

Exploring new possibilities with Flutter for web

By Pawan Kumar



Google Developer Expert
For Flutter, Firebase,
Dart & WebTech

Back in 2017

Me

Native
Android





In 2018
Declarative UI Pattern?



```
// Imperative style  
b.setColor(red)  
b.clearChildren()  
ViewC c3 = new ViewC(...)  
b.add(c3)
```



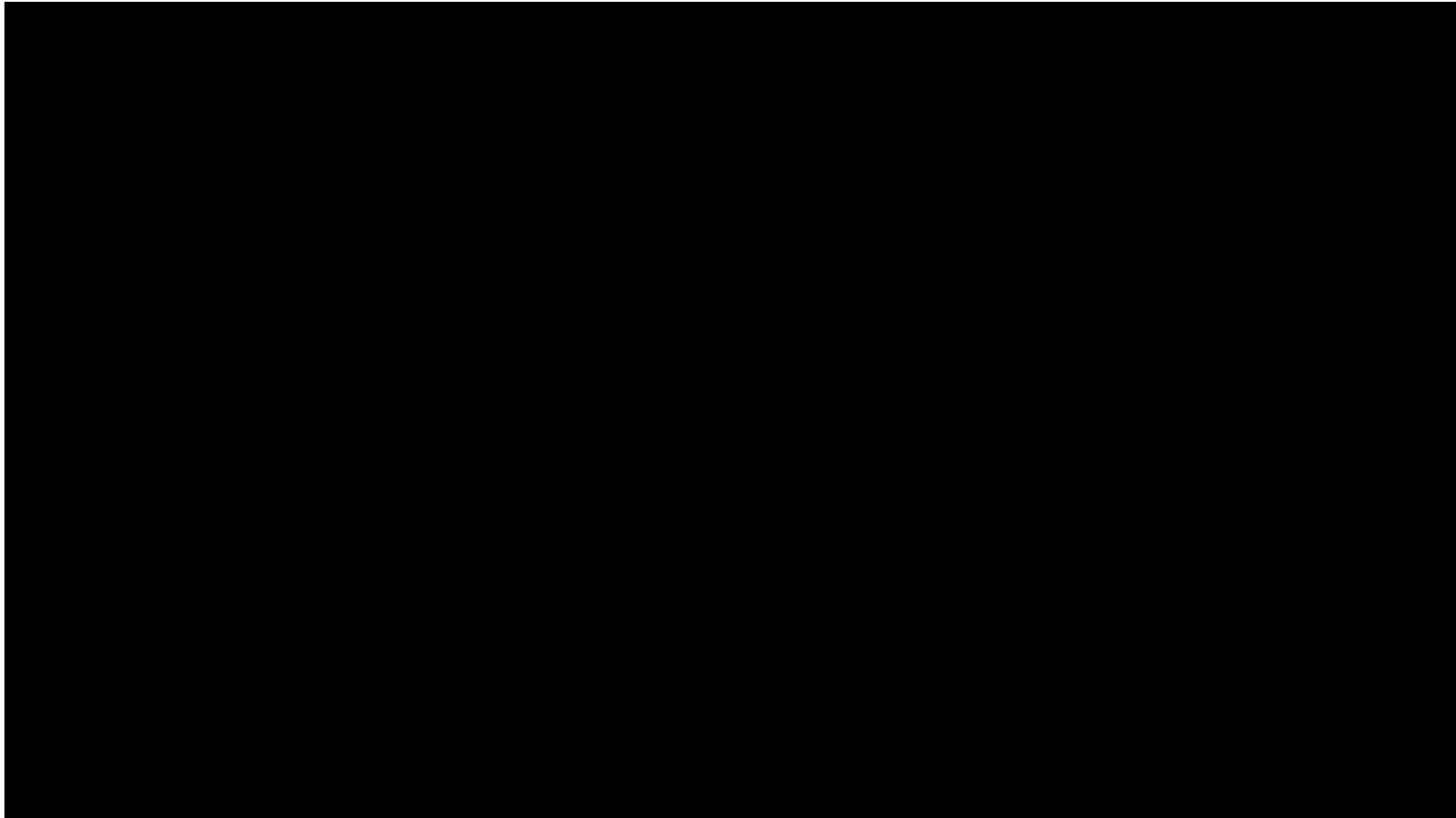
```
// Declarative style  
return ViewB(  
    color: red,  
    child: ViewC(...),  
)
```



In **2019**?

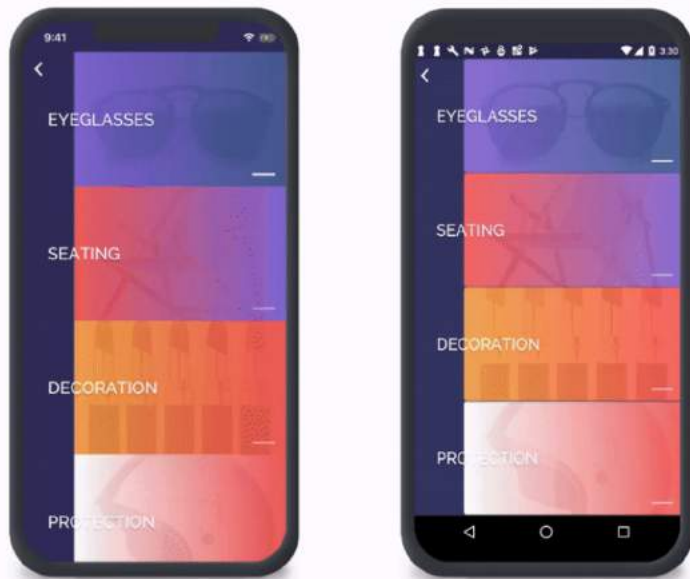


2019 for declarative UI Pattern



What is Flutter

Flutter is Google's portable UI toolkit for building beautiful, native applications for Android, iOS, Web and Desktop from a single codebase.



Expressive, beautiful UIs

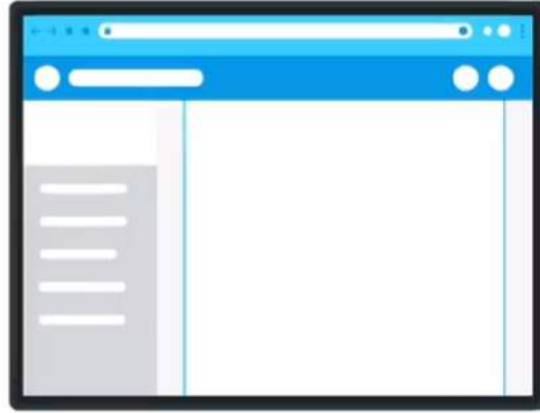
Stateful hot reload

Native performance

Flutter directly compiles your dart code to native machine code



Mobile



Web



Desktop

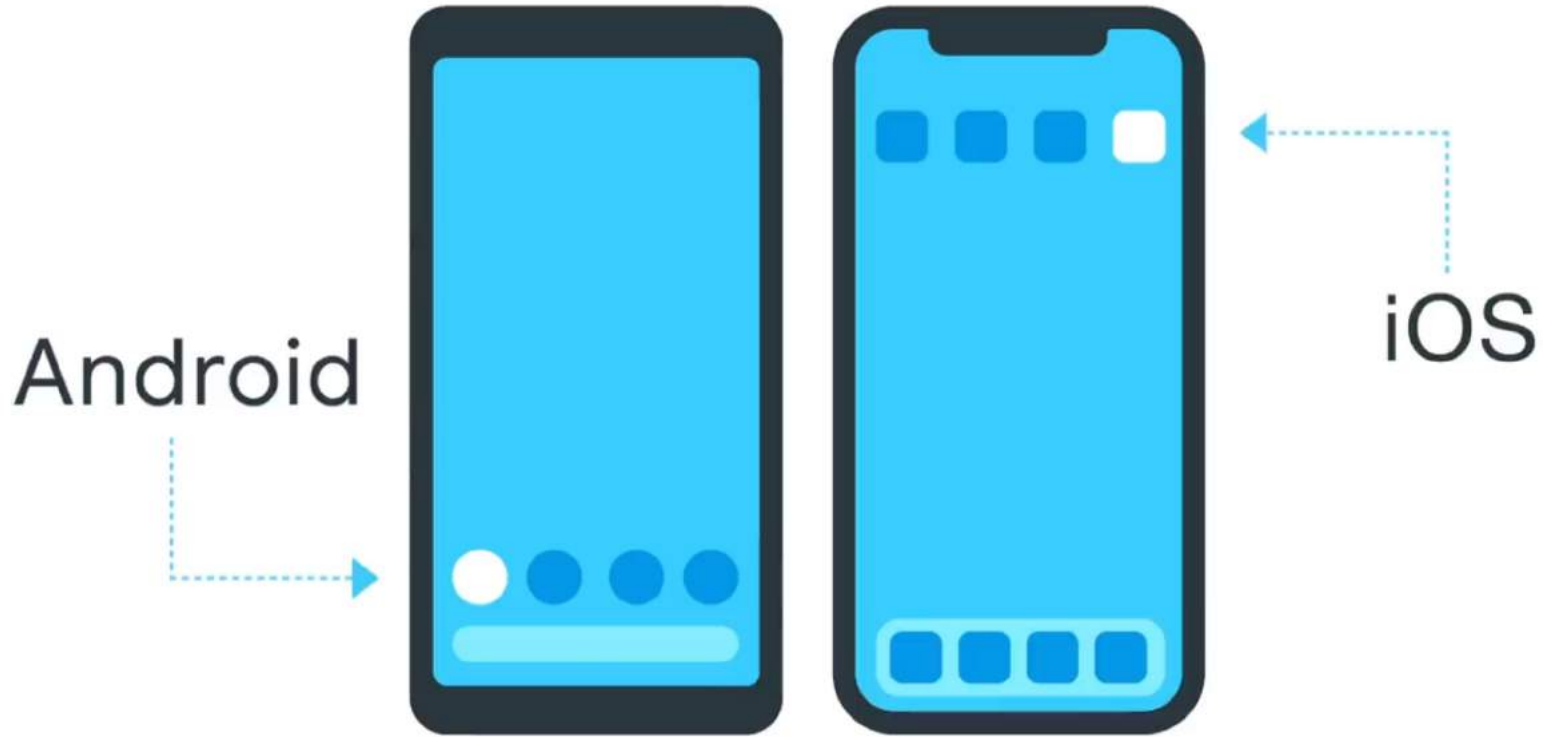


Raspberry Pi

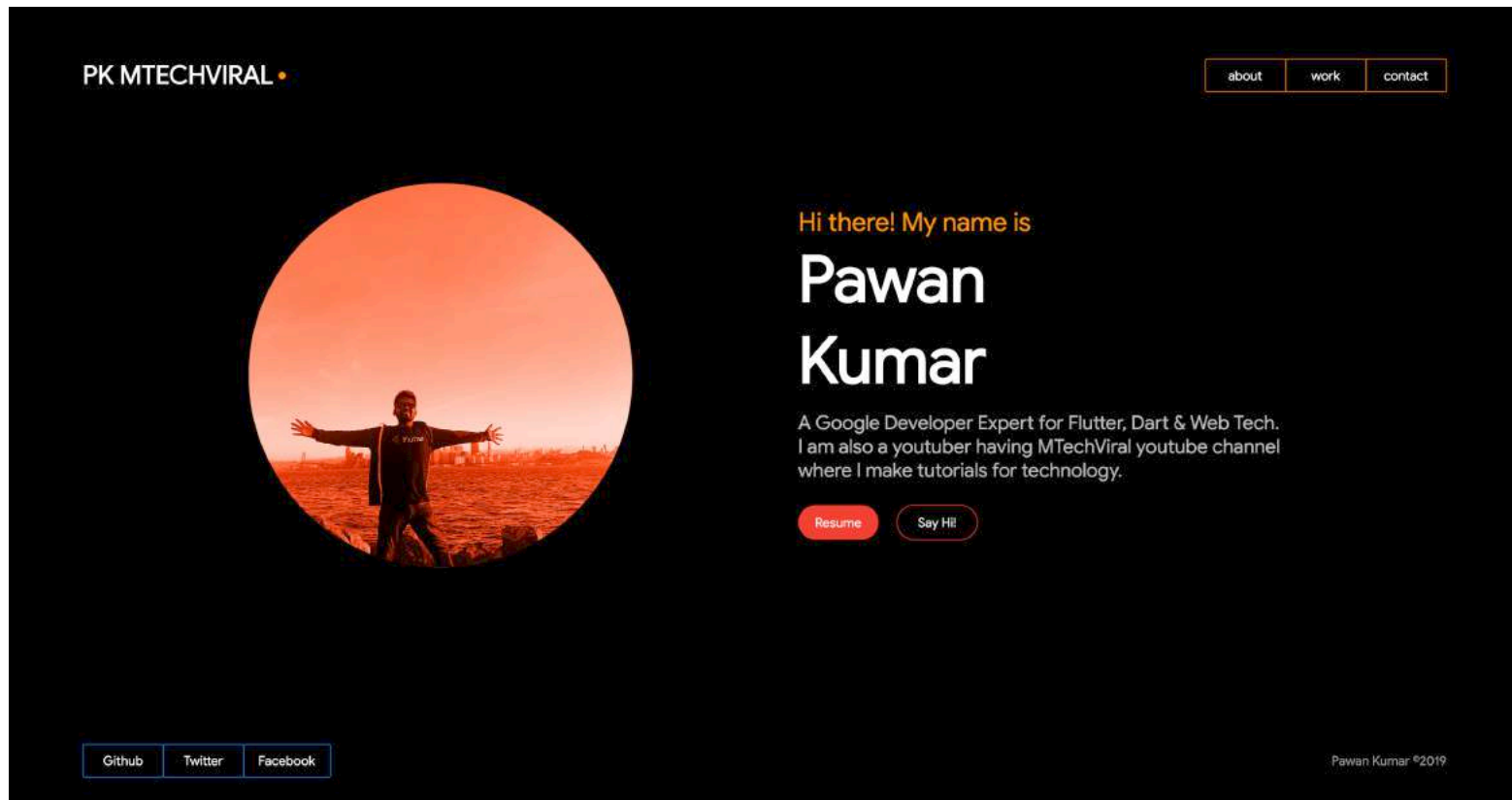


Desktop
Embedding

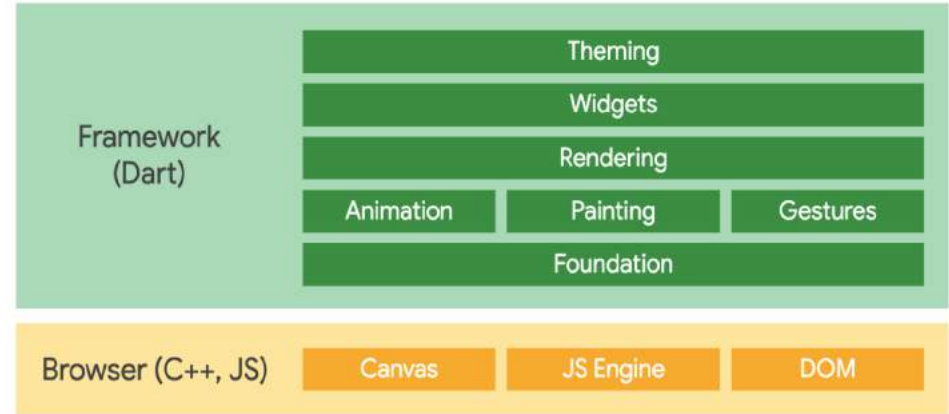
Integrate flutter app to existing iOS and Android app



Flutter For Web



Flutter vs Flutter For Web



Scenario

- A connected [Progressive Web Application](#) built with Flutter
- Embedded interactive content
- Embedding dynamic content in a Flutter mobile app

Flutter Web Support

Flutter 1.10



Cross Platform Mobile Apps



Dart

“Dart is particularly easy to learn because it has features that are familiar to users of both static and dynamic languages.”

Why Flutter uses Dart



Event loop





Just-in-time compilation





Ahead-of-time compilation





Type safe



TS



```
class Pizza {
```

```
}
```



```
class Pizza {  
  slices: number = 8;  
  
}
```



```
class Pizza {  
    slices: number = 8;  
  
    constructor(public toppings: string[]) {}  
  
}
```



```
class Pizza {  
  slices: number = 8;  
  
  constructor(public toppings: string[]) {}  
  
  async bake() {  
    return new Promise(resolve => setTimeout(resolve, 5 * 1000));  
  }  
}
```




```
class Pizza {  
    slices: number = 8;  
  
    constructor(public toppings: string[]) {}  
  
    async bake() {  
        return new Promise(resolve => setTimeout(resolve, 5 * 1000));  
    }  
}  
  
async function main() {  
  
}  
  
main();
```



```
class Pizza {  
    slices: number = 8;  
  
    constructor(public toppings: string[]) {}  
  
    async bake() {  
        return new Promise(resolve => setTimeout(resolve, 5 * 1000));  
    }  
}  
  
async function main() {  
    const dinner = new Pizza(['pineapple']);  
  
}  
  
main();
```



```
class Pizza {  
    slices: number = 8;  
  
    constructor(public toppings: string[]) {}  
  
    async bake() {  
        return new Promise(resolve => setTimeout(resolve, 5 * 1000));  
    }  
}  
  
async function main() {  
    const dinner = new Pizza(['pineapple']);  
    await dinner.bake();  
}  
  
main();
```


TS

```
class Pizza {  
  slices: number = 8;  
  
  constructor(public toppings: string[]) {}  
  
  async bake() {  
    return new Promise(resolve => setTimeout(resolve, 5 * 1000));  
  }  
}  
  
async function main() {  
  const dinner = new Pizza(['pineapple']);  
  await dinner.bake();  
  console.log(`yum! ${dinner.toppings.join(', ')}`);  
}  
  
main();
```



Dart



```
class Pizza {
```

```
}
```



```
class Pizza {  
    int slices = 8;  
    List<String> toppings;  
  
}
```



```
class Pizza {  
    int slices = 8;  
    List<String> toppings;  
  
    Pizza(this.toppings);  
  
}
```




```
class Pizza {  
  int slices = 8;  
  List<String> toppings;  
  
  Pizza(this.toppings);  
  
  bake() async {  
    return Future.delayed(const Duration(seconds: 5));  
  }  
}
```



```
class Pizza {  
  int slices = 8;  
  List<String> toppings;  
  
  Pizza(this.toppings);  
  
  bake() async {  
    return Future.delayed(const Duration(seconds: 5));  
  }  
}  
  
main() async {  
  
}
```



```
class Pizza {  
    int slices = 8;  
    List<String> toppings;  
  
    Pizza(this.toppings);  
  
    bake() async {  
        return Future.delayed(const Duration(seconds: 5));  
    }  
}  
  
main() async {  
    var dinner = Pizza(['pineapple']);  
  
}
```



```
class Pizza {  
    int slices = 8;  
    List<String> toppings;  
  
    Pizza(this.toppings);  
  
    bake() async {  
        return Future.delayed(const Duration(seconds: 5));  
    }  
}  
  
main() async {  
    var dinner = Pizza(['pineapple']);  
    await dinner.bake();  
}
```



```
class Pizza {
  int slices = 8;
  List<String> toppings;

  Pizza(this.toppings);

  bake() async {
    return Future.delayed(const Duration(seconds: 5));
  }
}

main() async {
  var dinner = Pizza(['pineapple']);
  await dinner.bake();
  print('yum! ${dinner.toppings.join(', ')}');
}
```

Familiar



Flutter



Everything's a
widgets!



```
class HelloWorld extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Text('Hello World');  
  }  
}
```



```
class HelloWorld extends HTMLElement {  
  connectedCallback() {  
    this.innerHTML = '<p>Hello World</p>';  
  }  
}
```





```
class HelloWorld extends React.Component {  
  render() {  
    return <p>Hello World</p>;  
  }  
}
```



```
<app-header reveals>
  <app-toolbar>
    <div main-title>My app</div>
  </app-toolbar>
</app-header>
<div style="text-align: center;">
  <p>hello world</p>
</div>
```



```
MaterialApp(  
  title: 'Welcome to Flutter',  
  home: Scaffold(  
    appBar: AppBar(  
      title: Text('Welcome to Flutter'),  
    ),  
    body: Center(  
      child: Text('Hello World'),  
    ),  
  ),  
);
```



Gotcha

No templates



```
<div class="entry">  
  <h1>{{title}}</h1>  
  <div class="body">  
    {{body}}  
  </div>  
</div>
```

No CSS



```
<style>
  .padded {
    padding: 8px;
  }
</style>
<div class="padded">...</div>
```




```
Padding(  
  padding: EdgeInsets.all(8.0),  
  child: const Card(child: Text('Hello World!')),  
);
```



```
GestureDetector(  
  onTap: () => print('Yes!'),  
  child: Text('Pick me!'),  
);
```





```
StoreProvider<int>(  
  store: store,  
  child: Text('I am great store'),  
);
```



“Flutter widgets are built using a modern
react-style framework”

StatefulWidget



```
class MyHomePage extends StatefulWidget {  
  _MyHomePageState createState() => _MyHomePageState();  
}
```





```
class _HomePageState extends State<HomePage> {  
  int counter = 0;  
  
  void increment() {  
    setState(() { this.counter++; });  
  }  
  
  Widget build(BuildContext context) {  
    return Button(child: Text('Click me'));  
  }  
}
```



State is long-lived



Widgets get reborn



Fast

Stateful hot reload

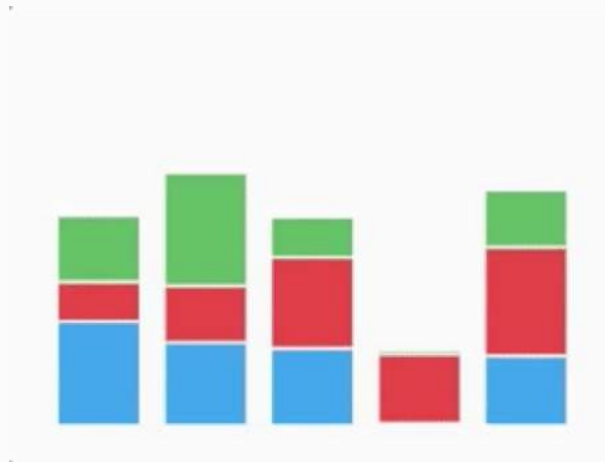
Reactive

Animation





Custom Animation with Tween



Zero to One with Flutter



Testing



Unit

package:test



```
import 'package:test/test.dart';

void main() {
  test('my first unit test', () {
    var answer = 42;
    expect(answer, 42);
  });
}
```



Widget

package:flutter_test



```
void main() {  
  testWidgets('MyWidget has a title and message', (WidgetTester tester) async {  
    await tester.pumpWidget(MyWidget(title: 'T', message: 'M'));  
    final titleFinder = find.text('T');  
    final messageFinder = find.text('M');  
  
    expect(titleFinder, findsOneWidget);  
    expect(messageFinder, findsOneWidget);  
  });  
}
```





Integration

“tested app is launched on a real device
or in an emulator”



```
void main() {  
  group('Counter App', () {  
    final counterTextFinder = find.byValueKey('counter');  
    final buttonFinder = find.byValueKey('increment');  
    FlutterDriver driver;  
  
    setUpAll(() async {  
      driver = await FlutterDriver.connect();  
    });  
  
    test('starts at 0', () async {  
      expect(await driver.getText(counterTextFinder), '0');  
    });  
  
    test('increments the counter', () async {  
      await driver.tap(buttonFinder);  
      expect(await driver.getText(counterTextFinder), '1');  
    });  
  });  
}
```

Test more things



Flutter For Web – Technical Preview

Android + iOS + (Not so good Web) = ❤️

<http://mtechviral.com/myportfolio/>

<http://mtechviral.com/Flutter-UI-Kit/>

<https://flutter.github.io/samples/>

Where to start

[Youtube.com/mtechviral](https://www.youtube.com/mtechviral)

Flutter.dev

flutter.dev/docs/codelabs

[github.com/Solido/
awesome-flutter](https://github.com/Solido/awesome-flutter)

Thank You

Twitter - @imthepk | Github - @iampawan | Youtube - @mtechviral