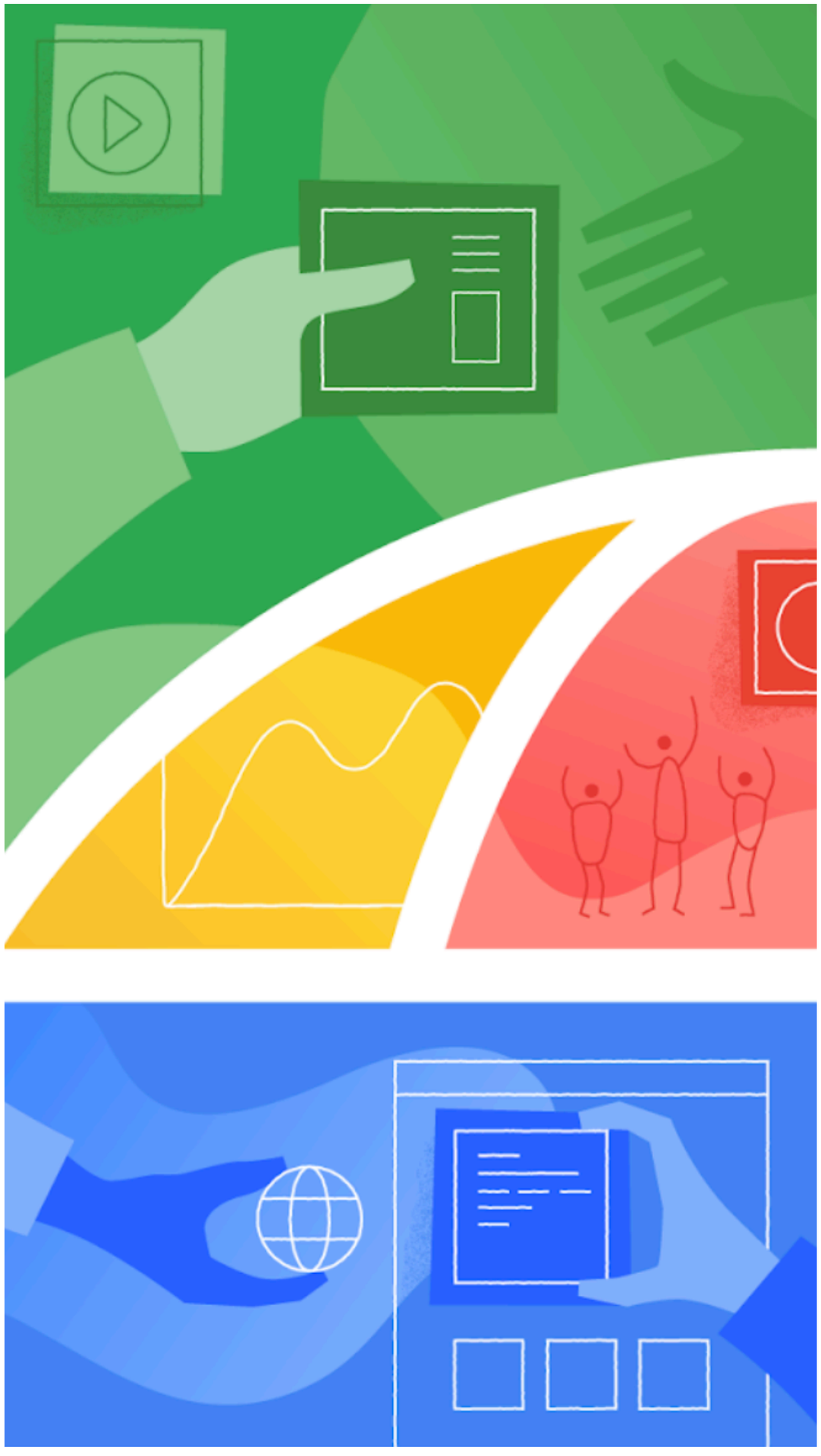


An Introduction to Flutter

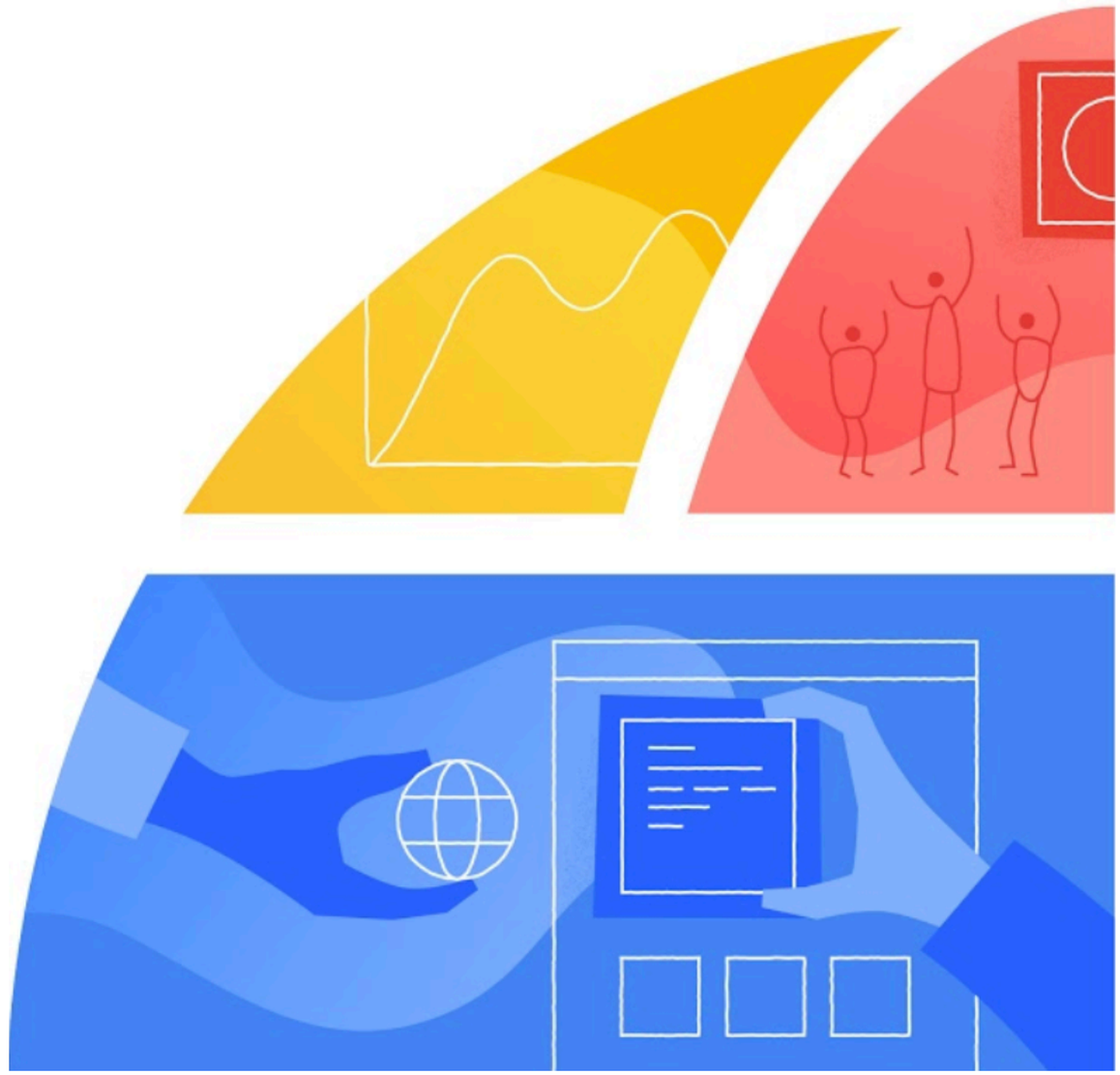
Don Ward
@donwardpeng

Who am I?

- Led Quicken Loans Mobile Team for a few years
- Founder and Co-Lead GDG Detroit for 7 years
- Founder and Leader of GDG Windsor ~ 1 year

The Agenda

1. What is Flutter?
2. What is Dart?
3. Why?
4. Let’s code!

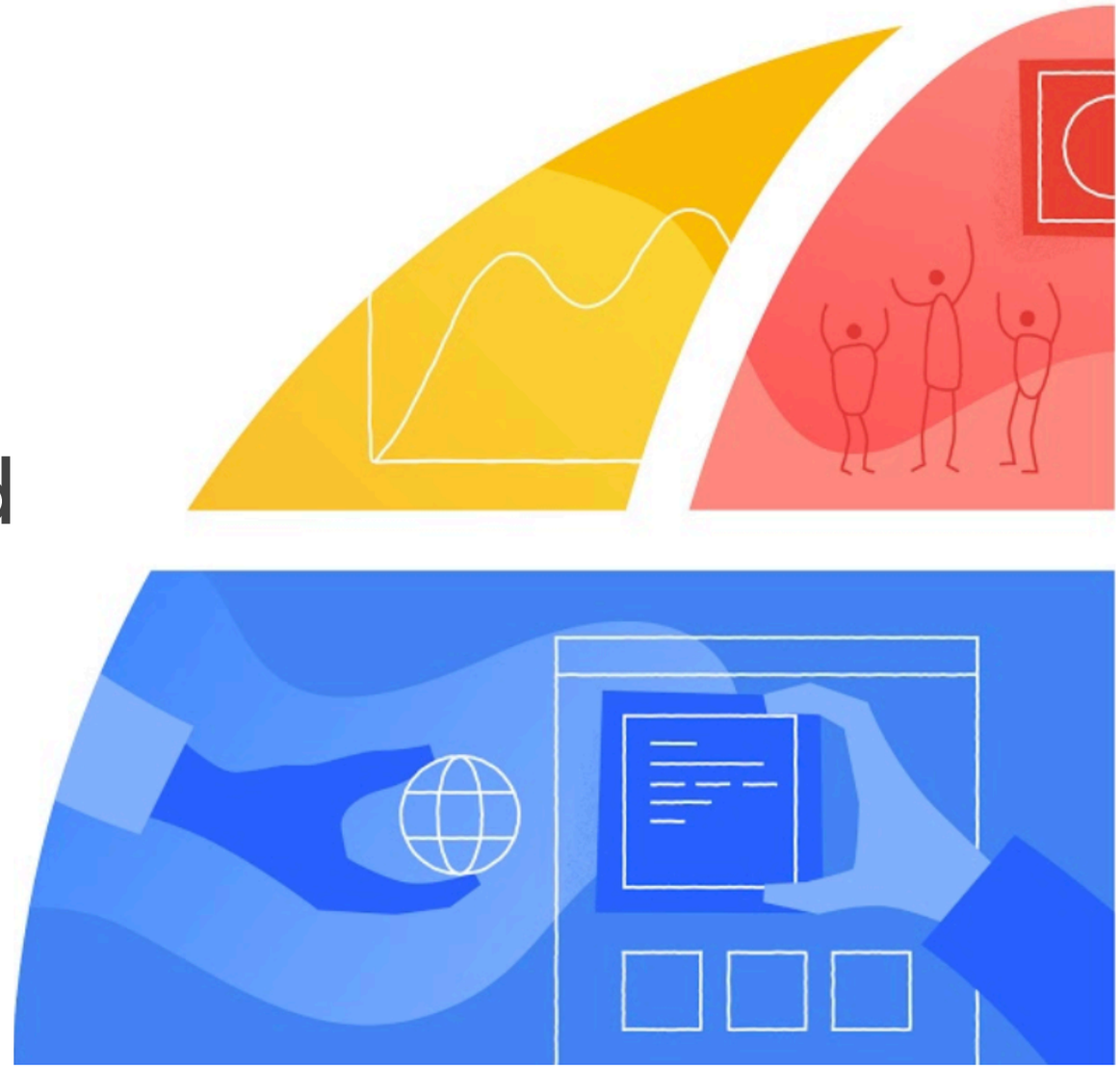


Flutter

What is Flutter?

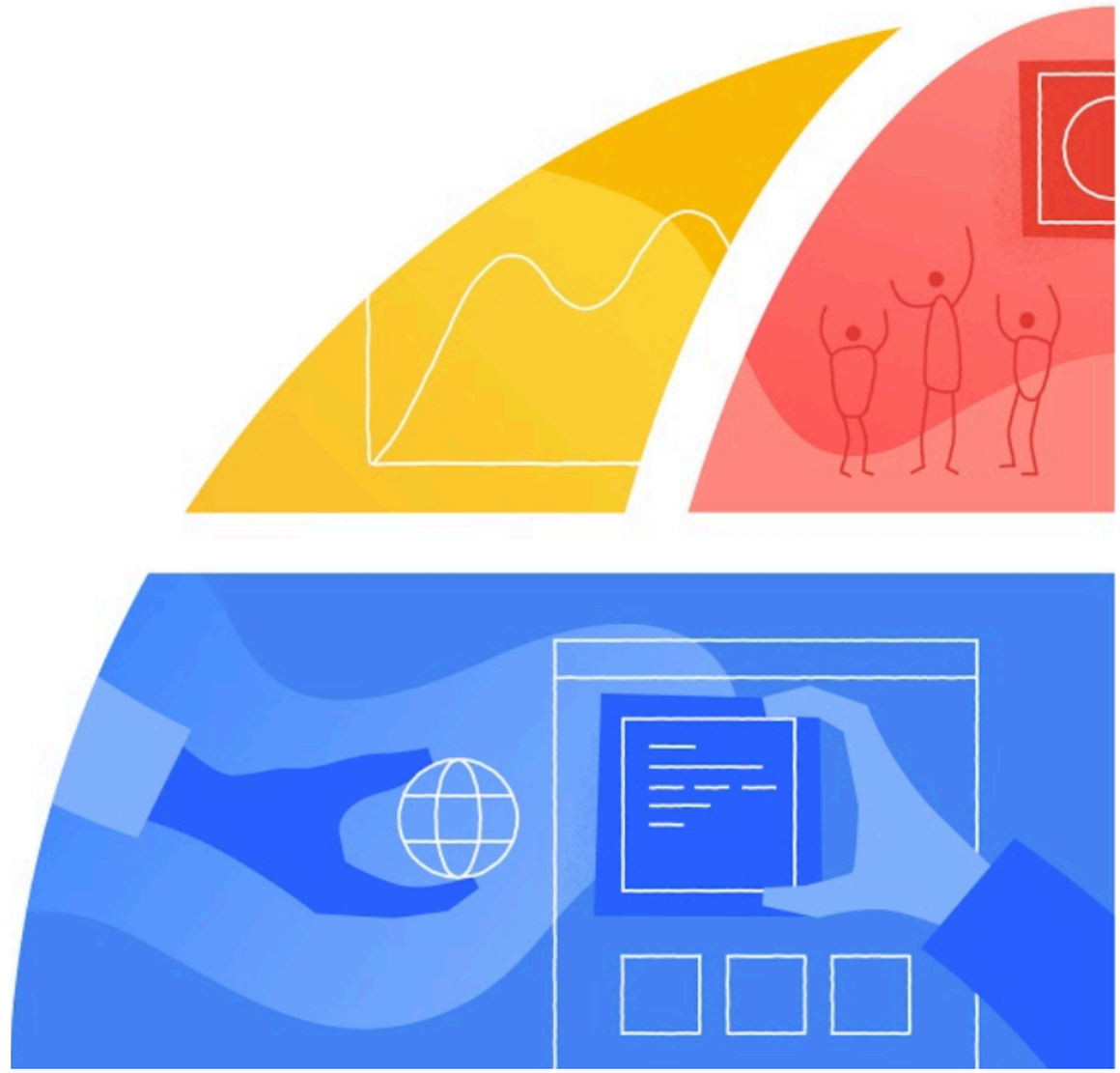
- Open Source Cross Platform Mobile Development
- Announced 2015
- Flutter 1.0 – Dec 2018
- Flutter 1.5 – May 2019
- Available on iOS, Android ChromeOS

Flutter

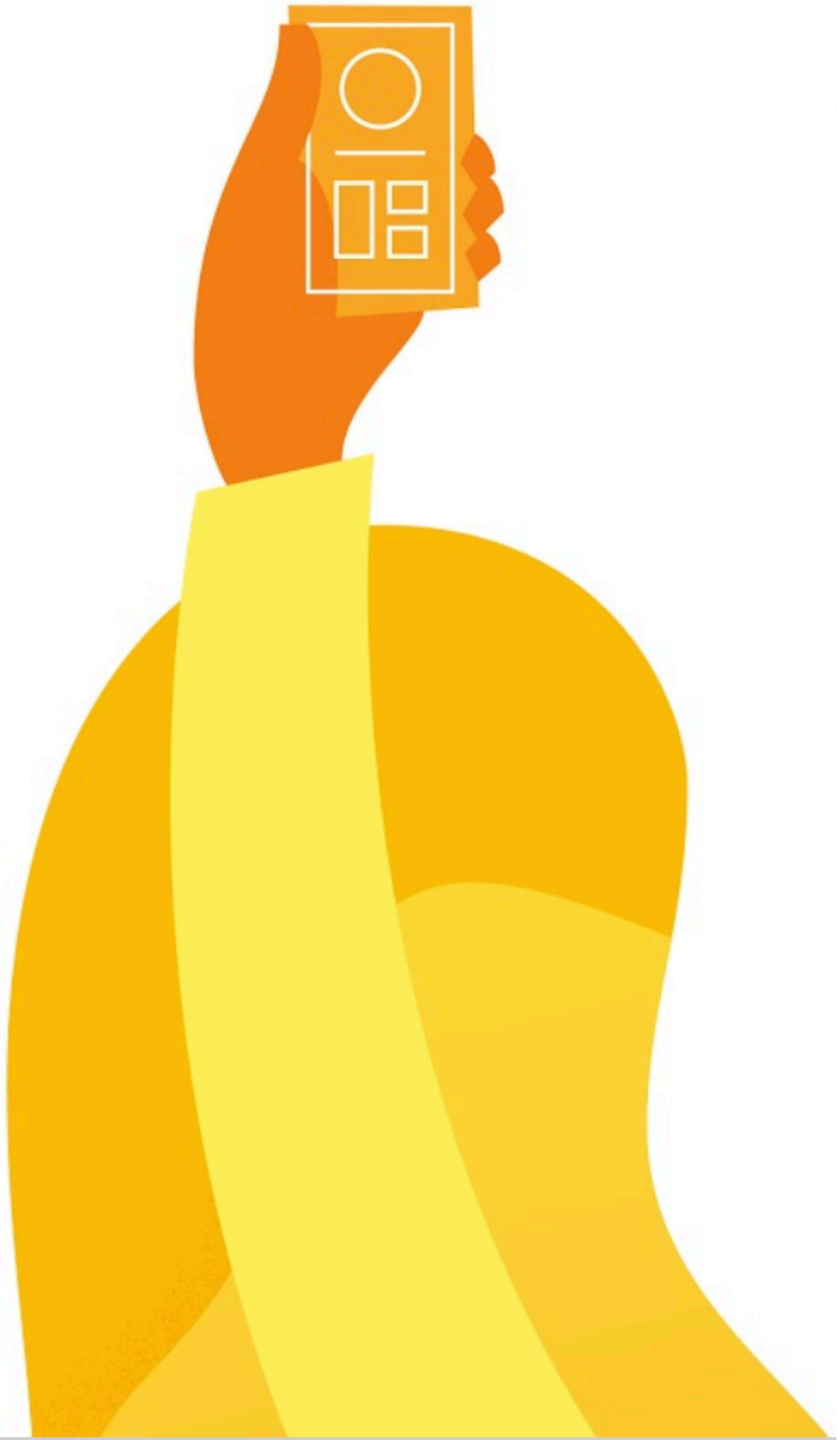


What is Dart?

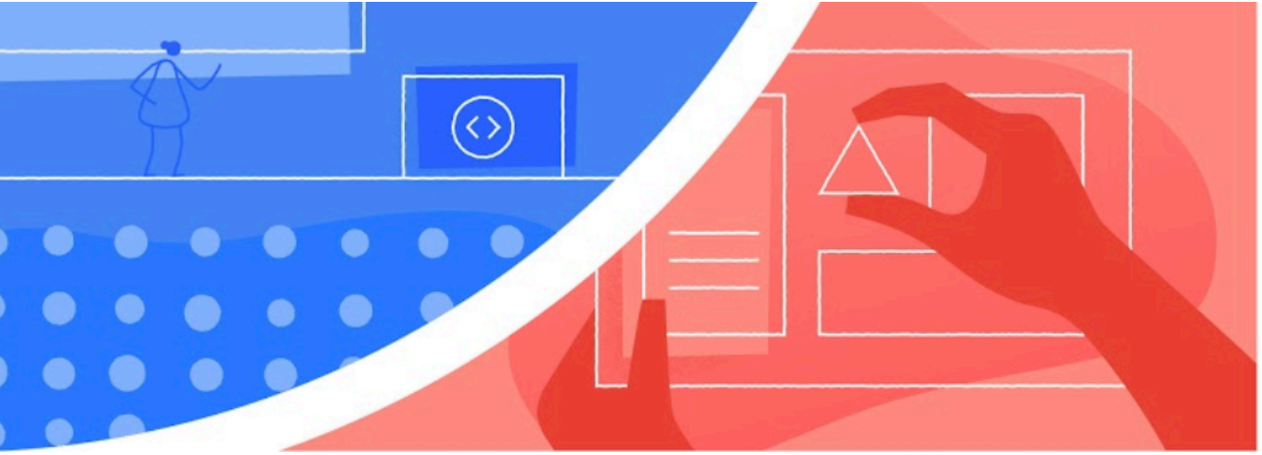
- Since 2011
- Statically typed language similar to JavaScript and
- Object-oriented
- Syntactic sugar

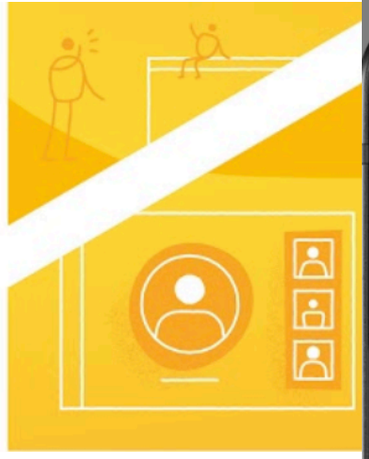


Dart

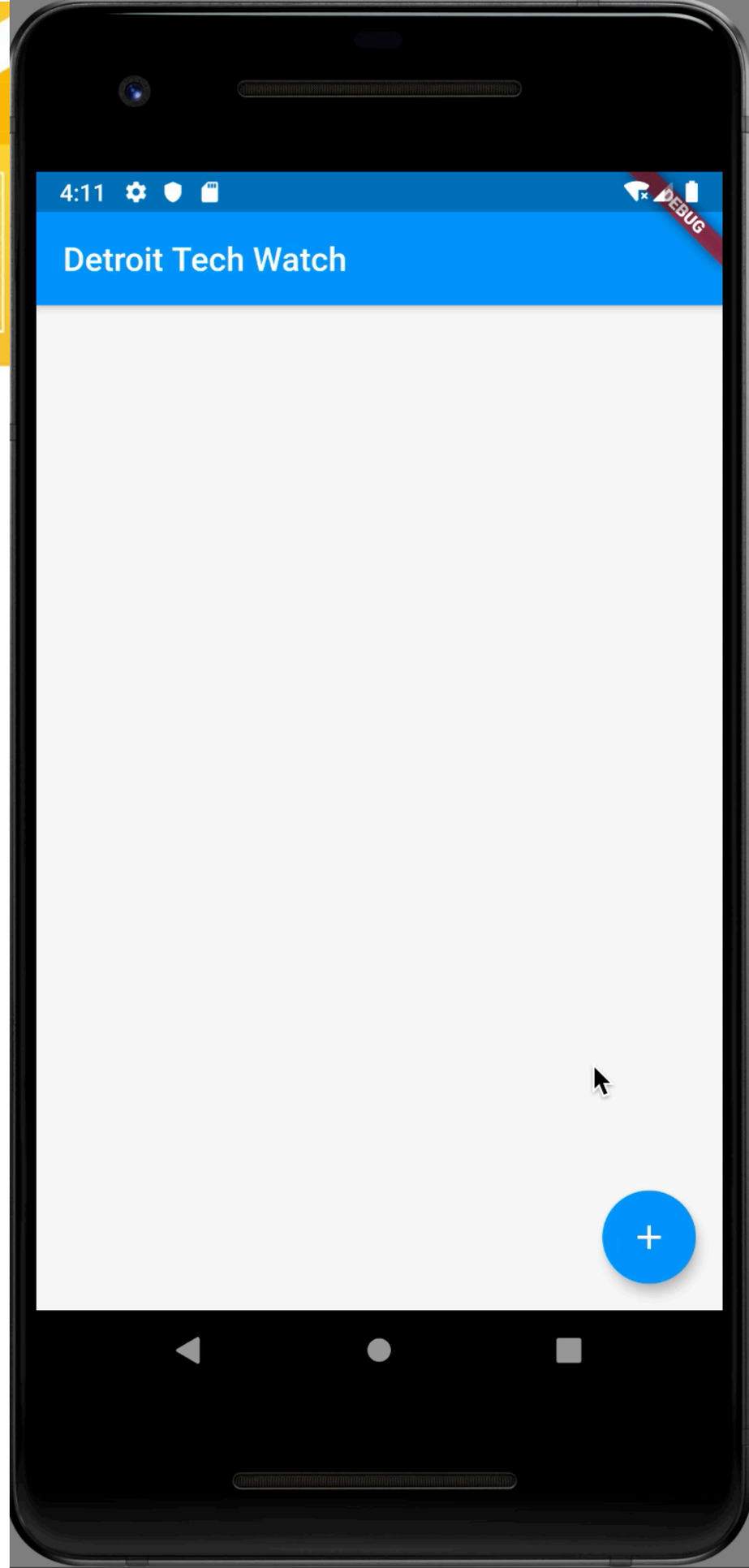


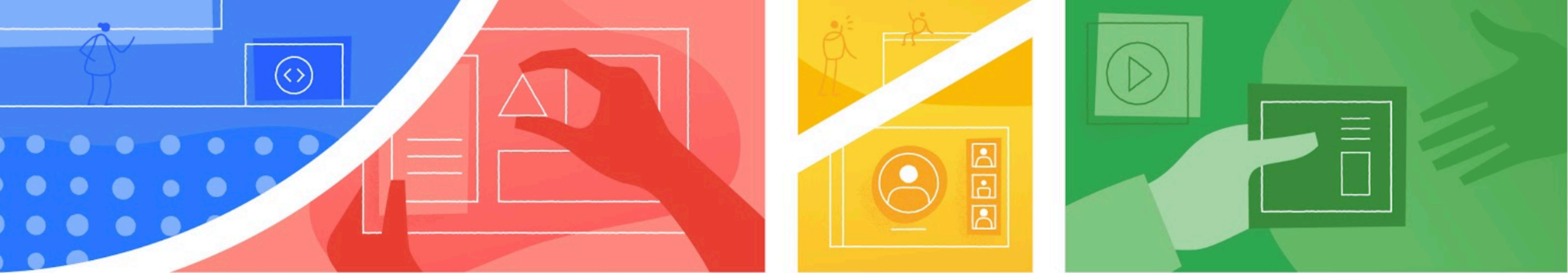
‘Why?’





What are we building today?





Let’s start

1. Flutter SDK installed
2. Visual Studio Code installed
3. Android Emulator or iOS Simulator running

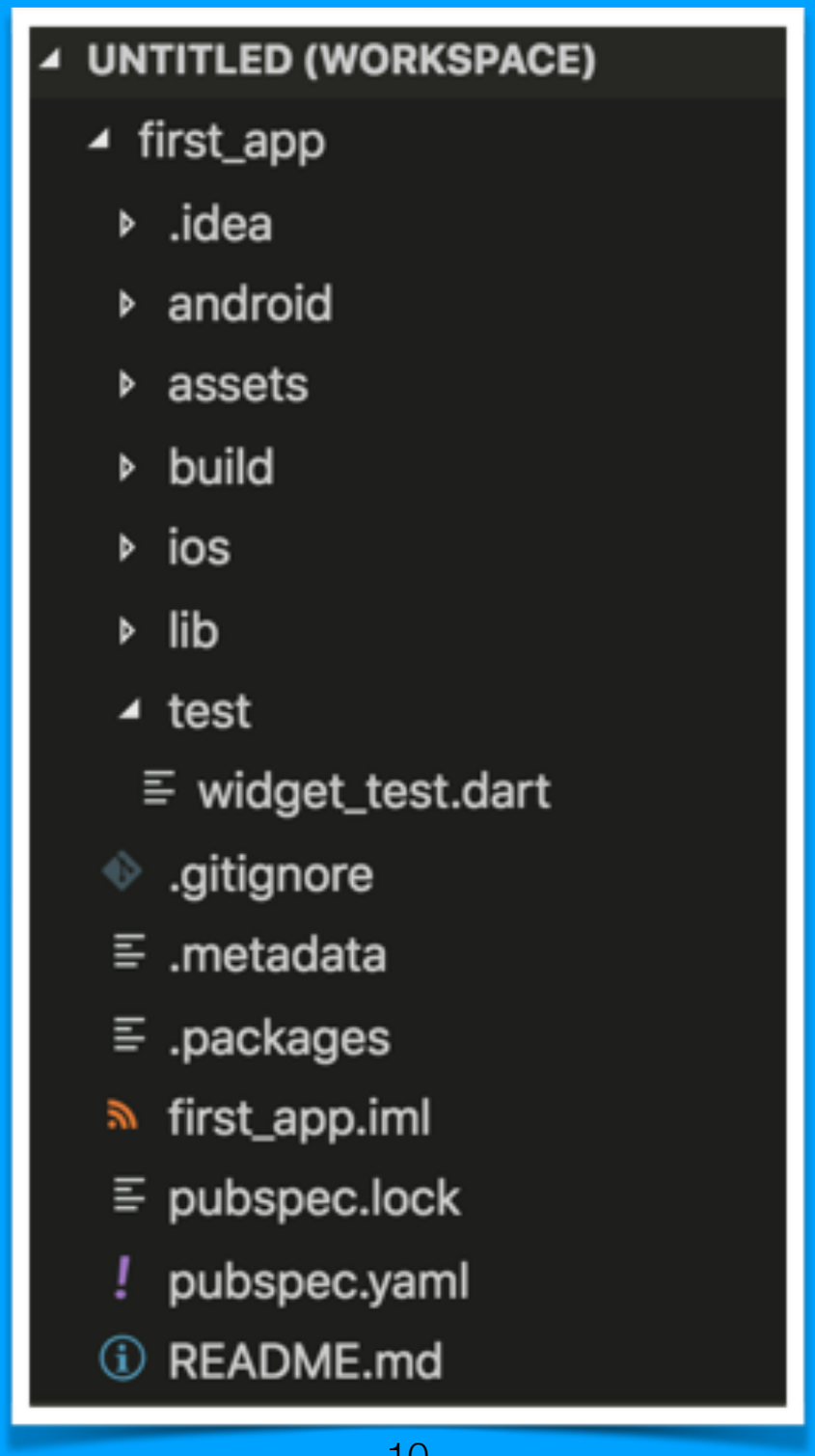
https://github.com/donwardpeng/Flutter-DetTechWatch

Let’s talk about the Command Line Tool (CLI)

- ‘flutter doctor’
- ‘flutter create <app_name>’
- In the <app_name> directory -> ’flutter run’
- Hot Reload and Restart
- While running, type ‘h’

Flutter

The Project Structure



Let’s start from scratch

1. In VSCode, open the main.dart file
2. Wipe it clean
3. Add the code below

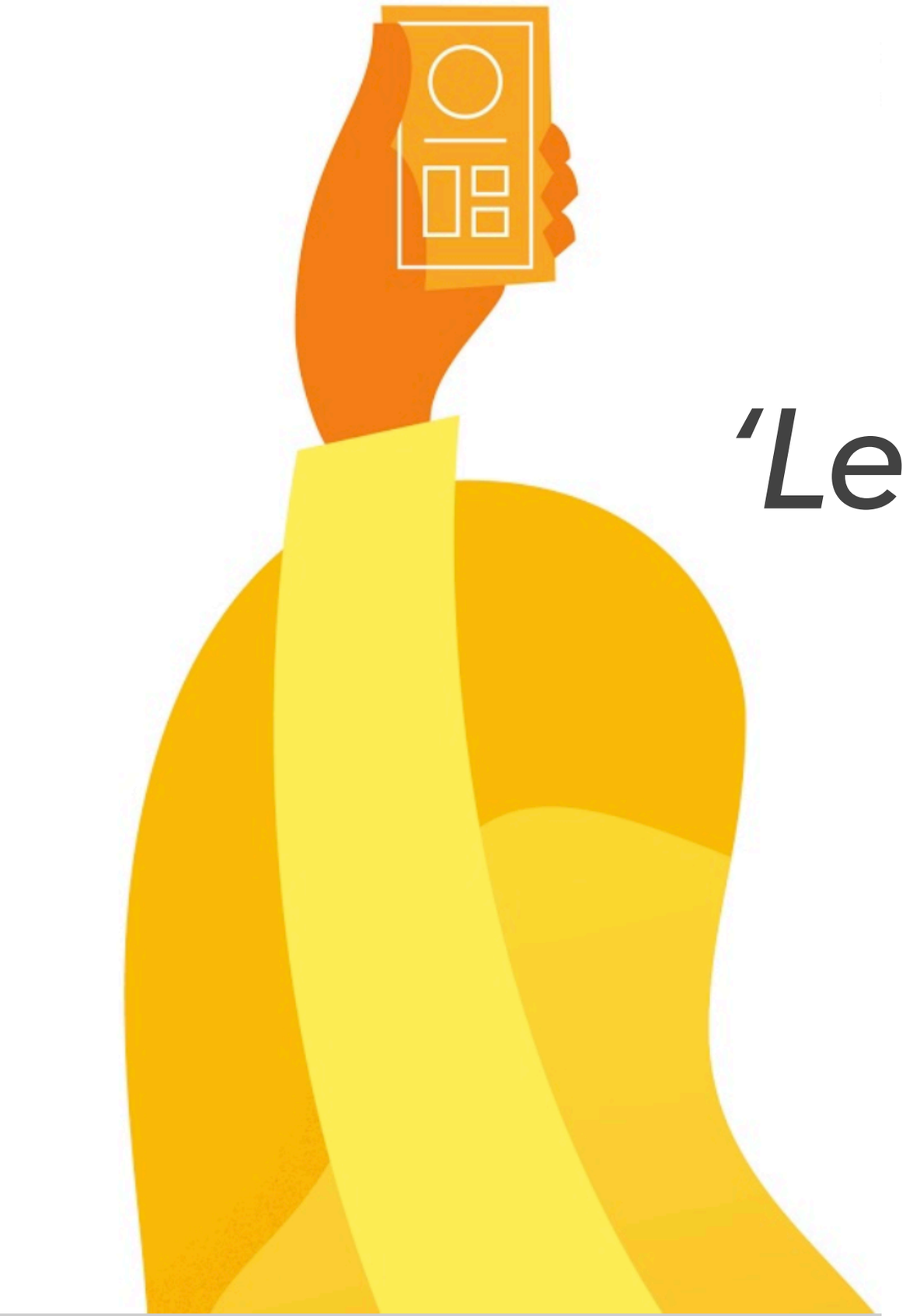
```
//Example 1
import 'package:flutter/material.dart';

void main() {}
```

Let’s attach something to the screen

The runApp method

- In Dart, '=> expr' is equivalent to {return expr};

```
//Example 2
import 'package:flutter/material.dart';

void main() => runApp(<Widget>);
```



'Let's talk about Widgets'

Widgets

- Everything is a Widget
- Stateless - display only
- Stateful - maintains some internal state

```
class StatelessWidgetExample extends StatelessWidget{};

class StateFullWidgetExample extends StatefulWidget{};
```

Let’s create our first Widget

```dart
//Example 3
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget{
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return null;
  }
}
```

Let’s add some Material Design

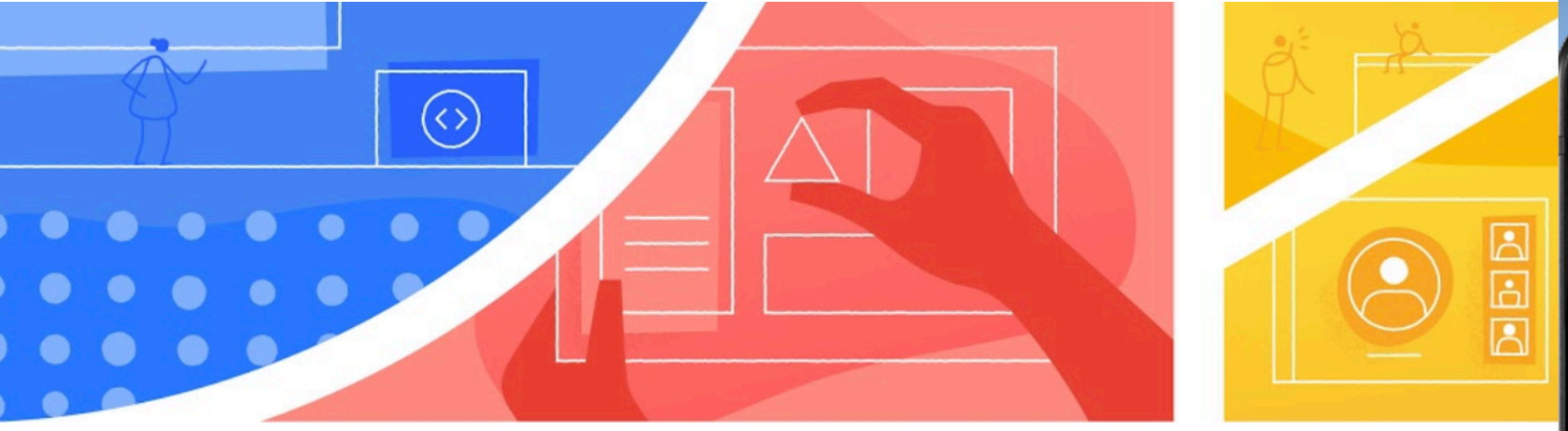
```
//Example 4
 @override
  Widget build(BuildContext context) {
     return MaterialApp(
       title: ‘Tech Watch Detroit’,
     );
   }
}
```

'Let's talk about Widgets and Constructors'

Widgets and Constructors

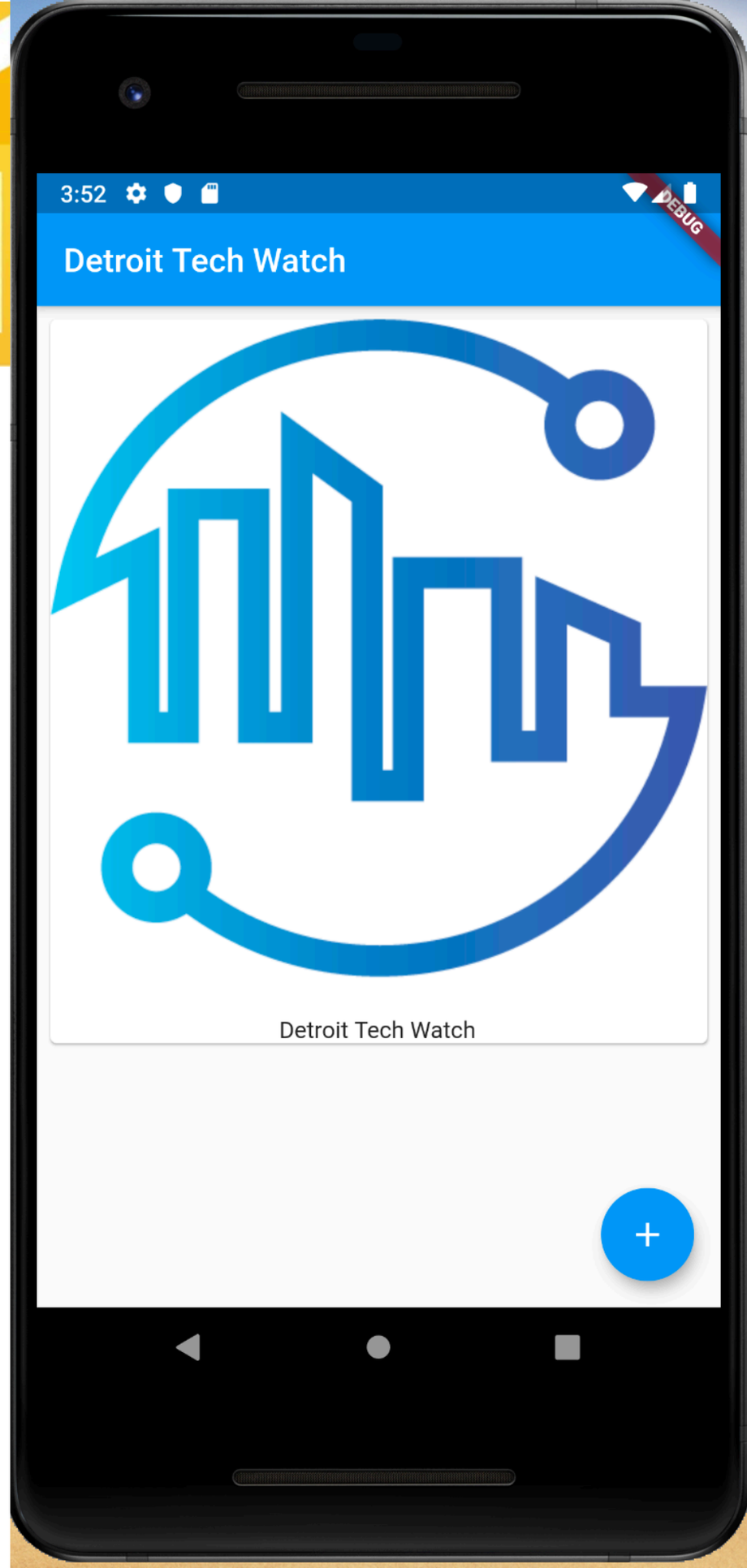
- Positional and Named Parameters
- Flutter -> heavily relies on Named Parameters
- Check out the Text Widget

```
Text(String data,
{ Key key, TextStyle style, StrutStyle strutStyle
, TextAlign textAlign, TextDirection textDirectio
n, Locale locale, bool softWrap, TextOverflow ove
rflow, double textScaleFactor, int maxLines, Stri
ng semanticsLabel })
```



What do we do first?

- Add a Card per Image
- Add a Floating Action Button



Add a Scaffold for the Screen

```
//Example 5
    return MaterialApp(
        title: 'Detroit Tech Watch',
        home: Scaffold(
          appBar: AppBar(
            title: Text('Detroit Tech Watch'),
          ),
        ));
```

Add a Card to the Body

```
//Example 6
  home: Scaffold(
   appBar: AppBar(title:Text('Detroit Tech Watch'),),
     body: Card(child: ,));
   }
}
```

Add a Column to the Card

```
//Example 7
        home: Scaffold(
            appBar: AppBar(
                title: Text('Detroit Tech Watch'),
                body: Card(child: Column(children:
<Widget>[],),
                ),
            ));
    }
}
```

Let’s add some images to the project

Steps

- Create a new assets folder and add images
- Modify pubspec.yaml file

GDG Detroit

Add a Image to the Card

```
//Example 8
            body: Card(
                child: Column(
                    children: <Widget>[
                      Image.asset('assets/dtw.png'),
                    ],
                ),
            )));
    }
}
```

Add some padding under the image

```
//Example 9
<code above here>
                    Padding(
                      padding: const EdgeInsets.all(8),
                    ),
                  ],
                ),
              ));
<code below here>
```

Add some margin around the image in the card

```
//Example 10
<code above here>
    body: Card(
              margin:EdgeInsets.all(8),
              child: Column(
                children: <Widget>[
                  Image.asset('assets/dtw.png'),
                 <code below here>
```

Wrap it in a ListView

```
//Example 11
<code above here>
            body: ListView(
                children: <Widget>[
                  Card(
                    margin: EdgeInsets.all(8),
                    child: Column(

<code below here>
```

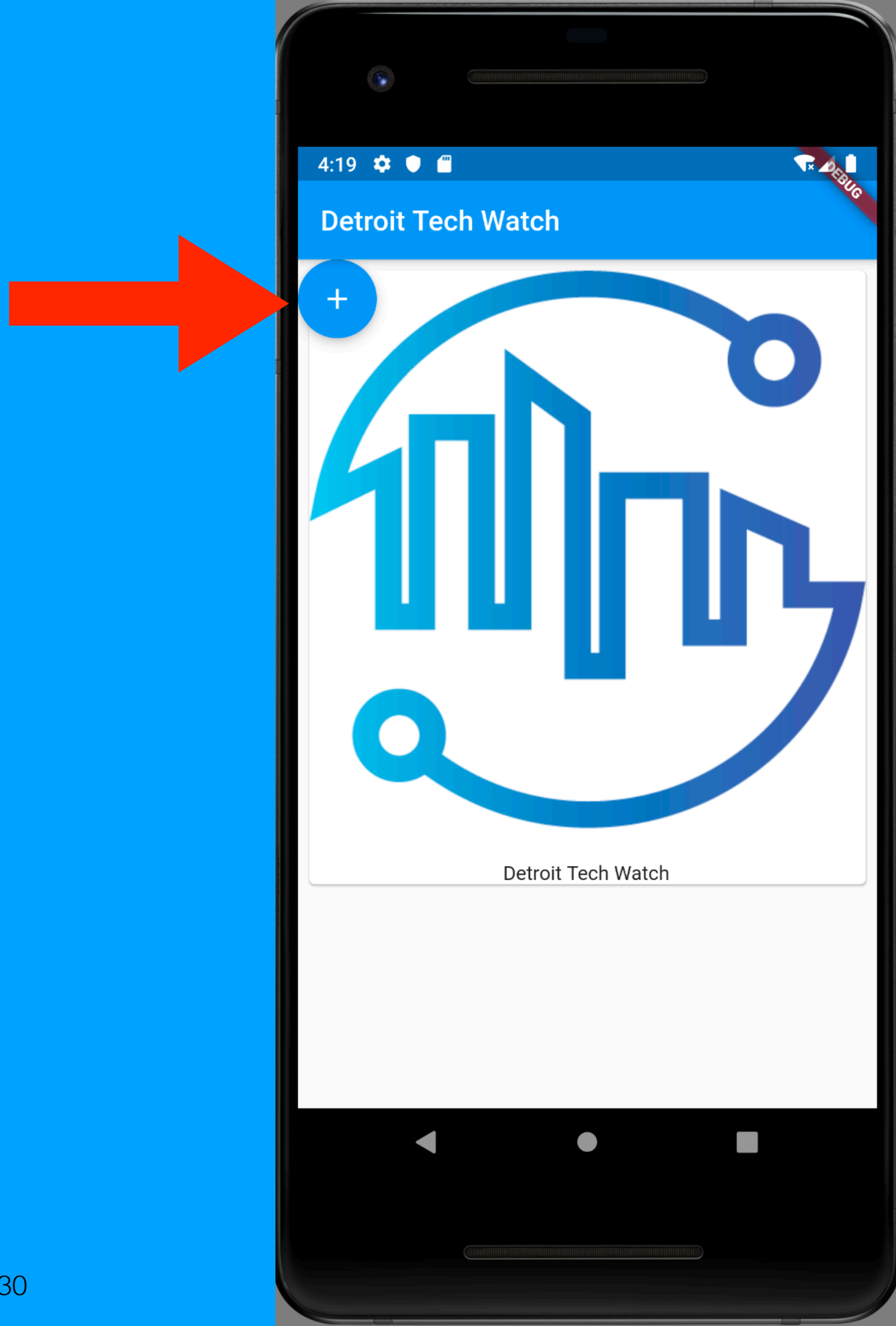
Wrap it in a Stack

```
//Example 12
<code above here>
            home: Scaffold(
              appBar: AppBar(
                title: Text('Windsor-Essex DevFest'),
              ),
              body: Stack(
                children: <Widget>[
                  ListView(
                    children: <Widget>[


<code below here>
```

Let's add a FAB

```
//Example 13
<Cardview code above here>
                    ),
                    FloatingActionButton(
                      child: const Icon(Icons.add),
                      onPressed: () {},
                    )
                  ],
                ));


<code below here>
```

What happened to the Floating Action Button?



Let’s move the Floating Action Button

```
//Example 14
<Cardview code above here>
                Positioned(
                    bottom: 16.0,
                    right: 16.0,
                    child: FloatingActionButton(
                      child: const Icon(Icons.add),
                      onPressed: () {},
                    ))
              ],
            ));
<code below here>
```

Everything is in place now – let's wire up the button and images to the list

WARNING: BOILERPLATE CODE AHEAD

Let's add some constants for the Images

```
//Example 15
class MyApp extends StatelessWidget {

// Declare a set list of images to cycle through
  final List<Image> _itemImages = [
    Image.asset('assets/Google.png'),
    Image.asset('assets/dtw.png'),
    Image.asset('assets/GDGDetroit.png')
  ];

int _index = 0;
List<Image> _listOfImagesForScreen = [];
```

Let’s add a button pressed handler

```
//Example 16
//Add to the main class above the build method
void onButtonPressed() {
//    setState(() {
      _index++;
      if (_index >= _itemImages.length) {
        _index = 0;
      }
      _listOfImagesForScreen.add(_itemImages[_index]);
    print('Item Name = ' +
_listOfImagesForScreen.toString());
    }
```

Boiler Plate Code Alert – Making a class Stateful

```
//Example 17

class MyApp extends StatefulWidget {
  @override
  State<StatefulWidget> createState() {
    return _MyAppState();
  }
}

class _MyAppState extends State<MyApp> {
```

Redraw the Listview with all of the Images every time state changes

```
//Example 18

ListView(
                children: _listOfImagesForScreen.map((element)
```

What did we learn?

- Command Line Tool for Flutter
- Project Structure
- Widgets
- Constructors and Named Parameters
- Stateless and Stateful Widgets

https://github.com/donwardpeng/Flutter-DetTechWatch

What questions do you have for me?

Don Ward
@donwardpeng