IEEE THAPAR STUDENT BRANCH & LEAD Learn | Emerge | Aspire | Desire

PRESENTS

FLUTTER!

APP DEVELOPMENT BOOTCAMP

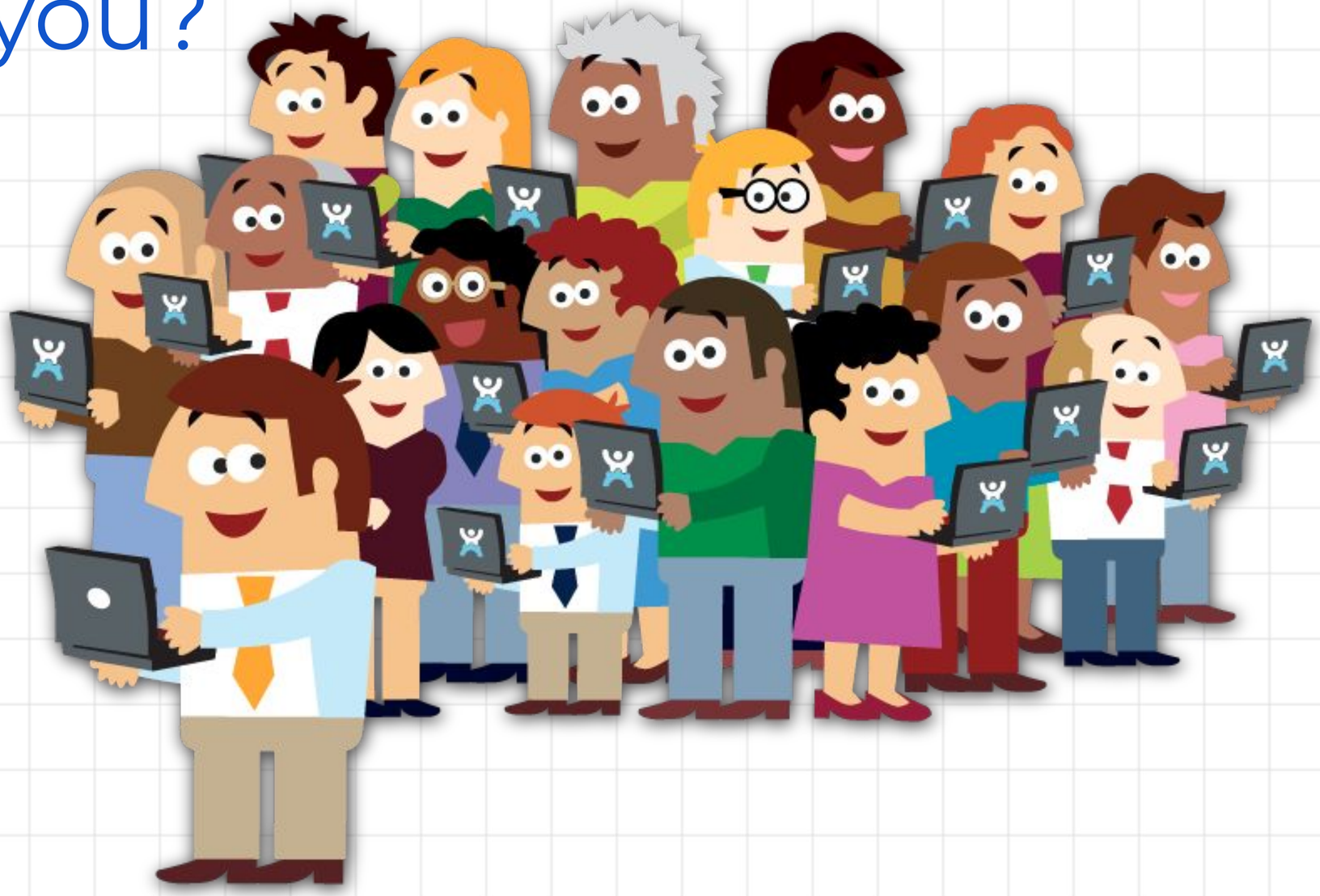
Your first step towards the real development!

5-8PM | January 14-15, 2020

Thapar Institute, Patiala

Tell me about you?

Developers ,
Designers , Mac ,
Linux , Windows ,
Web , Mobile ,
Android , ios ?



Agenda

Day 1 (January 14)		Day 2 (January 15)
15:00-17:00	Pre Workshop (Installation & Setup)	
17:00-17:15	Registration	
17:15-17:30	Opening keynote	
17:30-18:00	A basic Context About Apps	What are Native, Web, & Hybrid Apps?
18:00-18:30	What's flutter? Who's flutter for?	What makes Flutter awesome? Widgets, Rendering & performance
18:30-19:00	Fun Quiz	
19:00-19:30	Let's code. Dartpad / Android Studio.	Building Beautiful UIs.
19:30-20:30	How to go online? Connecting to network.	What are APIs? HTTP requests? Parsing JSON efficiently.
20:30-21:00	Doubts Resolution. One on One interaction.	
	See yaa tomorrow!	

Context about Apps

Native apps (advantages)

- **Quality**
- Aspect Ratios
- **Speed**
- IDE
- **Standing on the shoulders of giants**
- **Advanced UI Interactions**
- Native Look and Feel
- Usability
- New Features

They are Android(Java), Ios(Swift), Windows mobile(.net)

Web apps (advantages)

- **Reduced Development Costs**
- Ease of Integration
- Simplified Maintenance
- Easy installation
- Accessible anywhere
- **Easily customisable**
- Accessible for a range of devices
- **Easier to develop**

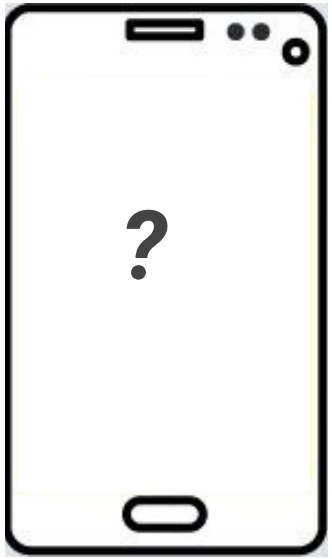
Use web technologies to build an “app”

Hybrid apps (advantages)

- **Reduced Development Costs**
- **Improved UI/UX**
- **Ease of Integration**
- **Offline Support**
- **Simplified Maintenance**

Single code base that compiles natively

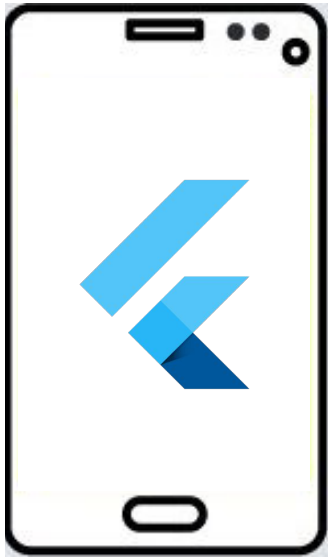
What do users(client, end user, dev, designer) want?



- **Reduced Development Costs**
- **Improved UI/UX**
- **Ease of Integration**
- **Offline Support**
- **Simplified Maintenance**
- **Quality**
- **Speed**
- **Standing on the shoulders of giants**
- **Advanced UI Interactions**

Single code base that compiles natively

What If I told you that flutter has got!



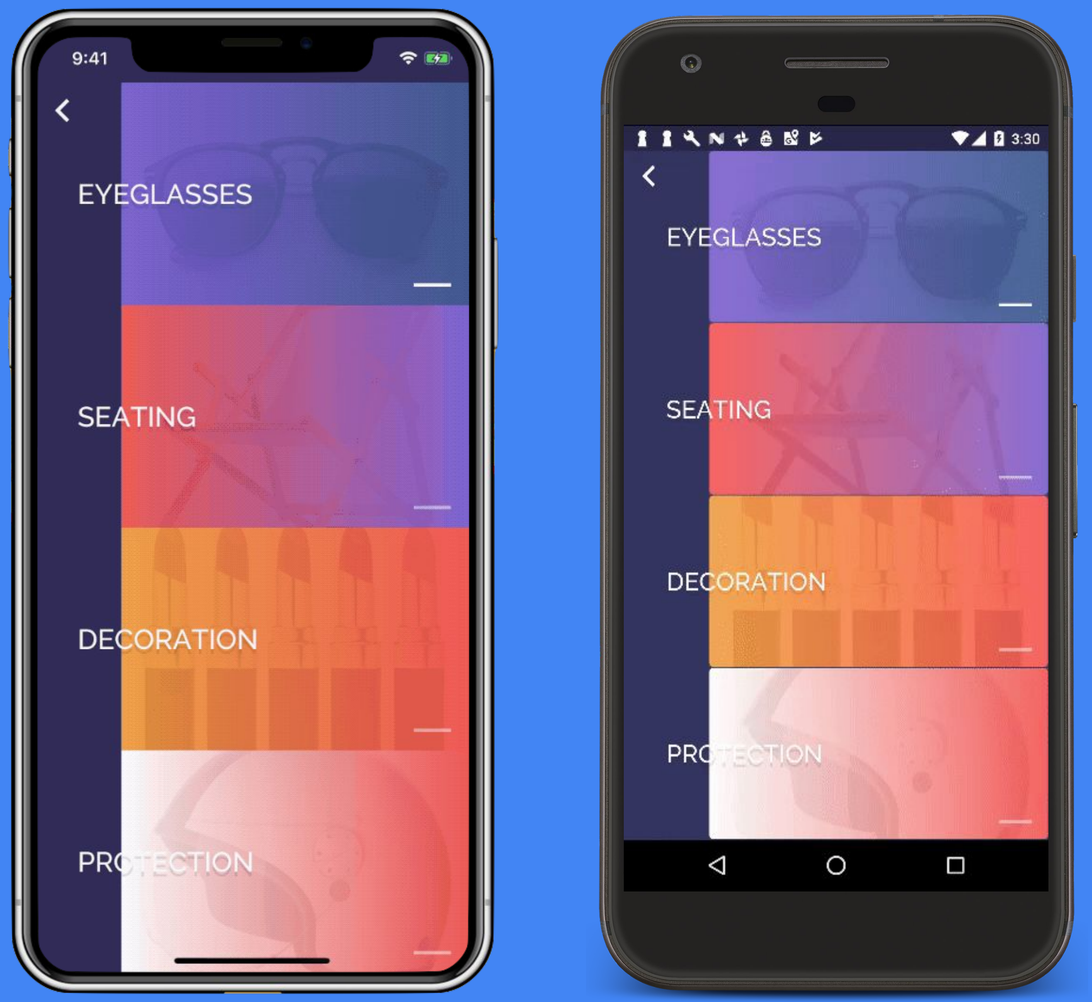
- **Reduced Development Costs**
- **Improved UI/UX**
- **Ease of Integration**
- **Offline Support**
- **Simplified Maintenance**
- **Quality**
- **Speed**
- **Standing on the shoulders of giants**
- **Advanced UI Interactions**

Single code base that compiles natively

Flutter

Flutter is Google's UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase

Made by Google

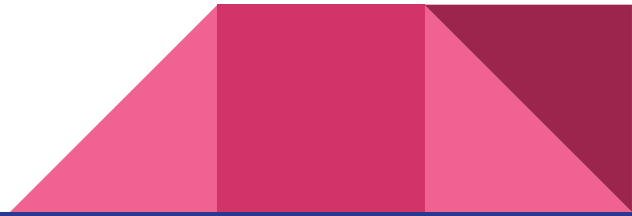


Who is Flutter for?

Designers converge on a brand-driven experience on Android and iOS

Prototypers enjoy a high-fidelity and fast way to build working prototypes.

Developers benefit from fantastic developer tools, an easy-to-use language, a rich set of widgets and great IDE support. Flutter frees up valuable time for working on features and delightful experiences.



Expressive, beautiful UIs

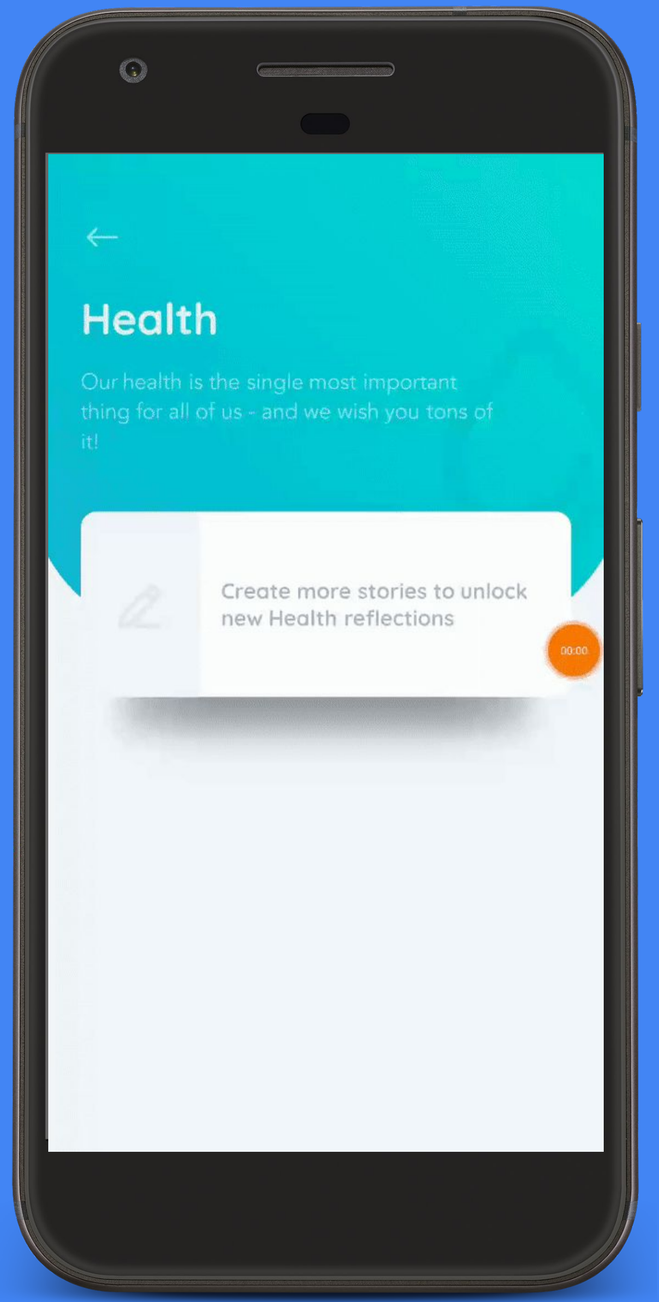
Control every pixel on the screen

Make your brand come to life

Never say "no" to your designer

Stand out in the marketplace

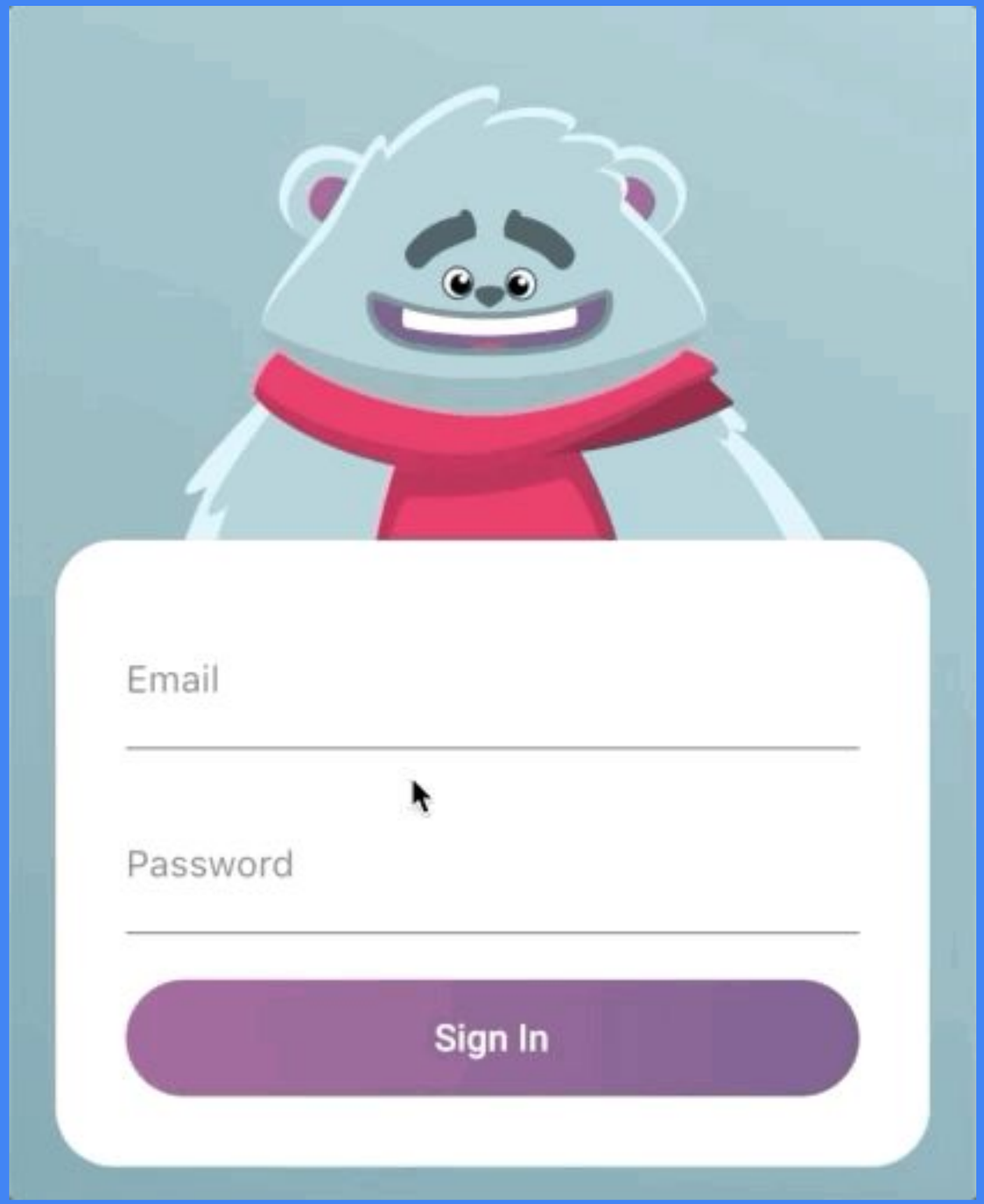
Win awards with beautiful UI



Beautiful Animations

Flutter support many types of Animations

- Tween
- Hero
- Sliver
- Transform
- FadeInWidget
- Animation Builder
- AnimatedOpacity
- Physics-based animation



Fast

Brings the power of a games engine to user experience development

60fps, GPU accelerated

Compiled to native machine code

Productive

Sub-second reload times

Paint your app to life

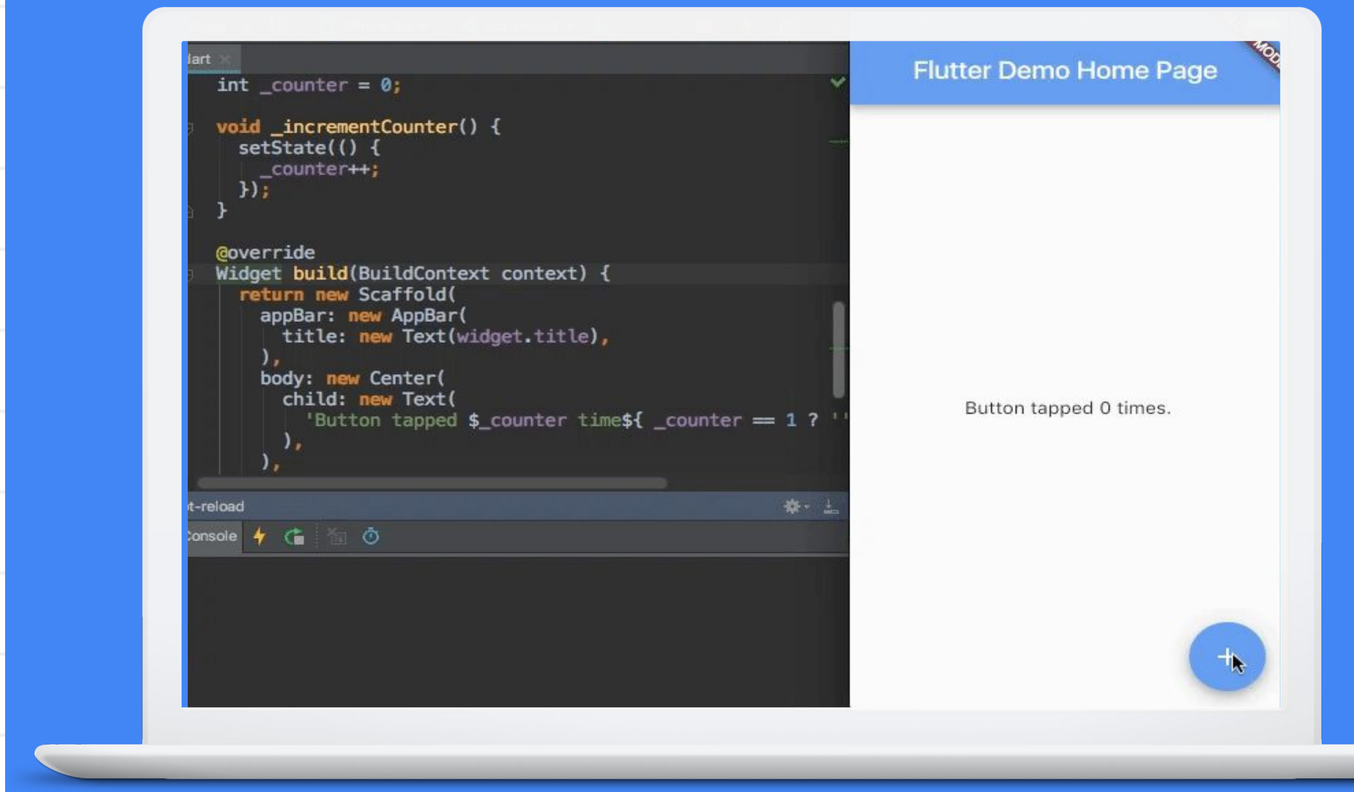
Iterate rapidly on features

Test hypotheses quicker than ever

More time to experiment & test features

Single-codebase for faster collab

3X Productivity Gains



Android

iOS

Flutter

Web
(Technical Preview)

Desktop

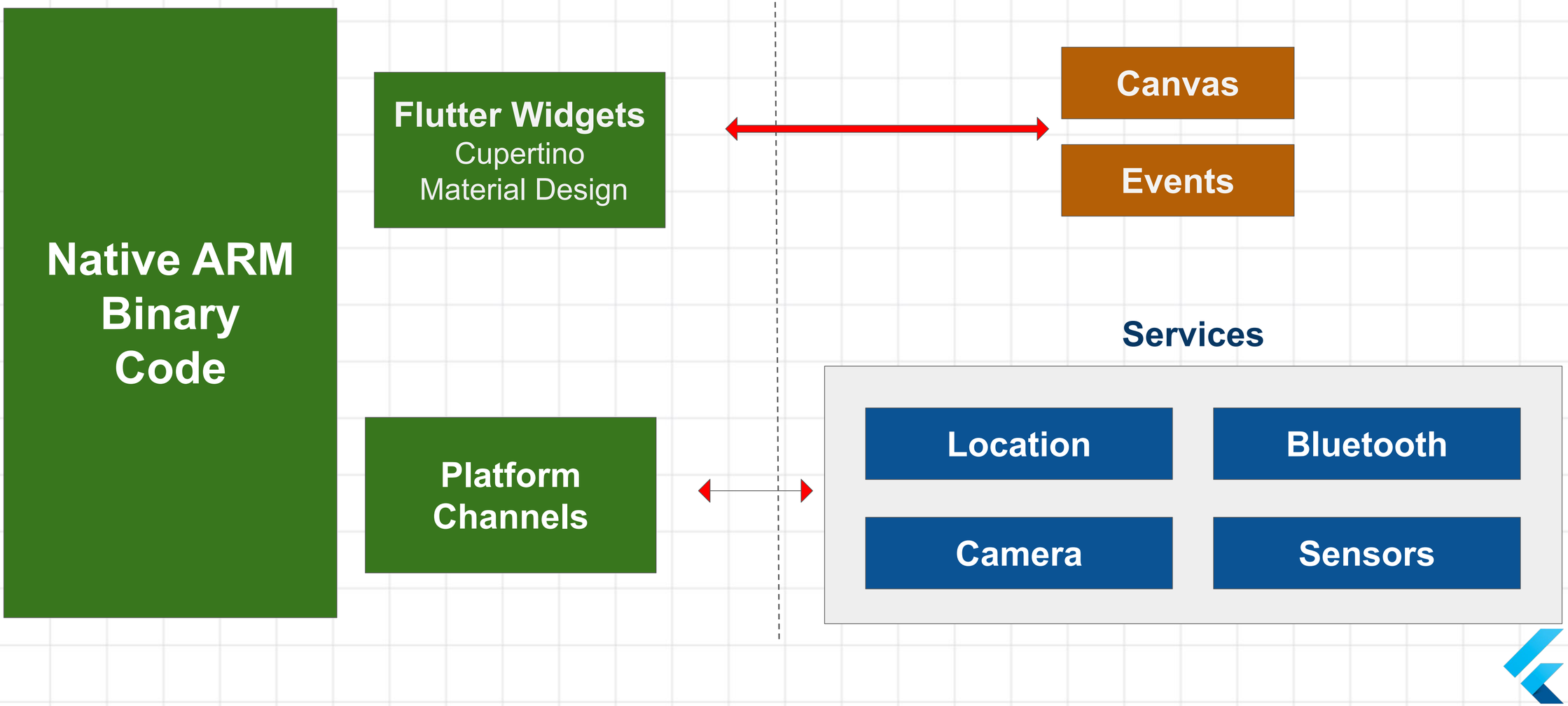
What makes Flutter unique?

- Compiles to Native Code (ARM Binary code)
- No reliance on OEM widgets
- No bridge needed
- No markup language (only Dart)

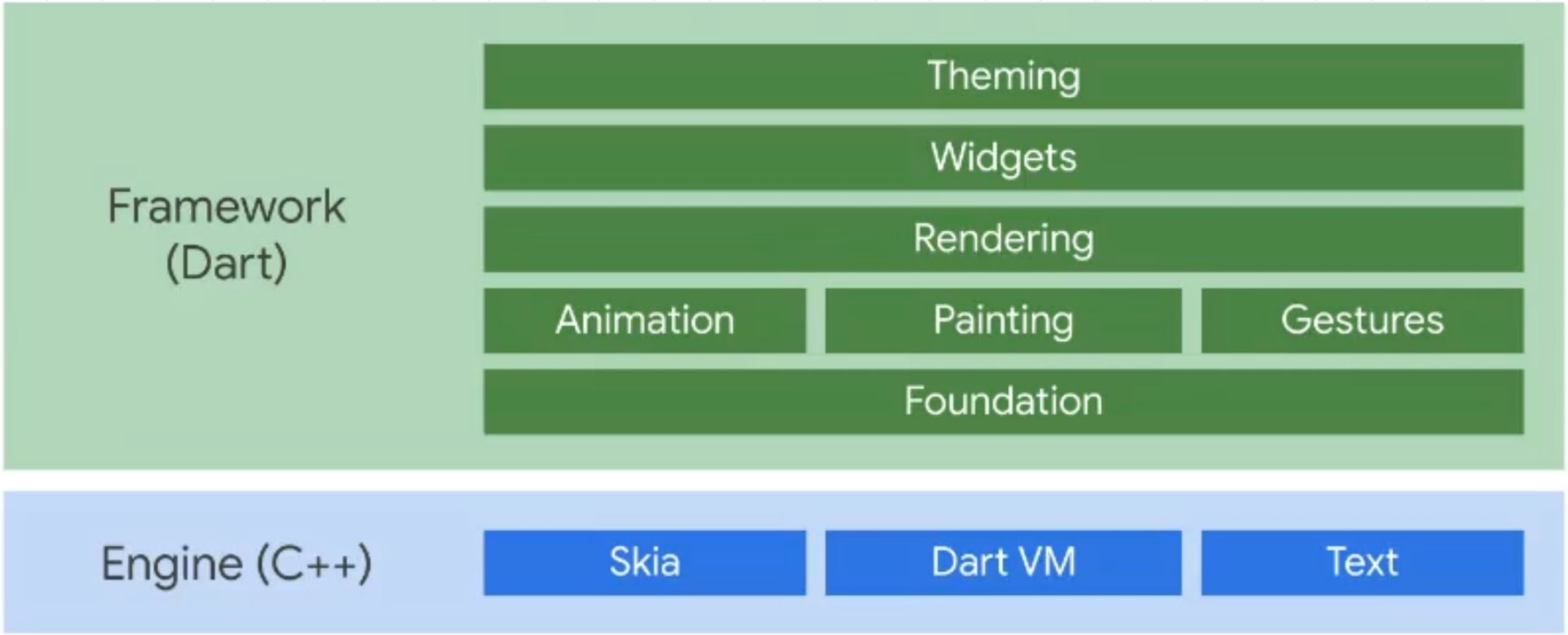
Key component of the Google Developer family

Flutter offers a portable, high-quality UI toolkit, and a fast, expressive way to build native app UIs.

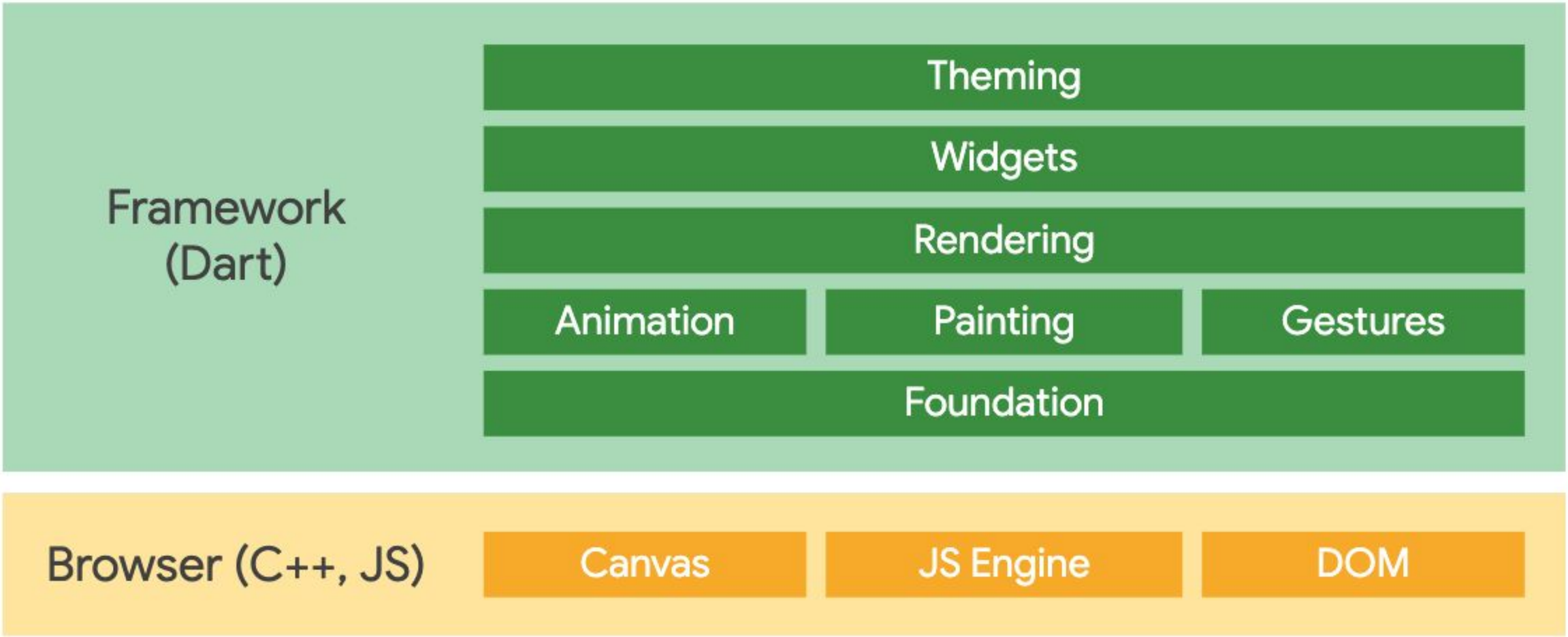
Flutter Approach



Flutter Architecture



Flutter Architecture

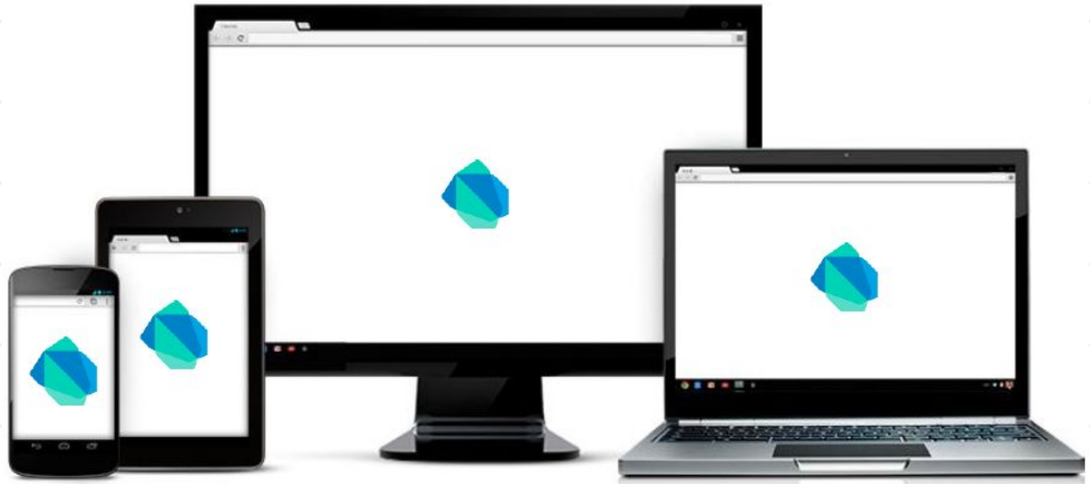


Dart

Made by Google

Dart is a client-optimized language for fast apps on any platform!
(Web - Desktop - Mobile - Embedded)?

- Language and Libraries
- Packages manager https://pub.dev
- Virtual machine
- Compile to Javascript dart2js



- Sound type system
- Tree Shaking
- Rich core libraries
- Multi-gen, lockless GC

- Single codebase for Android and iOS
- Rapid development cycles
- Great tooling

Design-oriented Development Flow

What do you see here?

Oeschinen Lake Campground

Kandersteg, Switzerland

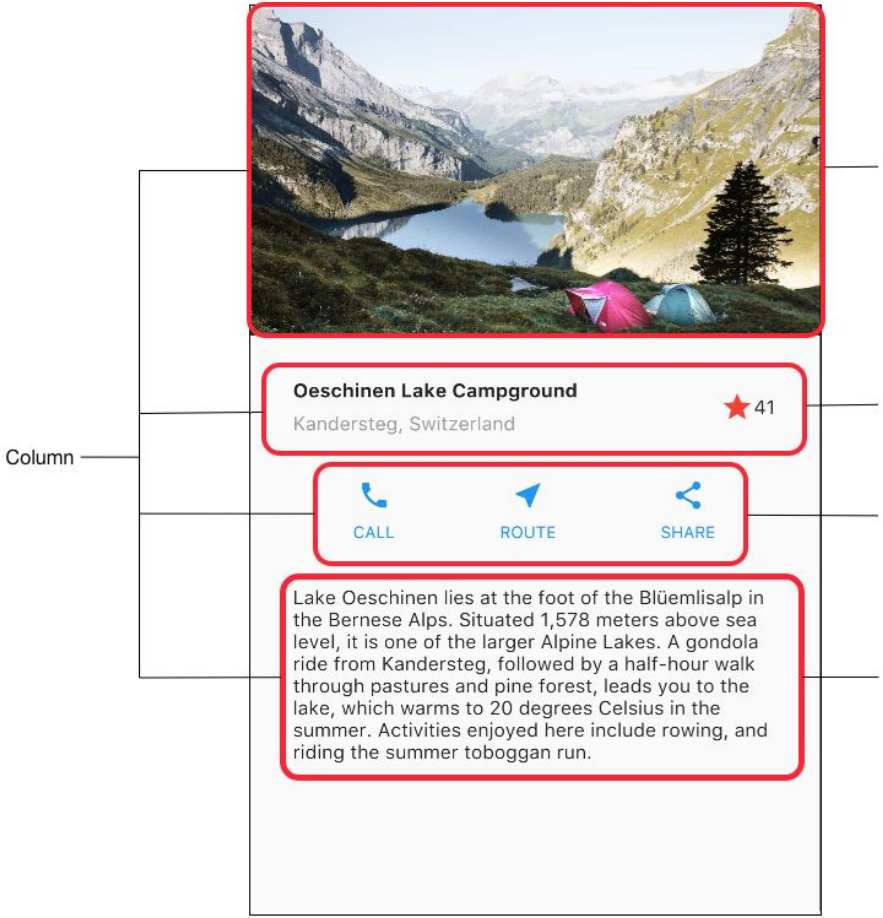
41

CALL ROUTE SHARE

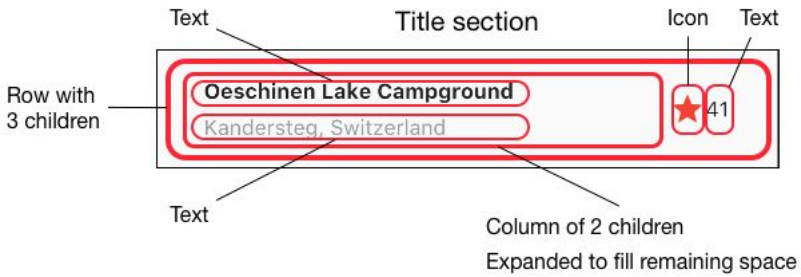
Lake Oeschinen lies at the foot of the Blüemlisalp in the Bernese Alps. Situated 1,578 meters above sea level, it is one of the larger Alpine Lakes. A gondola ride from Kandersteg, followed by a half-hour walk through pastures and pine forest, leads you to the lake, which warms to 20 degrees Celsius in the summer. Activities enjoyed here include rowing, and riding the summer toboggan run.

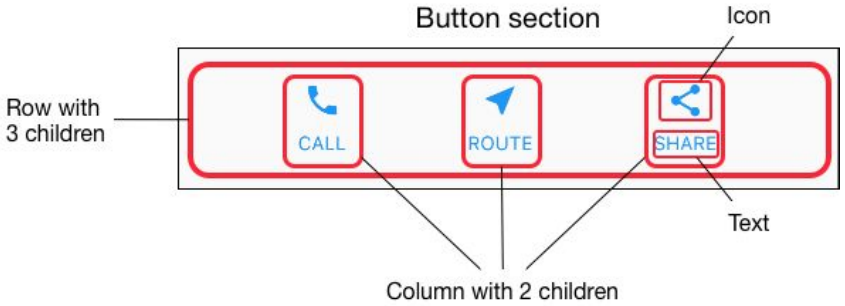
Diagram the Layout

- Look for rows and columns
- Is there a grid?
- Any overlapping elements?
- Do we need tabs?
- Padding, alignment or borders needed?



Designing bottom up





HTML/CSS Analogs in Flutter

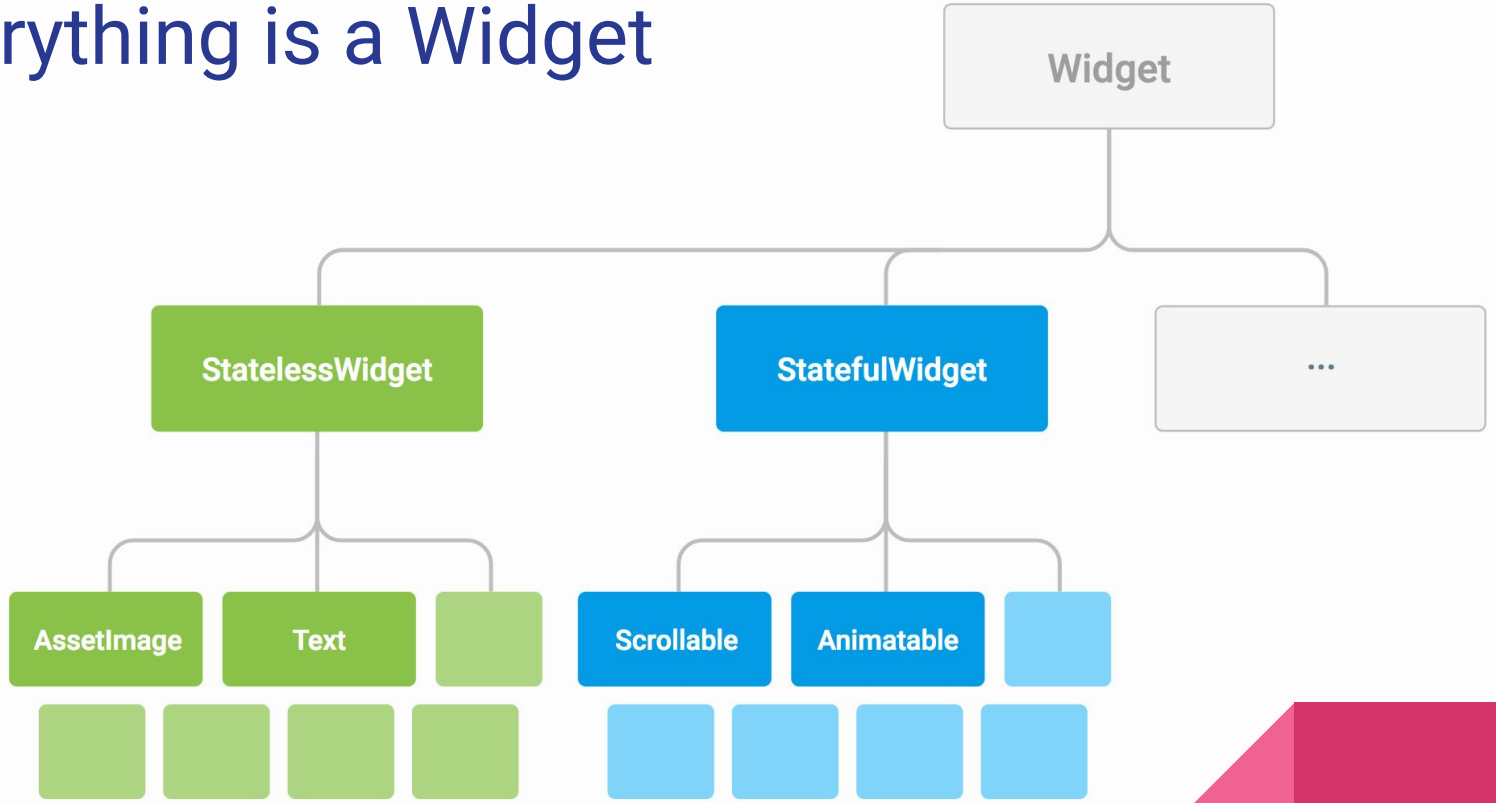
```
<div class="greybox">
    Lorem ipsum
</div>

.greybox {
      background-color: #e0e0e0; /* grey 300 */
      width: 320px;
      height: 240px;
      font: 900 24px Georgia;
    }
```

```
var container = new Container( // grey box
  child: new Text(
    "Lorem ipsum",
    style: new TextStyle(
      fontSize: 24.0
      fontWeight: FontWeight.w900,
      fontFamily: "Georgia",
    ),
  ),
  width: 320.0,
  height: 240.0,
  color: Colors.grey[300],
);
```

The Power of Widgets

Everything is a Widget



StatefulWidget
vs.
StatelessWidget

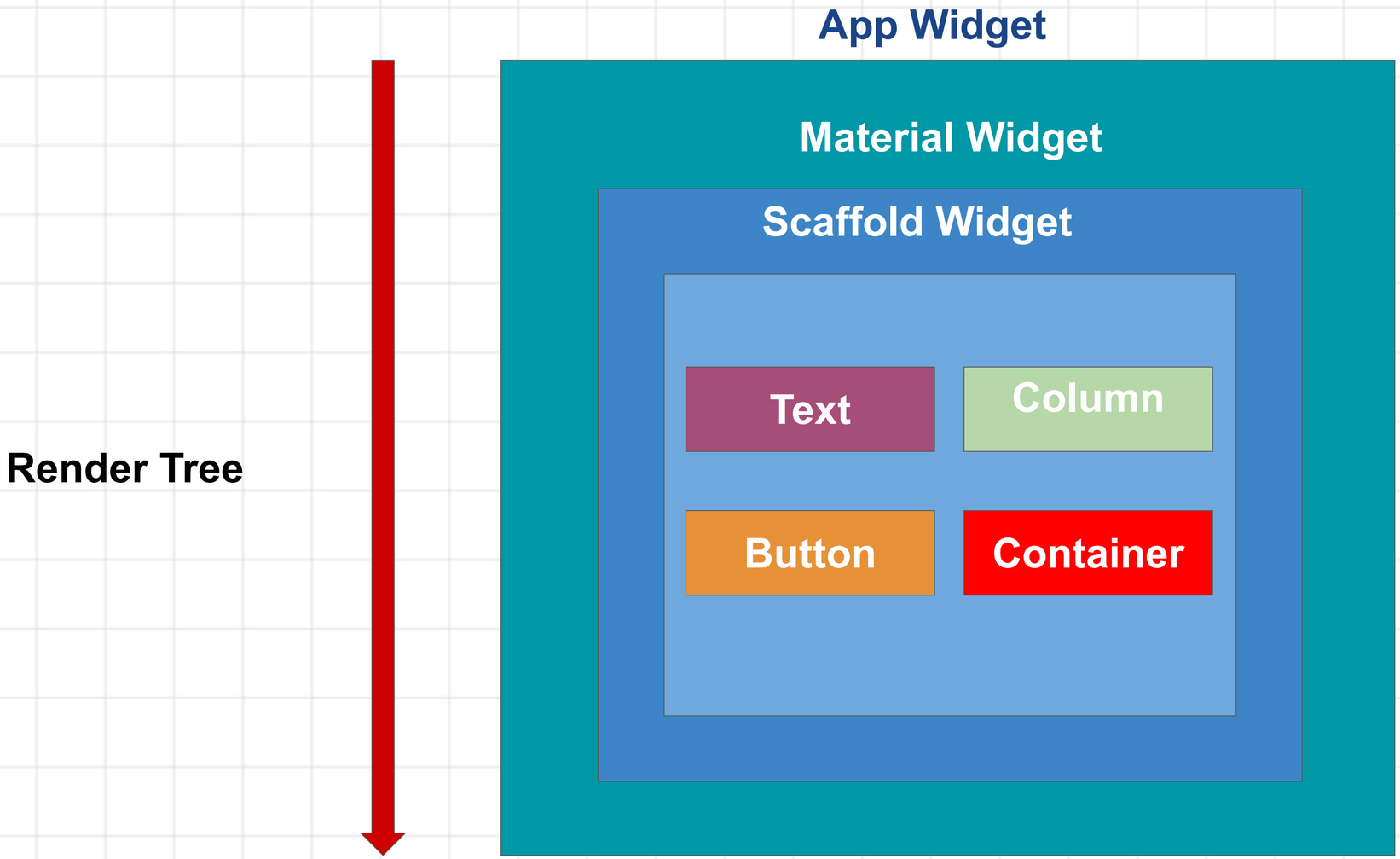
Customizing and extending Widgets

Flutter’s Widget system was designed to be easily customizable

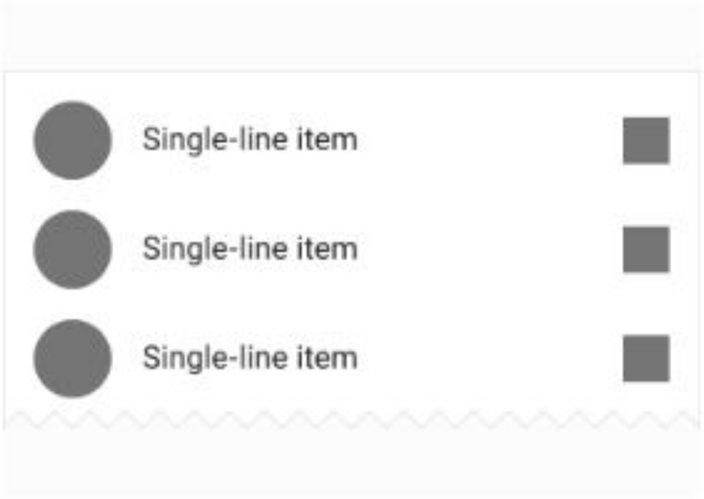
Composition: Widgets are built out of smaller widgets that you can reuse and combine in novel ways to make custom widgets

```
class RaisedButton extends StatelessWidget {
    ...
}
```

Everything is a Widget



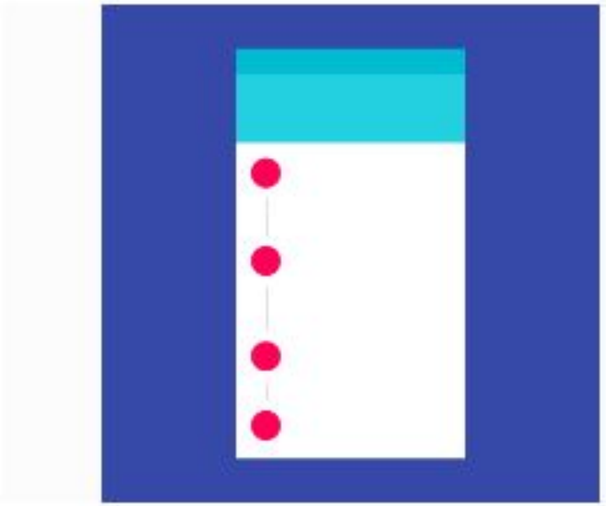
Layout



ListTile

A single fixed-height row that typically contains some text as well as a leading or trailing icon.

Documentation



Stepper

A material stepper widget that displays progress through a sequence of steps.

Documentation

Divider

A one logical pixel thick horizontal line, with padding on either side.

Documentation

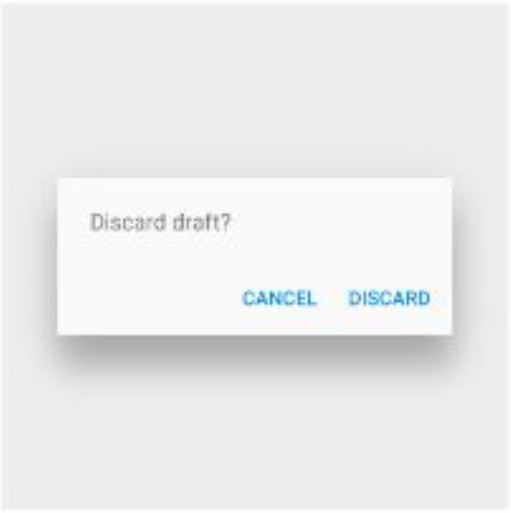
https://flutter.dev/docs/reference/widgets

Dialogs, alerts, and panels

SimpleDialog

Simple dialogs can provide additional details or actions about a list item. For example they can display avatars icons clarifying subtext or orthogonal actions...

Documentation



AlertDialog

Alerts are urgent interruptions requiring acknowledgement that inform the user about a situation. The AlertDialog widget implements this component.

Documentation

BottomSheet

Bottom sheets slide up from the bottom of the screen to reveal more content. You can call showBottomSheet() to implement a persistent bottom sheet or...

Documentation

https://flutter.dev/docs/reference/widgets

Efficient Tooling

`$ flutter doctor`

Checks your environment and displays a report to the terminal window

`$ flutter upgrade`

Updates both the Flutter SDK and your packages

pubspec.yaml

```
name: flutter_project

description: An amazing Flutter project using Firebase Auth

dependencies:
  flutter:
    sdk: flutter
  firebase_auth: "^0.2.5"
```

pubspec.yaml

```
name: flutter_project

description: An amazing Flutter project using Firebase Auth

dependencies:
  flutter:
    sdk: flutter
  firebase_auth: ">=0.1.2 <0.2.6"
```

```
$ flutter packages get
```

```
$ flutter packages upgrade
```

Will retrieve the highest available version of the package

`$ flutter format`

Automatically formats your code according to the Flutter-style

`$ flutter analyze`

Analyzes your code and help you find possible mistakes

Hot Reload

Injecting updated source code files into the running Dart VM

Stateful: App state is retained after a reload.

Quickly iterate on a screen deeply nested in your app

Dart Observatory

Statement-level single-stepping debugger and profiler

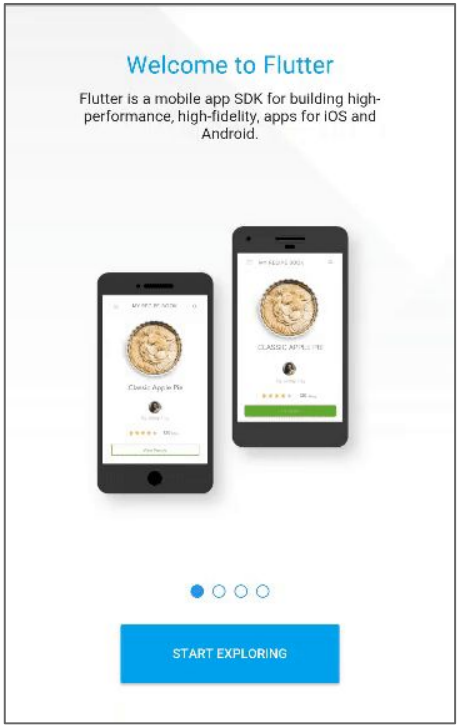
Automatically running when you start your app using `flutter run`

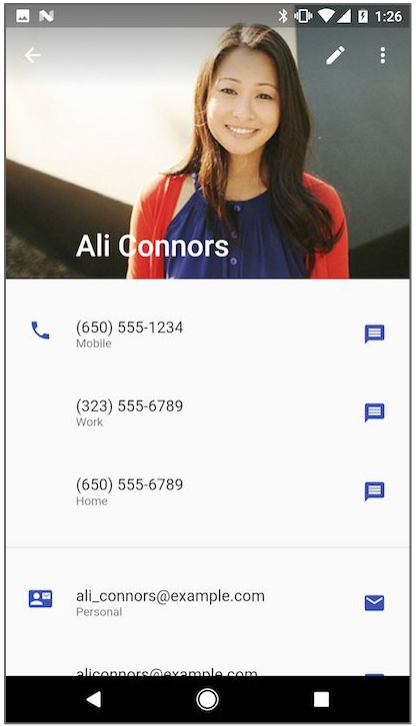
See which lines of code have executed

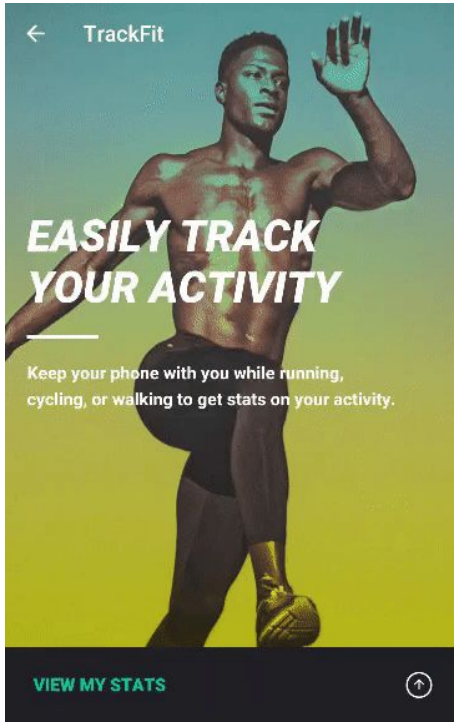
Check out memory allocations

Debug memory leaks & fragmentation

Great looking and fast Widgets



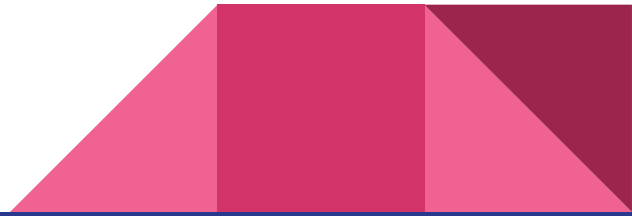


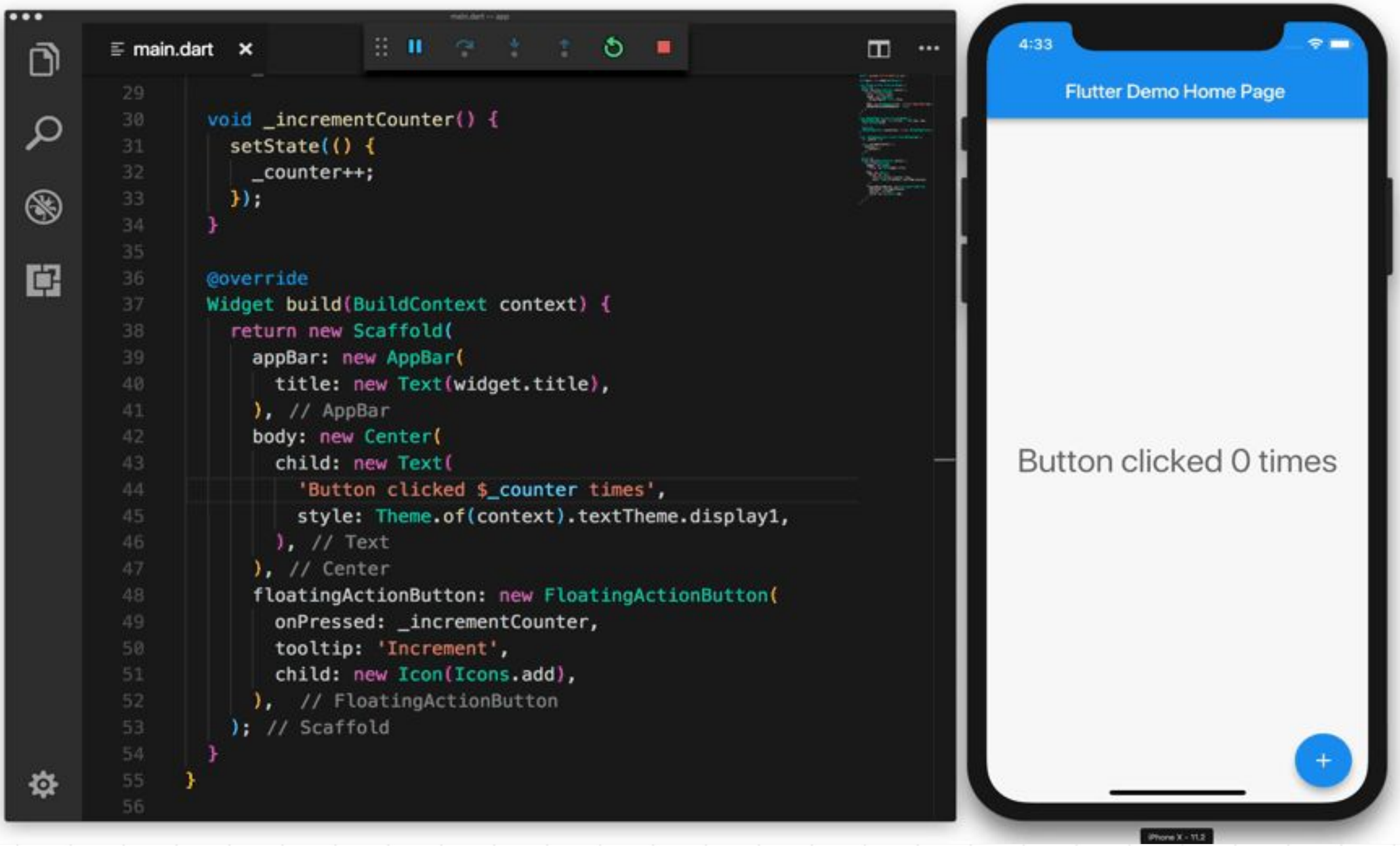


Goodbye, global layout system

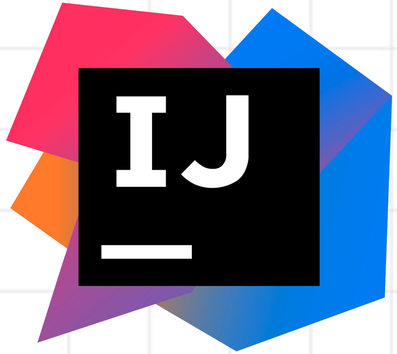
```
new Center(
    child: new Text('Centered Text', style: textStyle),
)
```

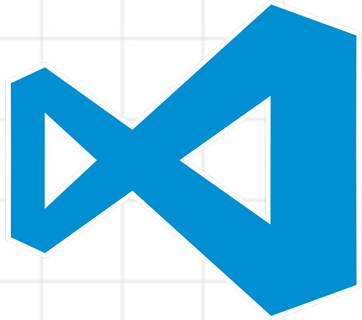
Local layouts: Every Widget defines it’s **own** layout. No need to tell the parent that it’s children are supposed to be centered.




main.dart
void _incrementCounter() {
setState(() {
_counter++;
});
}
@override
Widget build(BuildContext context) {
return new Scaffold(
appBar: new AppBar(
title: new Text(widget.title),
), // AppBar
body: new Center(
child: new Text(
'Button clicked $_counter times',
style: Theme.of(context).textTheme.display1,
), // Text
), // Center
floatingActionButton: new FloatingActionButton(
onPressed: _incrementCounter,
tooltip: 'Increment',
child: new Icon(Icons.add),
), // FloatingActionButton
); // Scaffold
}
}
4:33
Flutter Demo Home Page
Button clicked 0 times

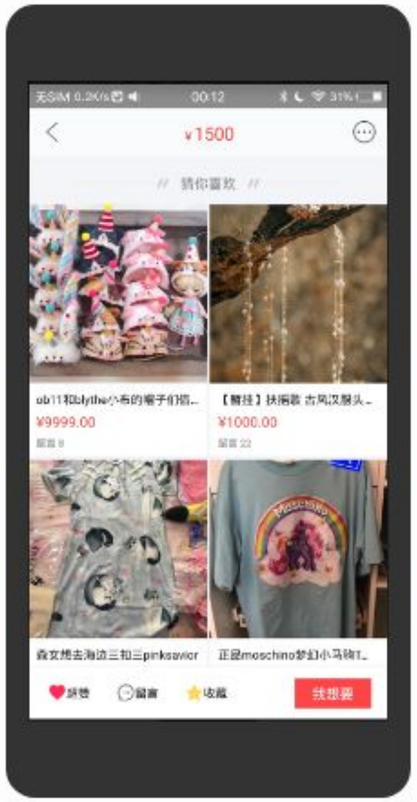
IDE Support ?





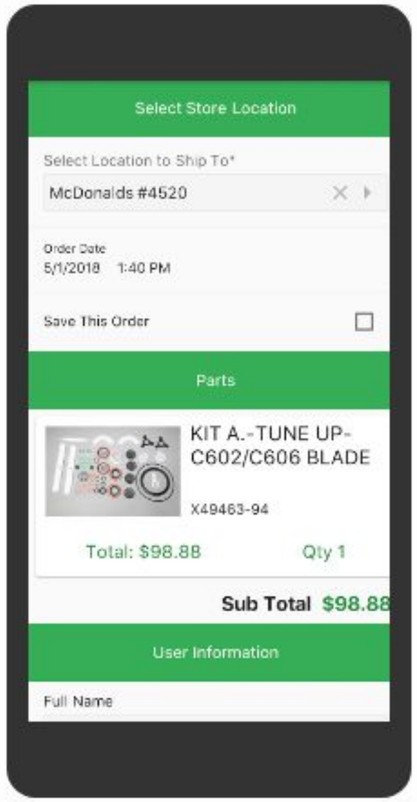
Flutter Showcase

Alibaba



Google Ads

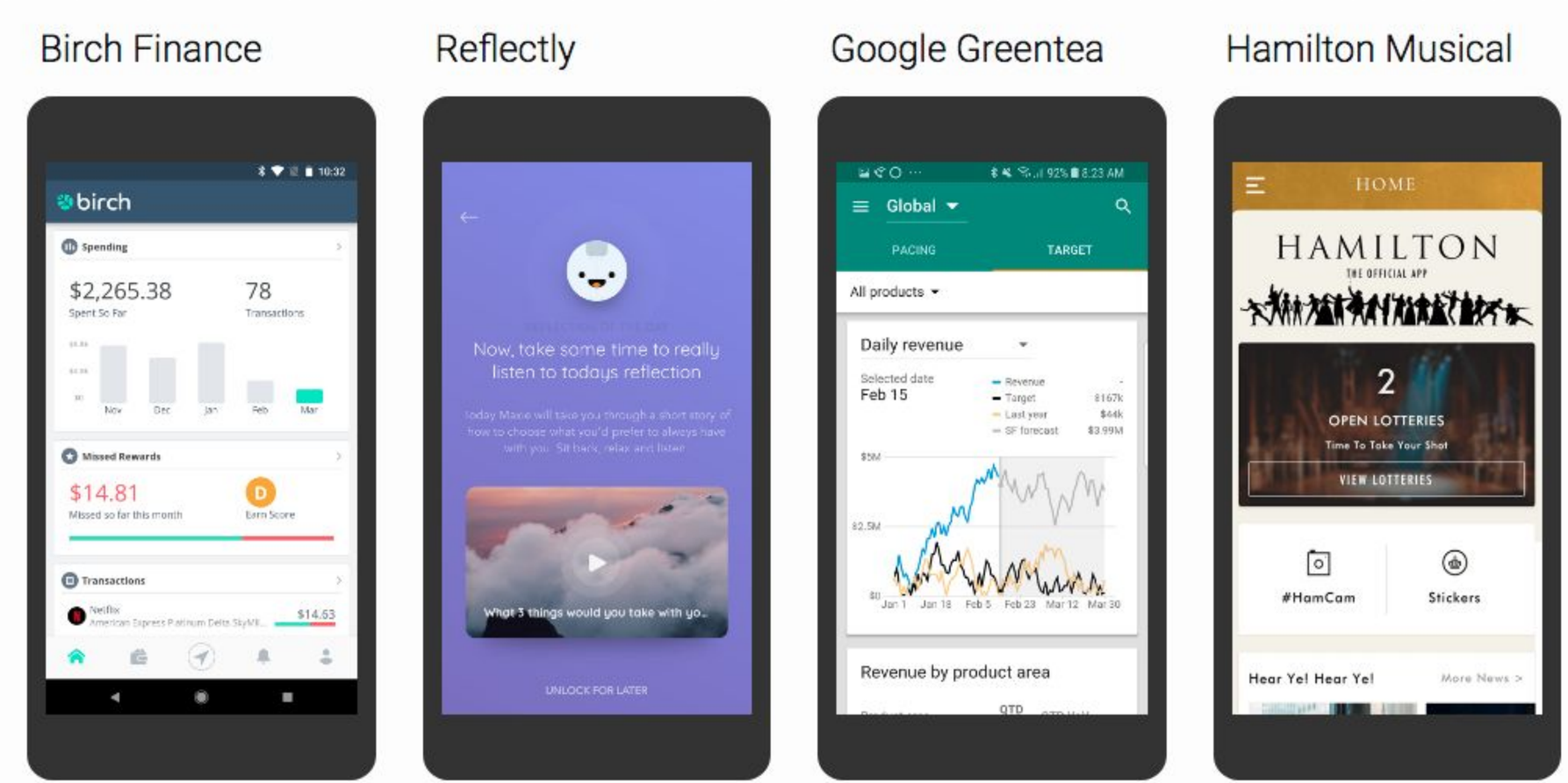
AppTree



Topline

https://flutter.dev/showcase & https://itsallwidgets.com

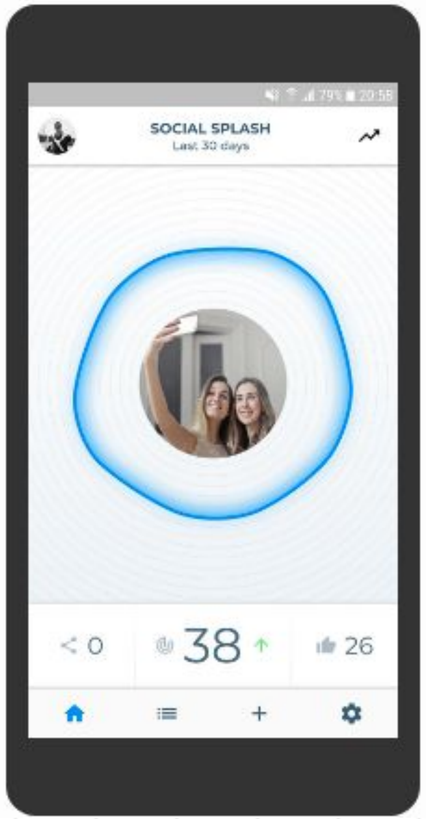
Flutter Showcase



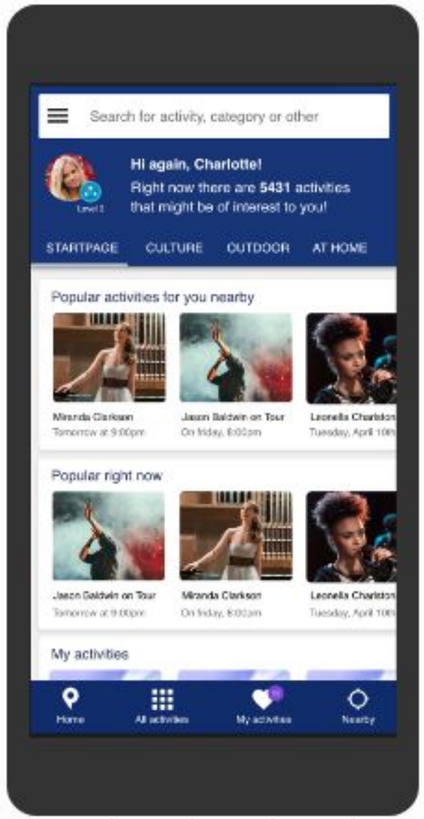
https://flutter.dev/showcase & https://itsallwidgets.com

Flutter Showcase

Hookle

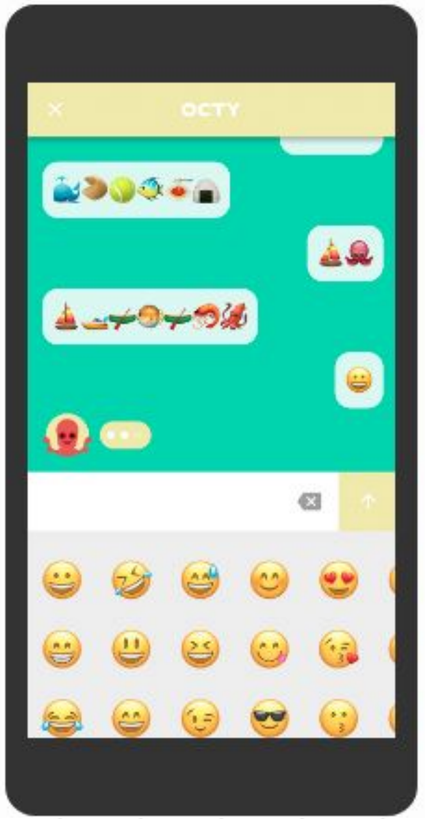


OfflinePal



VivaBem UOL

BetaBubs



https://flutter.dev/showcase & https://itsallwidgets.com

{>,;}

HAPPY CODING

Shivam Goyal

Flutter Dev | Enappd
Applied CS Facilitator
Google HashCode | AIR-33
itsshivam.com

Utkarsh Mishra

Blockchain Developer
IBC Member
SIH 2019 Winner
utkarshmishra.ml