



IT Minds

- › 180 employees - Aarhus, København, Oslo and Aalborg
- › Newest technologies
- › Our culture
- › Tech Talent Night events
 - Purpose
 - Future events
- › Orbit Lab - Thank you!
- › And then.. Orbit Lab, and Flutter!



IT MINDS





Rasmus Reimer



IT MINDS

- MSc Computer Engineering
- Senior Software Developer
- 5 years at IT Minds
- Full stack + mobile



Rasmus Thorsen



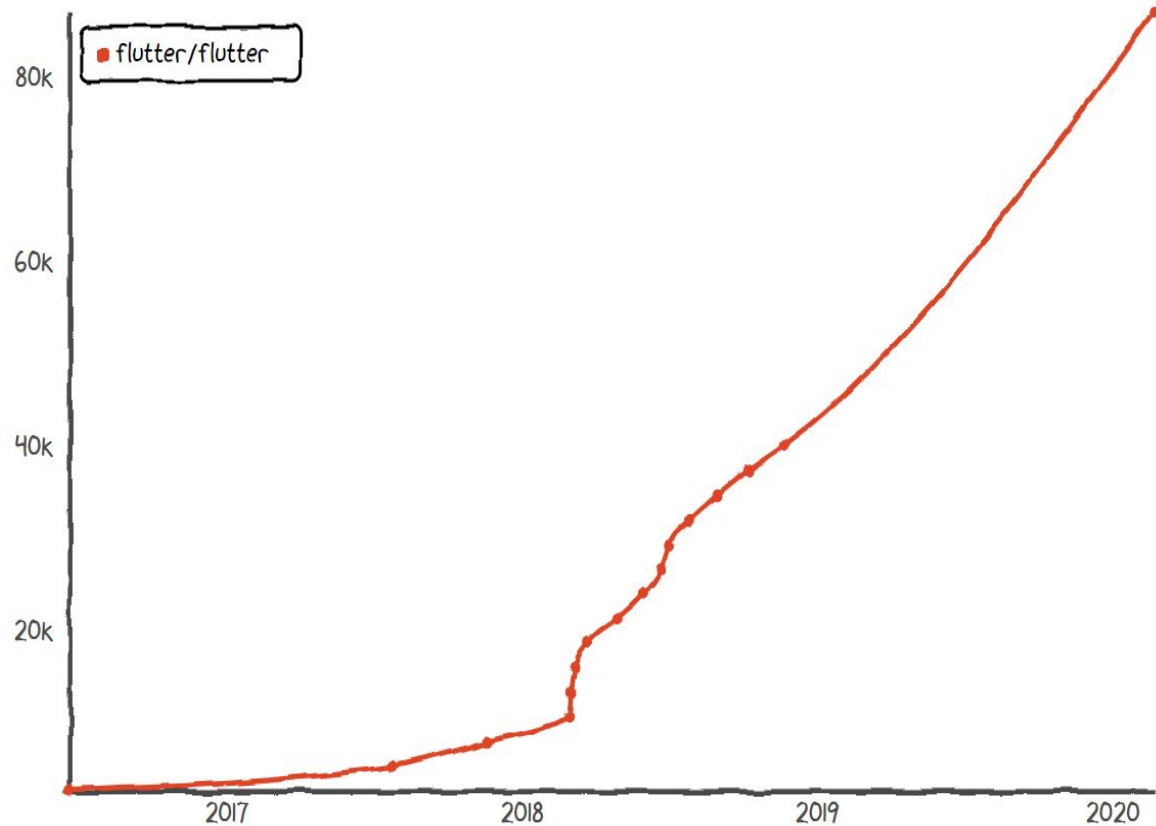
IT MINDS

- CE student at AU
- Software Developer
- 1.5 years at IT Minds
- Full stack + mobile

Flutter?

- ❏ Cross platform apps
- ❏ Unveiled 2015 Dart conf
- ❏ v1.0 released December 4, 2018
- ❏ Native performance

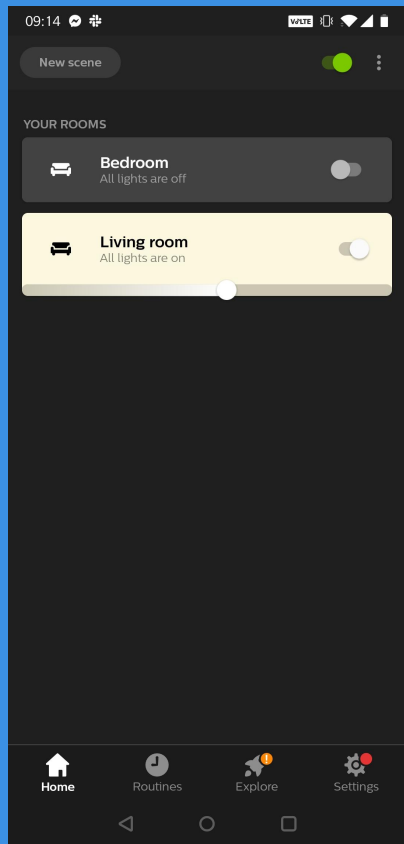
Github star history



Who is using Flutter?



Who is using Flutter?



Dart



Pros

OOP as we know it

Familiar to JavaScript and Java developers

Also compiles to JavaScript

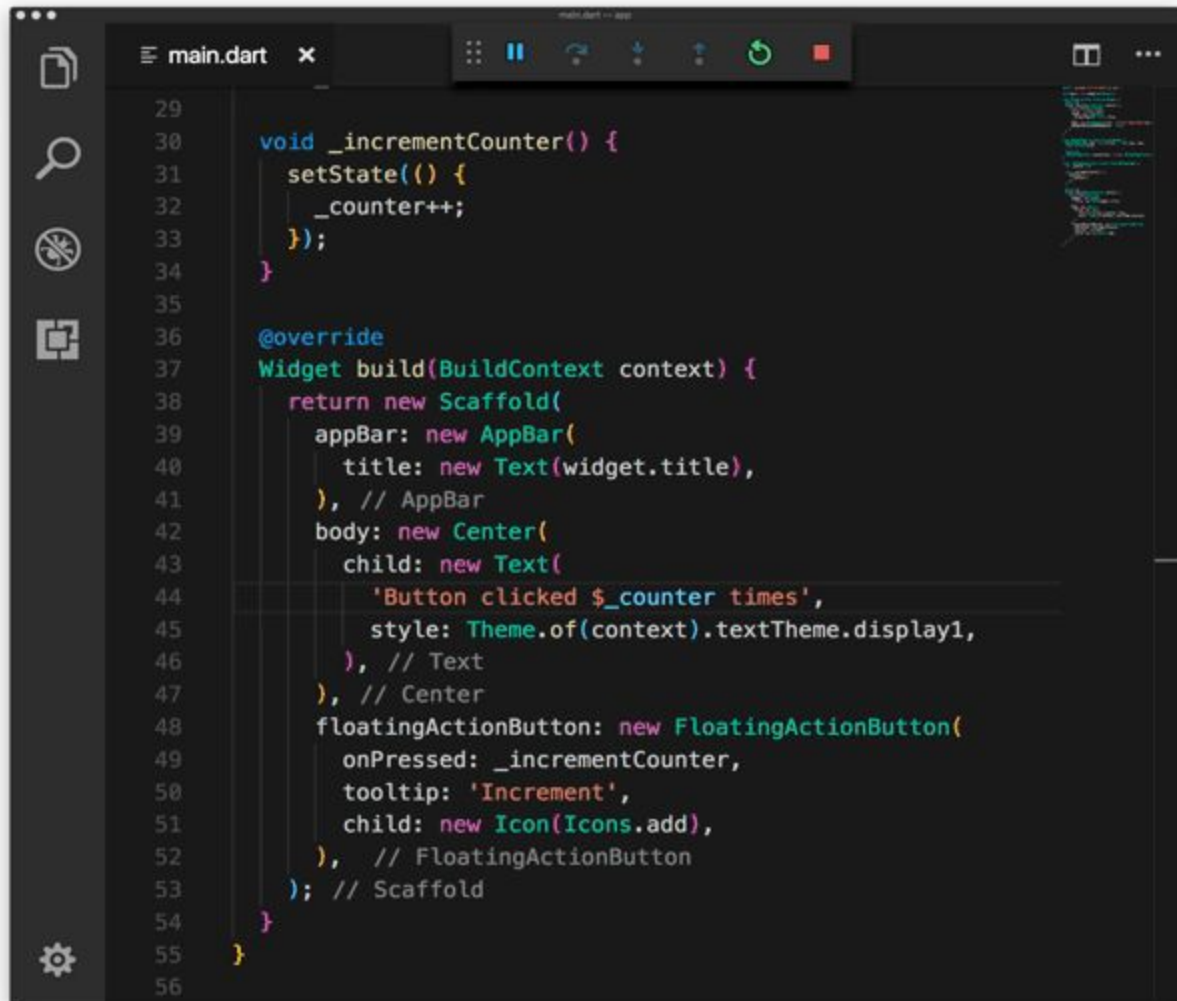
Supports async programming with
Futures and Streams

Cons

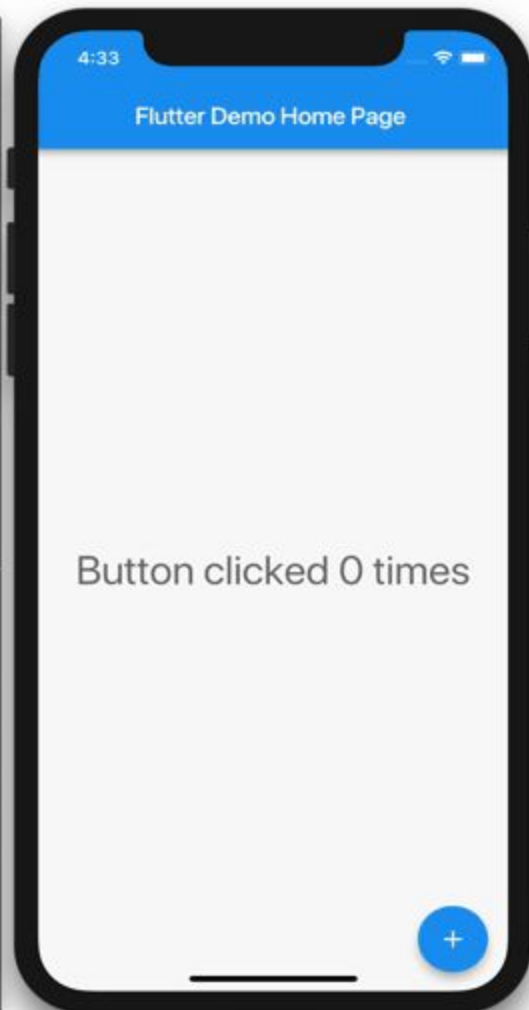
Niche

- Flutter SDK
- Hot Reloading
- Editors
 - VS Code
 - Android Studio
 - IntelliJ

Tooling



```
29
30 void _incrementCounter() {
31   setState(() {
32     _counter++;
33   });
34 }
35
36 @override
37 Widget build(BuildContext context) {
38   return new Scaffold(
39     appBar: new AppBar(
40       title: new Text(widget.title),
41     ), // AppBar
42     body: new Center(
43       child: new Text(
44         'Button clicked $_counter times',
45         style: Theme.of(context).textTheme.display1,
46       ), // Text
47     ), // Center
48     floatingActionButton: new FloatingActionButton(
49       onPressed: _incrementCounter,
50       tooltip: 'Increment',
51       child: new Icon(Icons.add),
52     ), // FloatingActionButton
53   ); // Scaffold
54 }
55
56
```



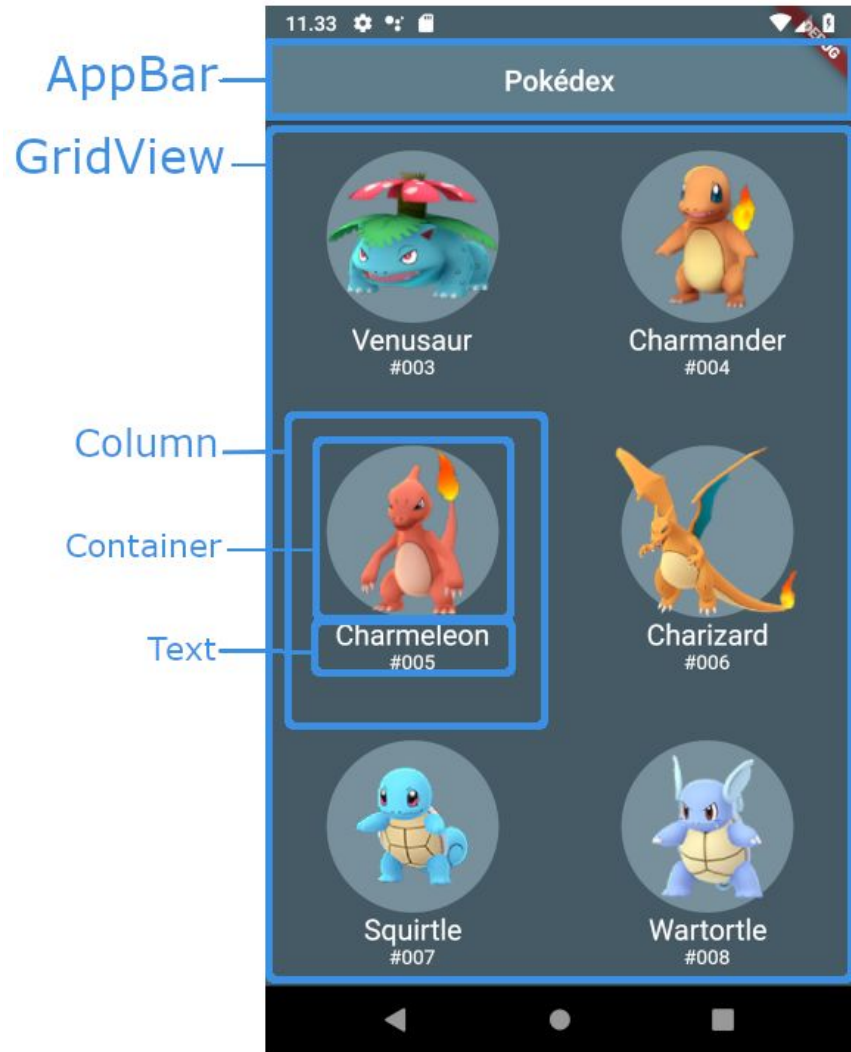
Let's get technical!

Everything's
a widget

Everything's a Widget

- ❑ The central idea is that you build your UI out of widgets.
- ❑ Widgets describe what their view should look like given their state.
- ❑ When a widget's state changes, the widget rebuilds.

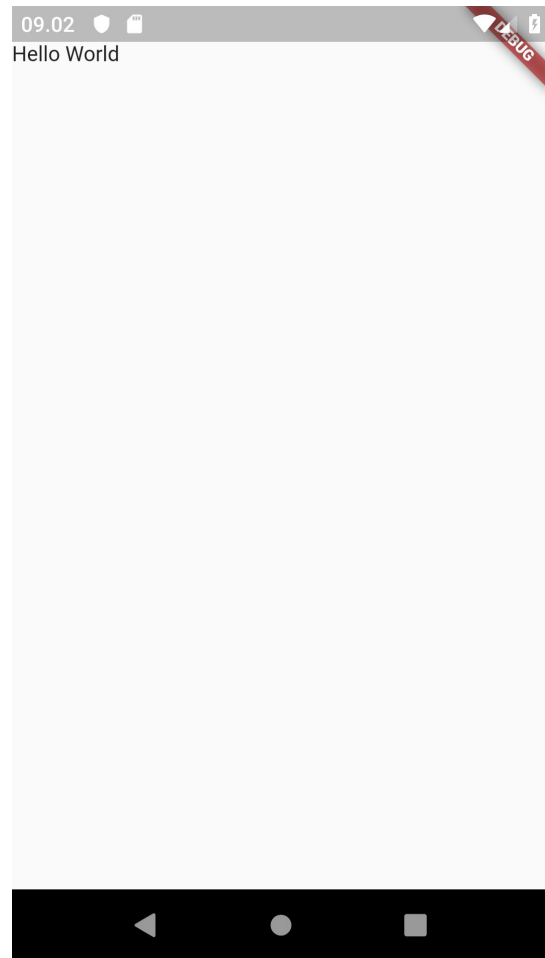
Everything's a Widget



Writing Flutter

Using Flutter widgets

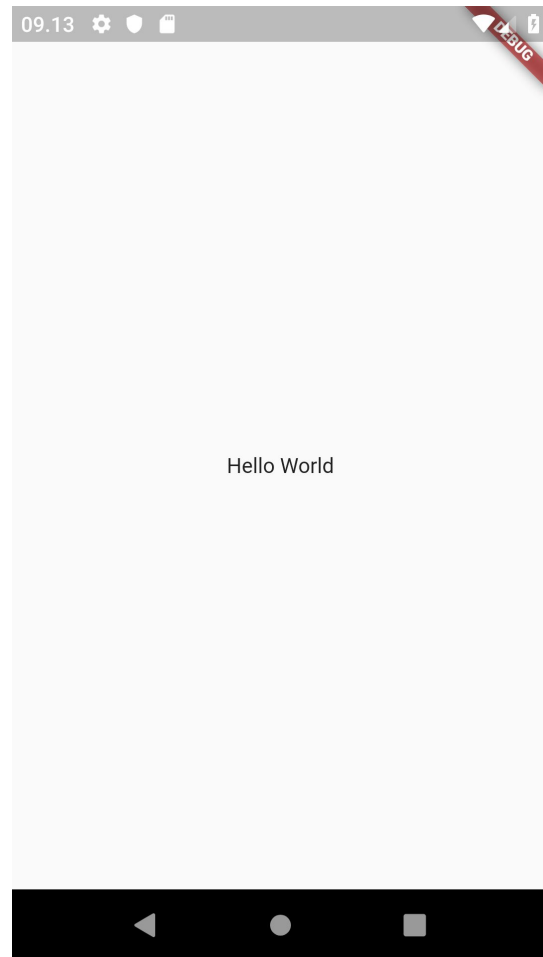
```
Text('Hello World');
```



Writing Flutter

Using Flutter widgets

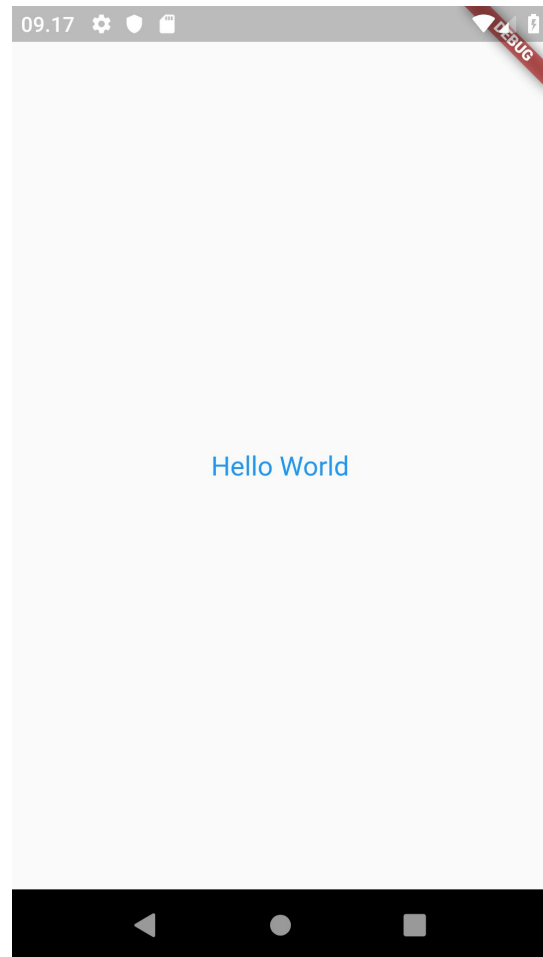
```
Center(  
  child: Text(  
    'Hello World',  
  ),  
);
```



Writing Flutter

Using Flutter
widgets

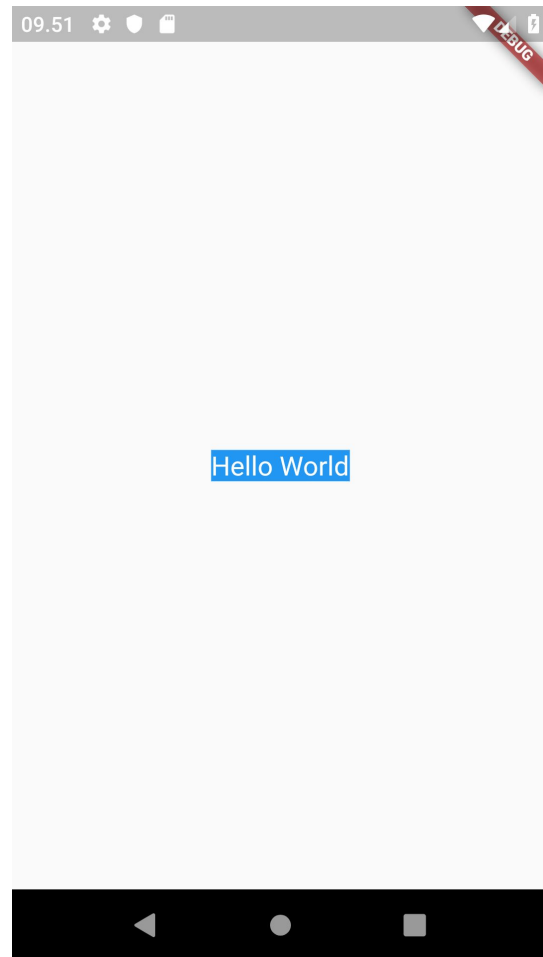
```
Center(  
  child: Text(  
    'Hello World',  
    style: TextStyle(  
      color: Colors.blue,  
      fontSize: 18,  
    ),  
  ),  
);
```



Writing Flutter

Using Flutter widgets

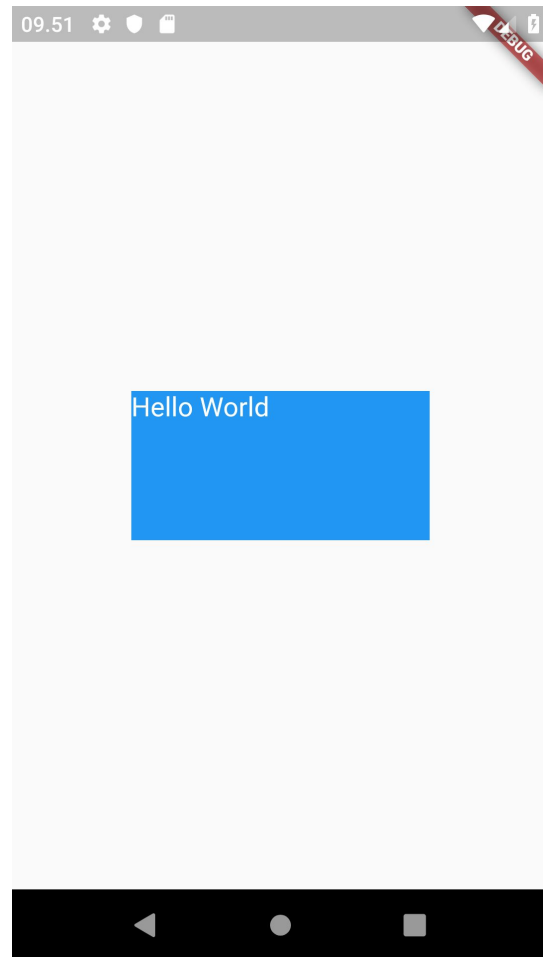
```
Center(  
  child: Container(  
    child: Text(  
      'Hello World',  
      style: TextStyle(  
        color: Colors.white,  
        fontSize: 18,  
      ),  
    ),  
    color: Colors.blue,  
  ),  
);
```



Writing Flutter

Using Flutter widgets

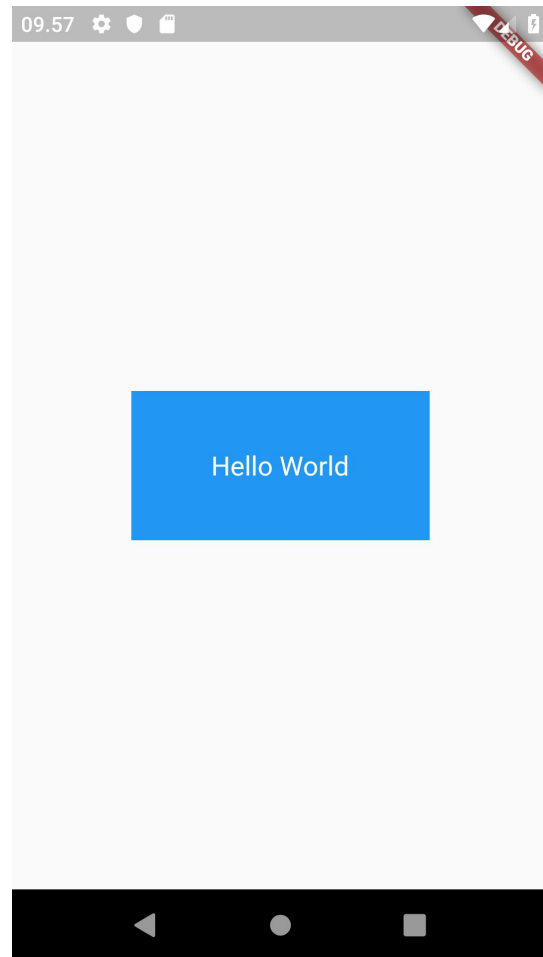
```
Center(  
  child: Container(  
    child: Text(  
      'Hello World',  
      style: TextStyle(  
        color: Colors.white,  
        fontSize: 18,  
      ),  
    ),  
    color: Colors.blue,  
    width: 200,  
    height: 100,  
  ),  
);
```



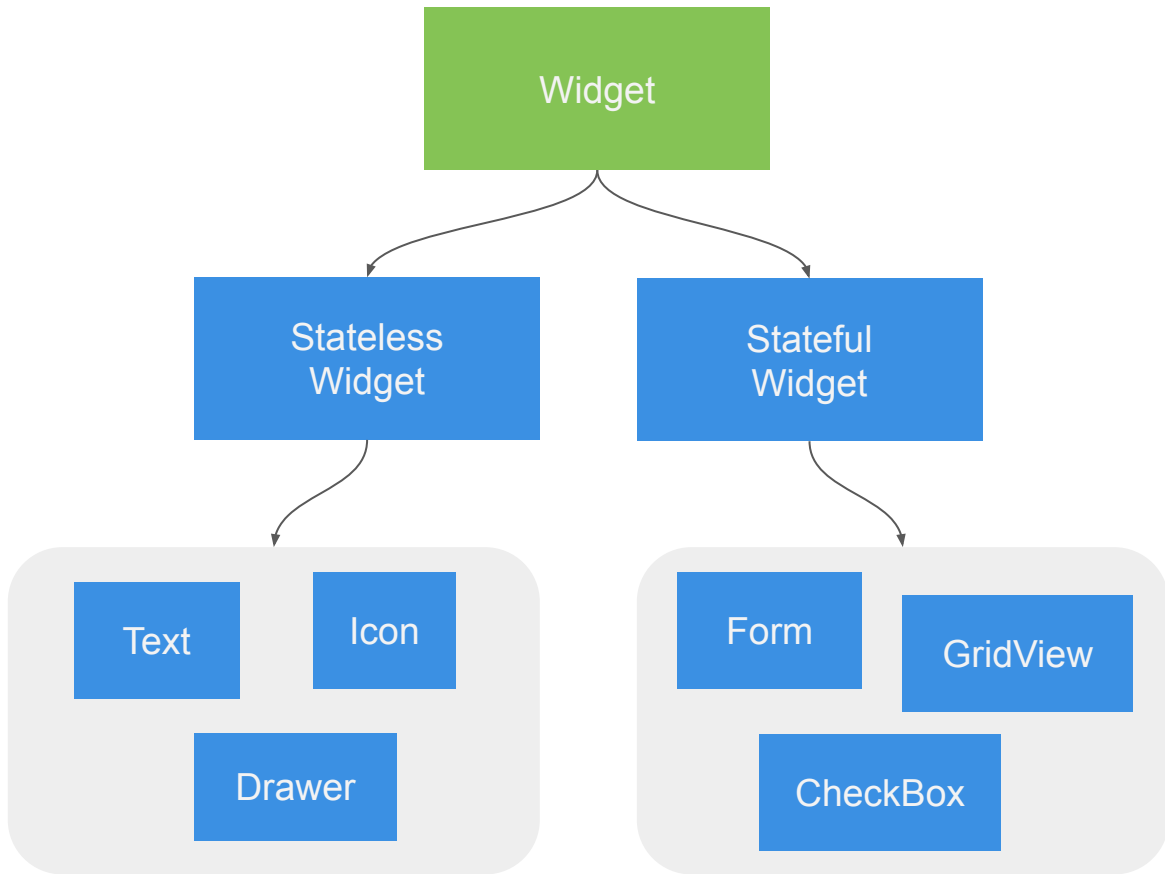
Writing Flutter

Using Flutter widgets

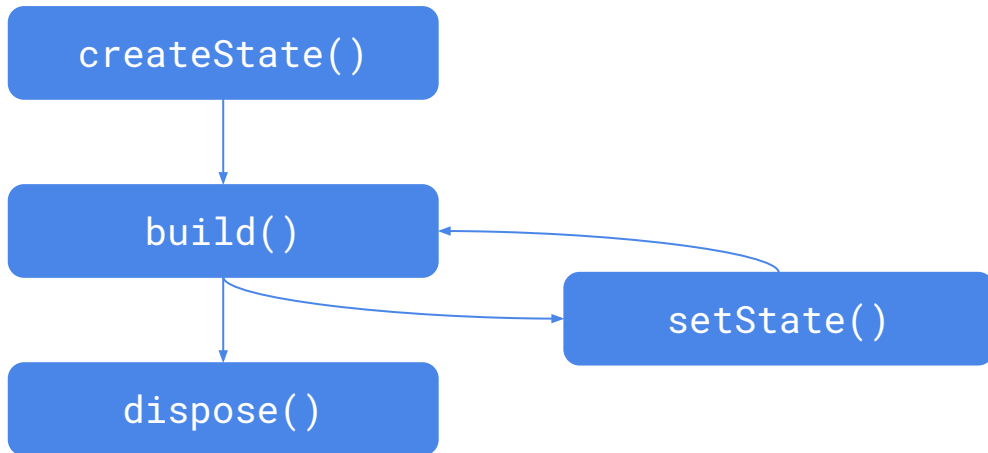
```
Center(  
  child: Container(  
    child: Text(  
      'Hello World',  
      style: TextStyle(  
        color: Colors.white,  
        fontSize: 18,  
      ),  
    ),  
    color: Colors.blue,  
    width: 200,  
    height: 100,  
    alignment: Alignment.center,  
  ),  
);
```



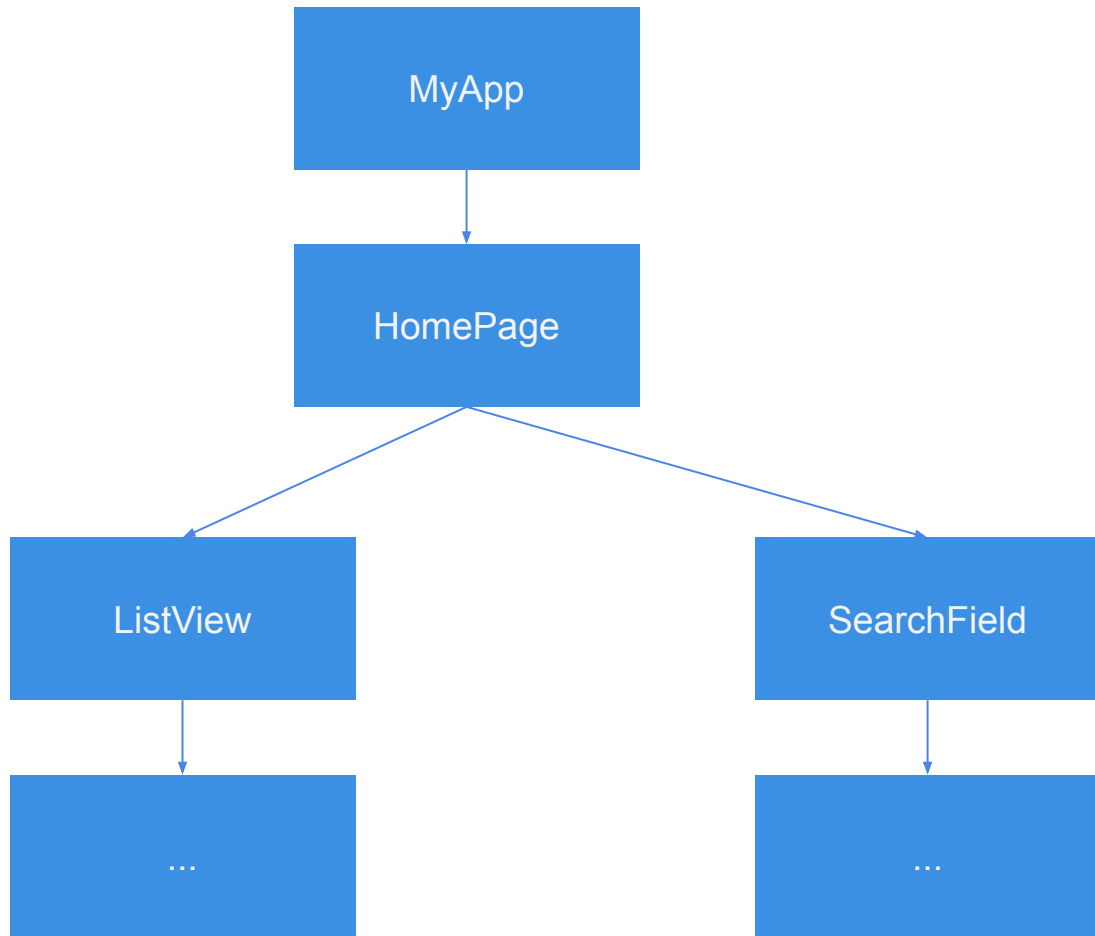
Everything's a Widget



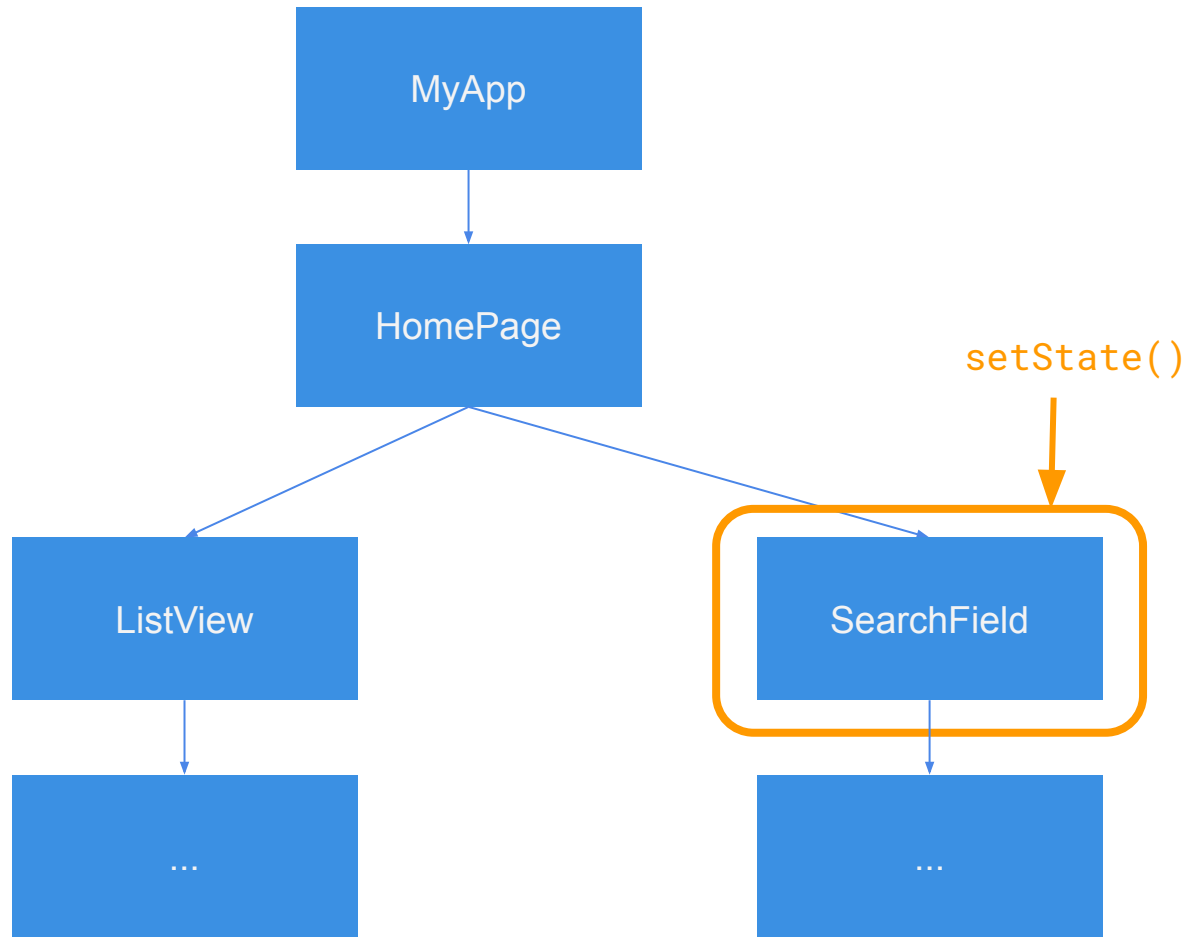
Smart Rendering



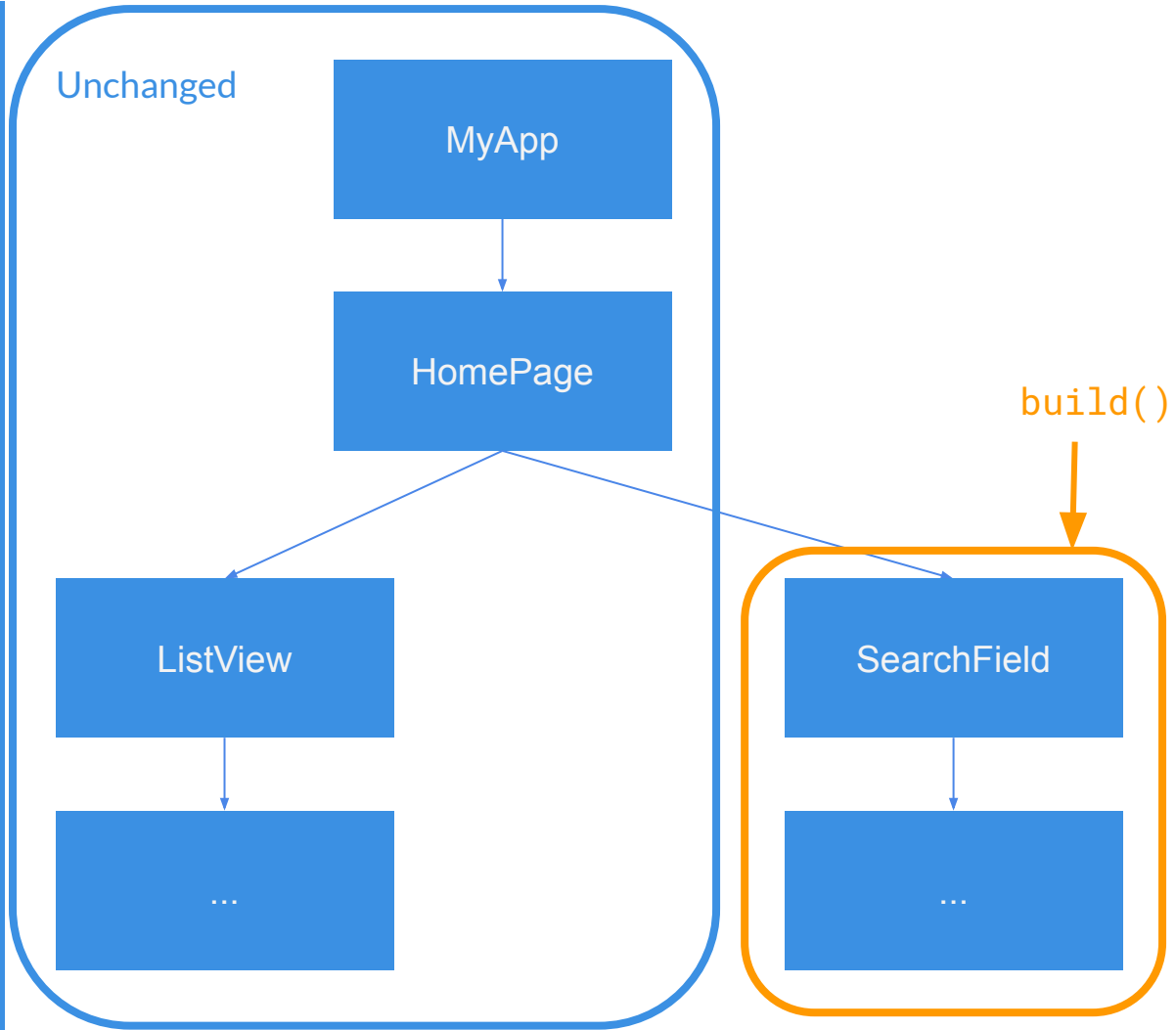
Smart Rendering



Smart Rendering



Smart Rendering



Writing Flutter Bootstrap

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp(title: "Flutter Workshop"));
}
```

Writing Flutter Stateless Widget

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp(title: "Flutter Workshop"));
}

class MyApp extends StatelessWidget {
  final String title;

  MyApp({this.title = ""});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: title,
      home: MyHomePage(),
    );
  }
}
```

Writing Flutter Stateful Widget

```
class MyHomePage extends StatefulWidget {  
  @override  
  _MyHomePageState createState() => _MyHomePageState();  
}  
  
class _MyHomePageState extends State<MyHomePage> {  
  int _counter = 0;  
  
  void _incrementCounter() {  
    setState(() => _counter++);  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Column(  
      children: <Widget>[  
        Text('You have pushed the button $_counter times'),  
        FlatButton(onPressed: _incrementCounter, child: Text('Click me'))  
      ],  
    );  
  }  
}
```

- ❏ Stateful widgets
- ❏ Provider

State management

State management Provider

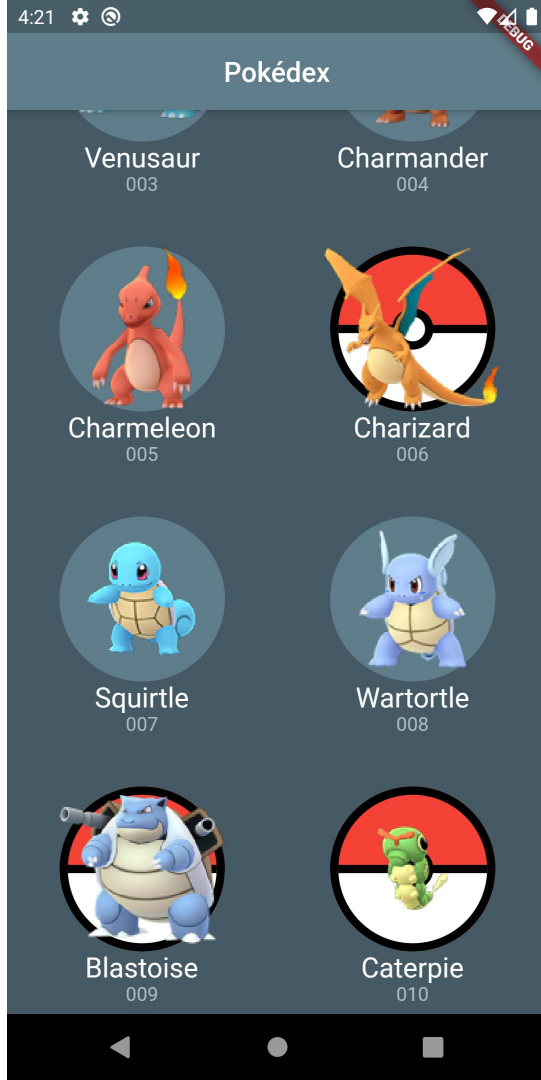
```
class Counter extends ChangeNotifier {  
  int count = 0;  
  
  void increment() {  
    count++;  
    notifyListeners();  
  }  
}  
  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MultiProvider(  
      providers: [  
        ChangeNotifierProvider(create: (_) => Counter())  
      ],  
      child: MaterialApp(  
        title: 'Flutter Demo',  
        home: Scaffold(body: Center(child: MyHomePage()))  
      ),  
    );  
  }  
}
```

State management Provider

```
class MyHomePage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Consumer<Counter>(  
      builder: (context, counter, _) {  
        return Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: [  
            Text("${counter.count}"),  
            FlatButton(  
              child: Text('Increment'),  
              onPressed: () {  
                counter.increment();  
              },  
            )  
          ]  
        );  
      },  
    );  
  }  
}
```

Time to code!

Lets
make
a
Pokédex!



Lets make a Pokédex!

- ❏ Working with lists
- ❏ List / detail views
- ❏ Hero animation
- ❏ Fetching data - json

github.com/rsr-itminds/flutter-workshop