extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter example',
      home: Scaffold(
        appBar: AppBar(
          title: Text('Example'),
        ),
        body: Container(
          child: Center(
            child: Text('Hello, Mobile People!'),
          ),
        ),
      ),
    );
  }
}
```





# Material Components widgets

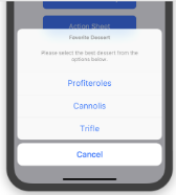
 <p><b>TextField</b></p> <p>Touching a text field places the cursor and displays the keyboard. The TextField widget implements this component.</p> <p><a href="#">Documentation</a></p>	 <p><b>Checkbox</b></p> <p>Checkboxes allow the user to select multiple options from a set. The Checkbox widget implements this component.</p> <p><a href="#">Documentation</a></p>	 <p><b>Radio</b></p> <p>Radio buttons allow the user to select one option from a set. Use radio buttons for exclusive selection if you think that the user needs to see all available options side-by-side.</p> <p><a href="#">Documentation</a></p>
 <p><b>Switch</b></p> <p>On/off switches toggle the state of a single settings option. The Switch widget implements this component.</p> <p><a href="#">Documentation</a></p>	 <p><b>Slider</b></p> <p>Sliders let users select from a range of values by moving the slider thumb.</p> <p><a href="#">Documentation</a></p>	 <p><b>Date &amp; Time Pickers</b></p> <p>Date pickers use a dialog window to select a single date on mobile. Time pickers use a dialog to select a single time (in the hours:minutes format) on mobile.</p> <p><a href="#">Documentation</a></p>



<https://flutter.dev/docs/development/ui/widgets/material>



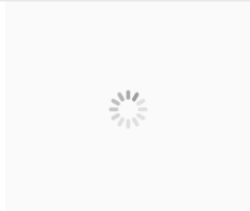
# Cupertino (iOS-style widgets)



## CupertinoActionSheet

An iOS-style modal bottom action sheet to choose an option among many.

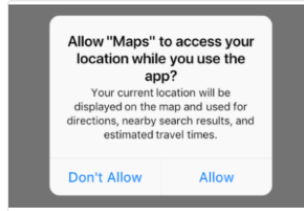
[Documentation](#)



## CupertinoActivityIndicator

An iOS-style activity indicator. Displays a circular 'spinner'.

[Documentation](#)



## CupertinoAlertDialog

An iOS-style alert dialog.

[Documentation](#)



## CupertinoButton

An iOS-style button.

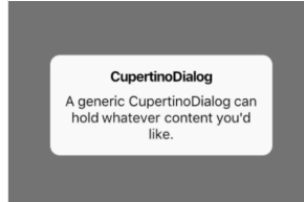
[Documentation](#)



## CupertinoDatePicker

An iOS-style date or date and time picker.

[Documentation](#)



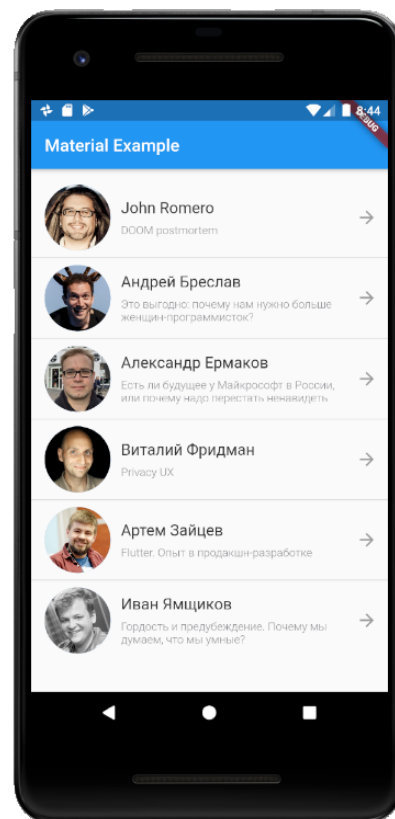
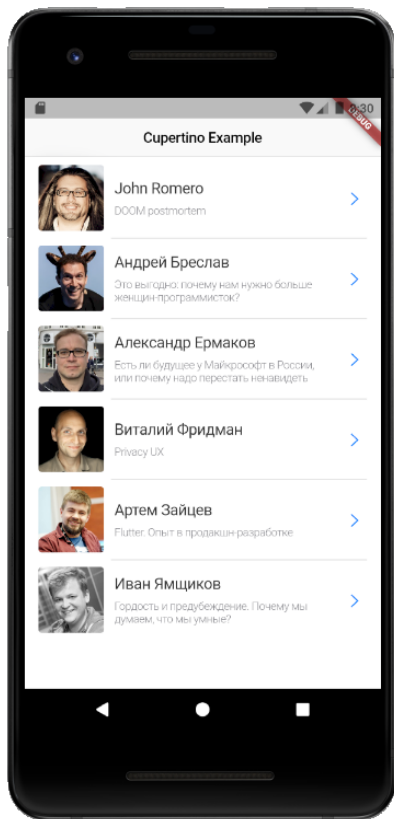
## CupertinoDialog

An iOS-style dialog.

[Documentation](#)

<https://flutter.dev/docs/development/ui/widgets/cupertino>

# Material and Cupertino



# Material and Cupertino

---

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    if (Platform.isIOS) {  
      return CupertinoApp(  
        home: CupertinoHomePage(),  
      );  
    } else  
      return MaterialApp(  
        home: MaterialHomePage(),  
      );  
    }  
  }  
}
```

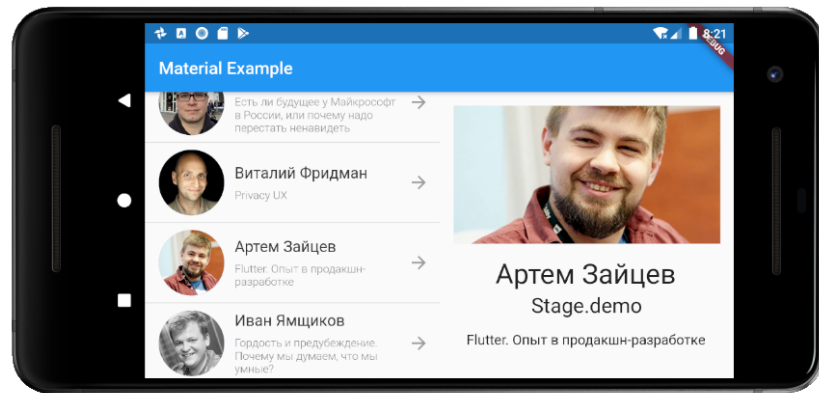
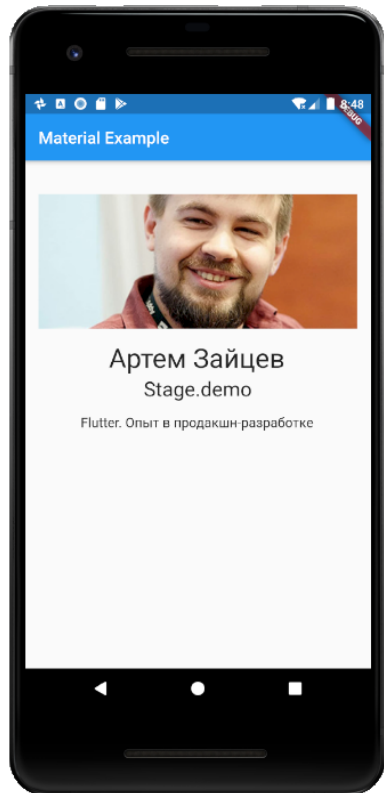
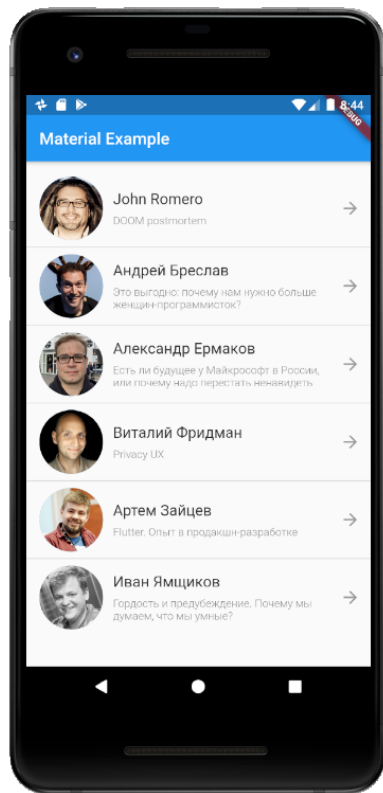
## Switch.adaptive()



## Slider.adaptive()



# Screen Orientation



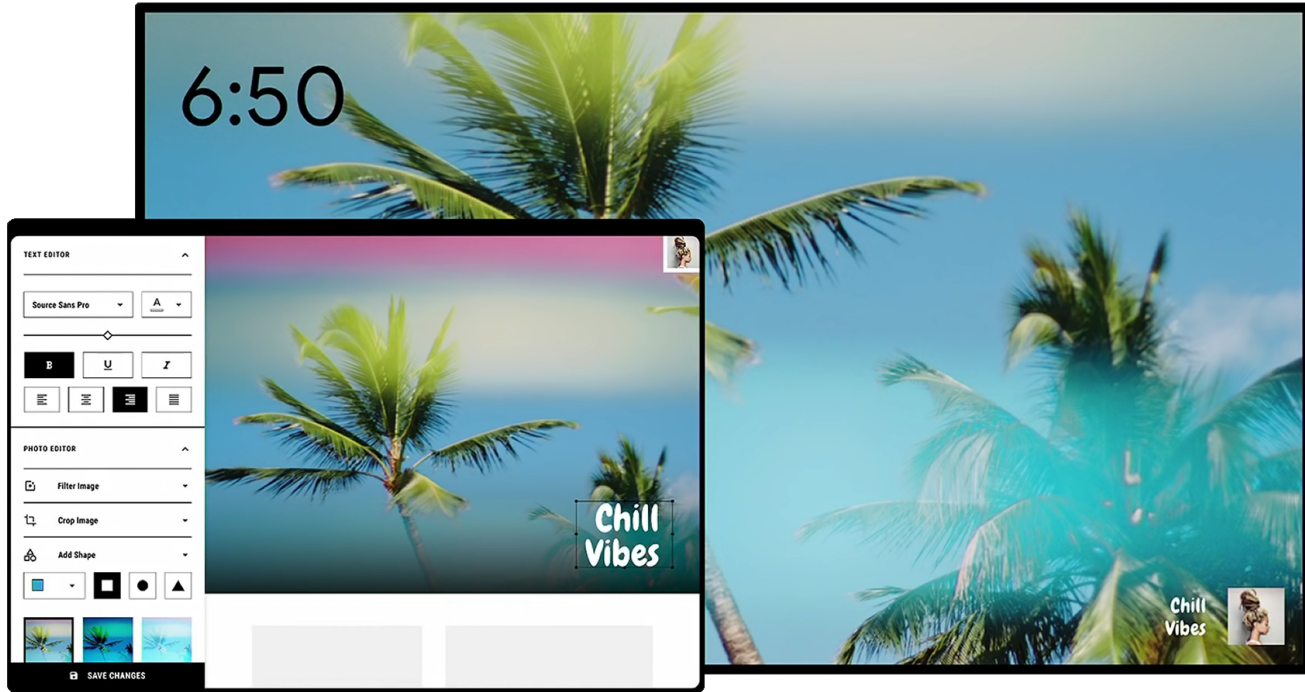
# Screen Orientation

---

```
OrientationBuilder(builder: (context, orientation) {  
  if (orientation == Orientation.portrait) {  
    return MaterialSpeakersList(  
      (speaker)=>Navigator.push(context, SpeakerDeetailsRoute, speaker))  
  } else {  
    return Row(  
      children: <Widget>[  
        MaterialSpeakersList((speaker)=>setState(()=>speaker = s)),  
        SpeakerDetails(speaker ?? speakers[0])  
      ],  
    );  
  }  
}
```

# Passive and Active UI

---



# How to understand, what is the platform?

---

```
if (Platform.is) {
```

f	isWindows	bool
f	isAndroid	bool
f	isIOS	bool
f	isFuchsia	bool
f	isLinux	bool
f	isMacOS	bool

[https://pub.dev/  
device\\_info](https://pub.dev/device_info) 0.4.0+2

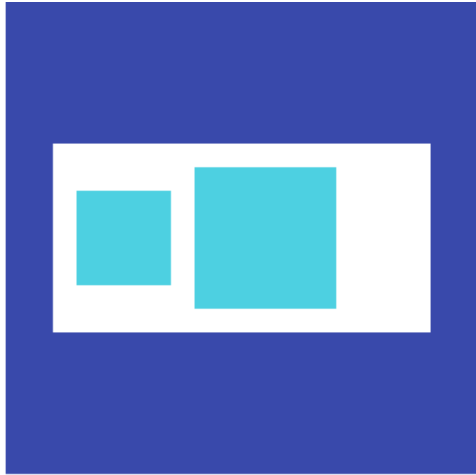


# Adaptive components

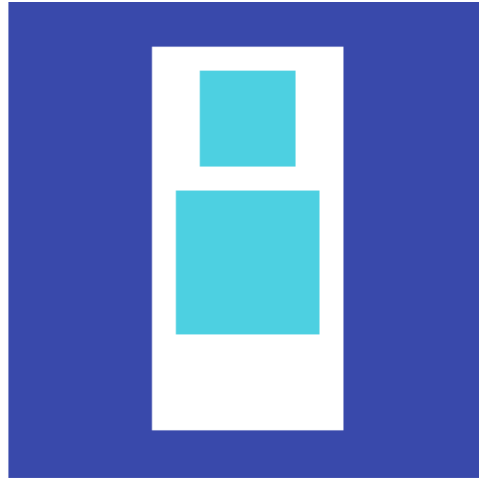
- Design concept
- Avoid layout code complexity
  - Good: Just a few parameters
  - Not so good: Structural layout differences
- Making widgets is easy

# Layout Widgets

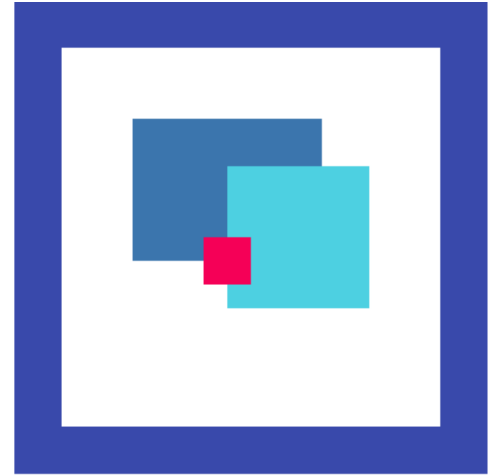
---



Row

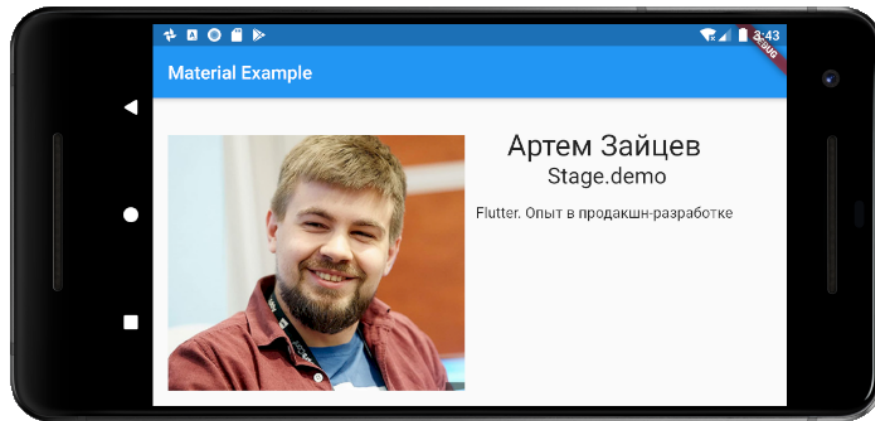
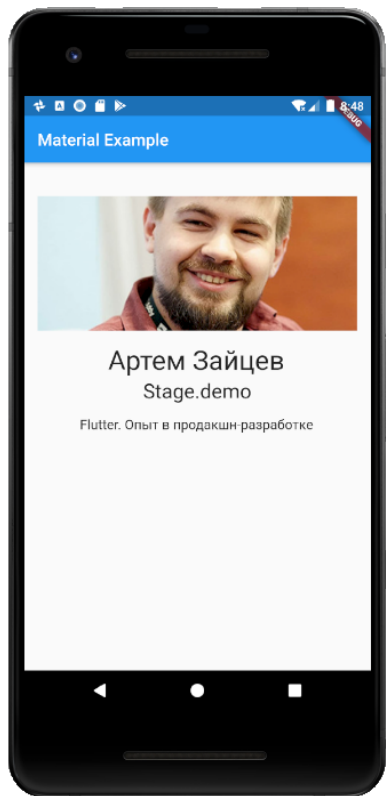


Column



Stack

# Row and Column

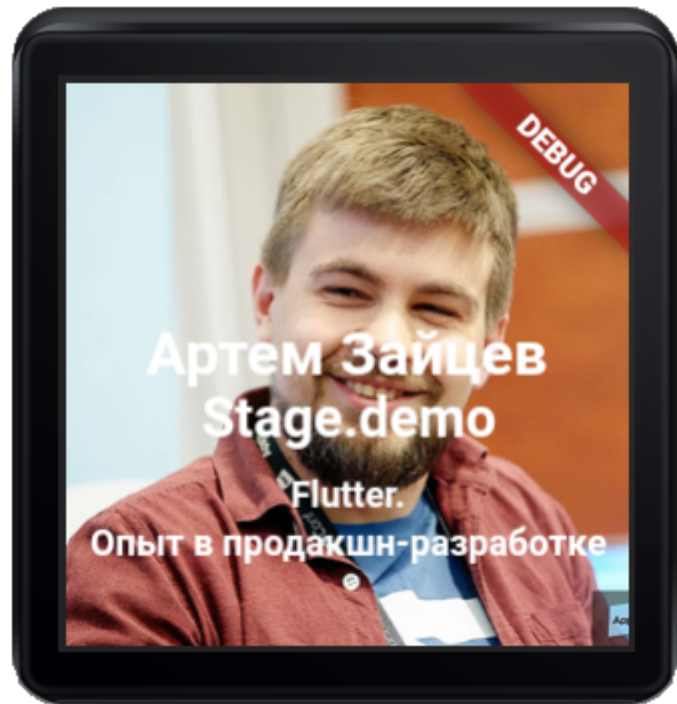


## Row or Column?

---

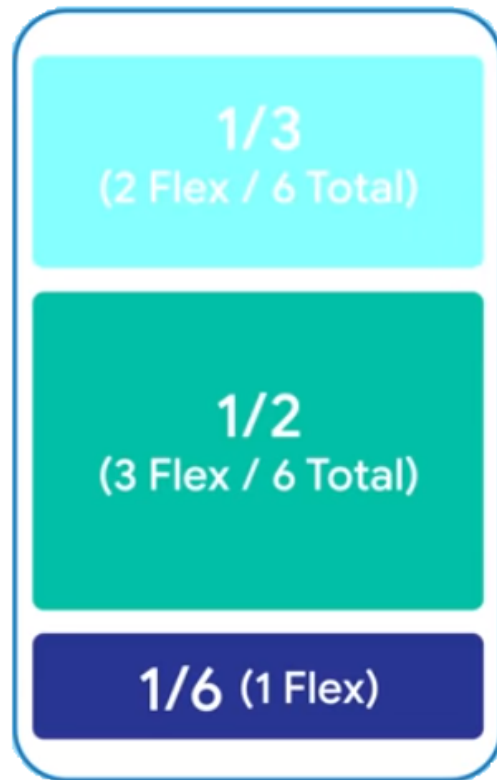
### MediaQuery.of(context)

- size
  - height
  - width
- orientation
- devicePixelRatio
- textScaleFactor



# Flexible

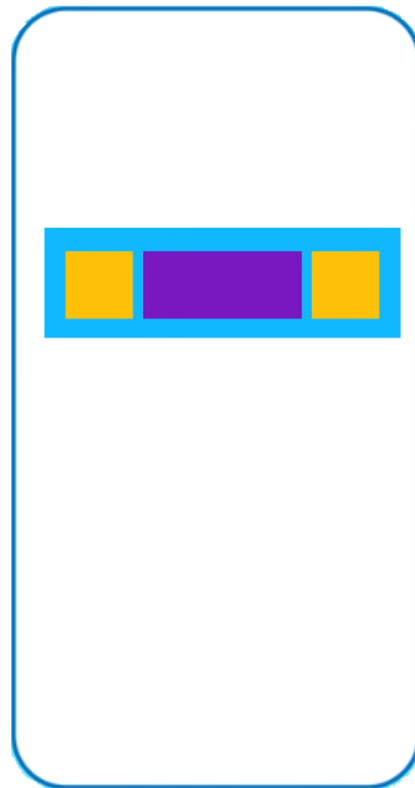
```
Column(  
  children: <Widget>[  
    Flexible(flex: 2,  
      fit: FlexFit.tight,  
      child: Container(  
        height: 100, color: Colors.cyan,  
      ),  
    ),  
    Flexible(flex: 3,  
      child: Container(  
        color: Colors.teal,  
      ),  
    ),  
    Flexible(flex: 1,  
      child: Container(  
        color: Colors.indigo,  
      ),  
    ),  
  ],  
)
```

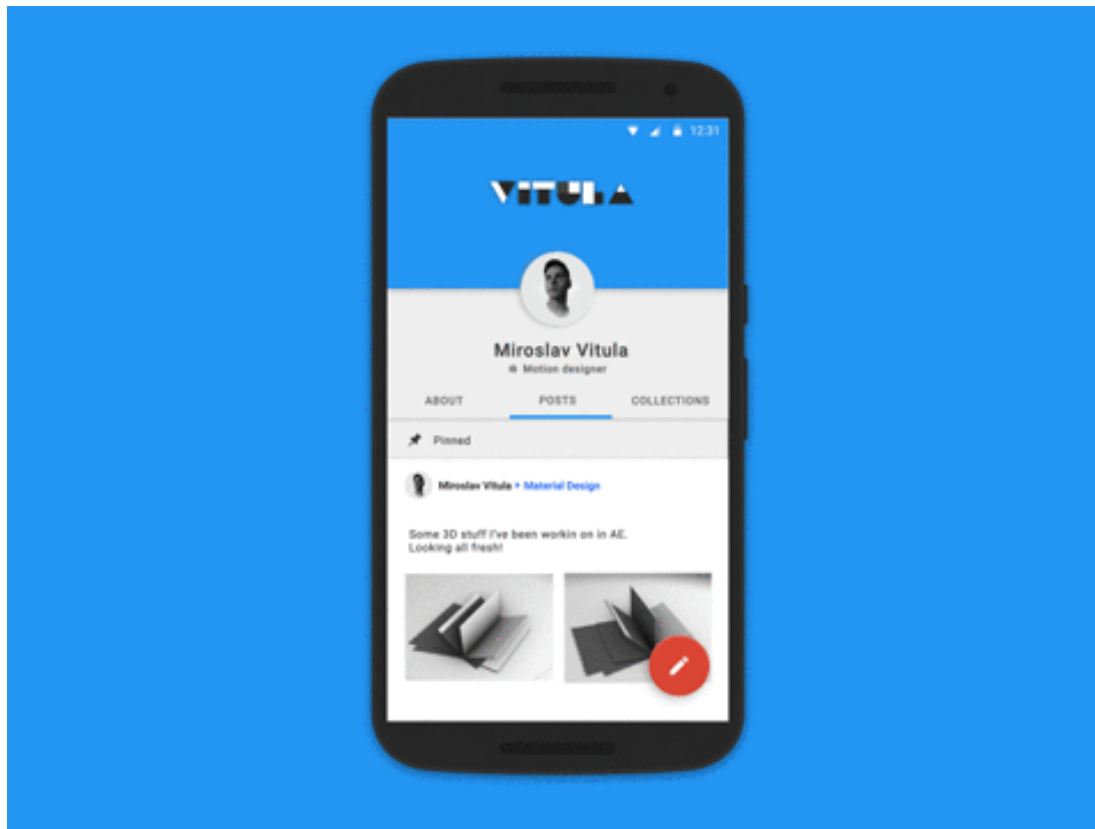


# Expanded

---

```
Row(children: <Widget> [
  MyWidget(),
  Expanded(
    child: MyWidget(),
  ),
  MyWidget()
])
```





# Why Flutter?

---

- It's **easy to start** for any kind of developer
- **Single codebase for all of platforms with rich set of widgets OOB**
- The best tooling on familiar IDE (Android Studio, VS Code etc)
- Animations implementation is much easier.
- APIs for unit and integration tests
- Great documentation and support





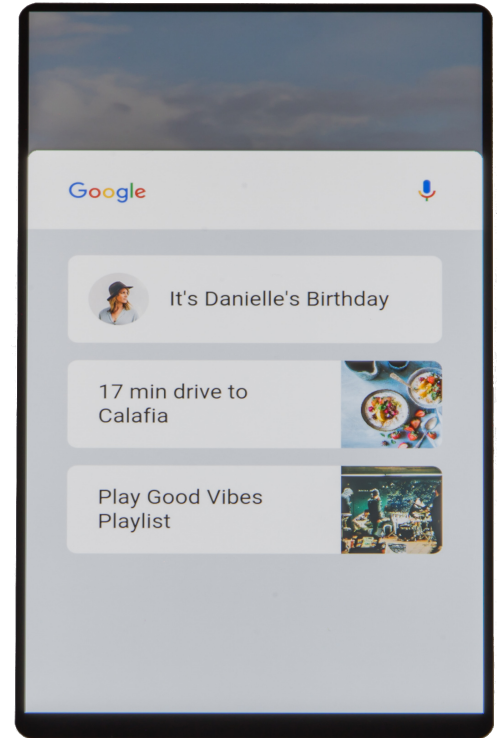
# Fuchsia! To be or not to be...

Pink + Purple == Fuchsia (a new Operating System)



# Fuchsia

<https://fuchsia.dev>



# Fuchsia is not Linux!

---

## Supporting Languages

- FIDL (Fuchsia Interface Definition Language)
- C/C++
- Dart (Flutter modules)
- Go
- Python
- Rust

## Supporting Devices

- Intel NUC
- Acer Switch Alpha 12
- Pixelbook



Fuchsia



<https://github.com/DenisovAV/DevFestApp>



<https://twitter.com/ShuregDenisov>



<https://t.me/flutterdevpodcast>



<https://t.me/MobilePeopleTalks>

THANK YOU!  
QUESTIONS?