05-10-2019

Flutter
After Hello World
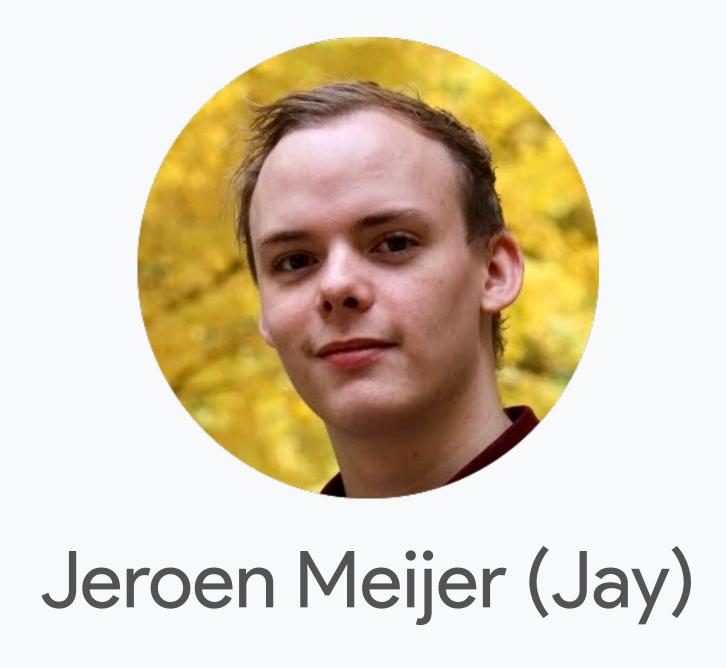
Jeroen Meijer (Jay)

https://jfk.dev

@JFKDEV

/jeroen-meijer

#DevFest

GDG

Who am I

Who am I

▸ Jeroen Meijer (Jay)

▸ Dart & Flutter GDE

▸ Flutter freelancer

▸ Admin and Head of GitHub for the Flutter Community 💙

▸ In love with Flutter for 2 years.

▸ Writer of the VS Code extension Pubspec Assist

Flutter

Flutter

▸ Portable UI toolkit

▸ Open source

▸ Native apps on almost any platform (iOS, Android, Web)

▸ High performance

▸ Uses custom rendering engine and UI elements

▸ Complete control over the screen

IDE

IDE

▸ **Choose your editor** (VS Code, IntelliJ/Android Studio, Other)

IDE

▸ **Choose your editor** (VS Code, IntelliJ/Android Studio, Other)

▸ **Learn your editor** (extensions, breakpoints, hotkeys)

GDG

IDE

▸ **Choose your editor** (VS Code, IntelliJ/Android Studio, Other)

▸ **Learn your editor** (extensions, breakpoints, hotkeys)

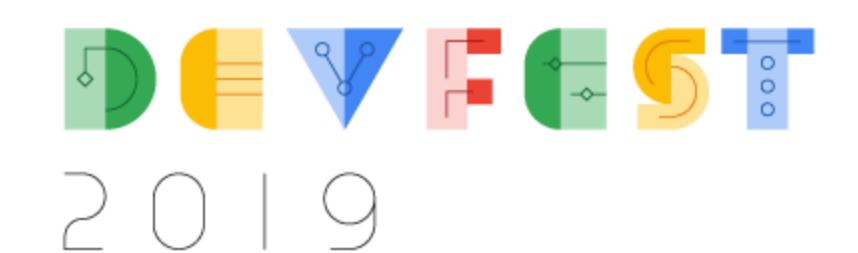▸ **Use linters** (clean code, consistency, readability)

GDG

Tooling

Tooling
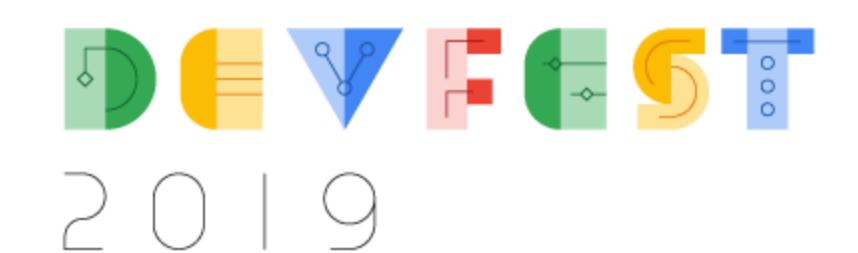
‣ `flutter` `run` `-d` `DEVICE_ID`

GDG

Tooling

▸ `flutter run -d DEVICE_ID`

▸ `flutter packages get | flutter pub get`

Tooling

- `flutter run -d DEVICE_ID`
- `flutter packages get | flutter pub get`
- `flutter devices`

Tooling

- `flutter run -d DEVICE_ID`
- `flutter packages get | flutter pub get`
- `flutter devices`
- `flutter doctor`

Tooling

▸ `flutter run -d DEVICE_ID`

▸ `flutter packages get | flutter pub get`

▸ `flutter devices`

▸ `flutter doctor`

▸ `flutter analyze`

GDG

Tooling

- ▸ flutter `run -d DEVICE_ID`
- ▸ flutter `packages get | flutter pub get`
- ▸ flutter `devices`
- ▸ flutter `doctor`
- ▸ flutter `analyze`
- ▸ flutter `clean`

GDG

Widgets

GDG

Widgets

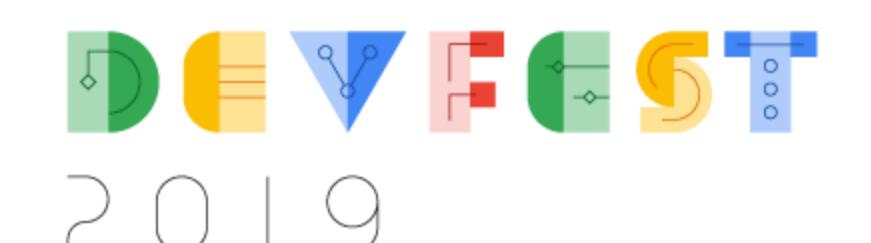▸ Immutable Classes, PODOs (Plain Old Dart Objects)

GDG

Widgets

▸ Immutable Classes, PODOs (Plain Old Dart Objects)

▸ 2 types: `StatelessWidget` and `StatefulWidget`

Widgets

▸ Immutable Classes, PODOs (Plain Old Dart Objects)

▸ 2 types: `StatelessWidget` and `StatefulWidget`

▸ Composed together to create UI

GDG

Widgets

▸ Immutable Classes, PODOs (Plain Old Dart Objects)

▸ 2 types: `StatelessWidget` and `StatefulWidget`

▸ Composed together to create UI

▸ Can do anything other classes can

GDG

Widgets
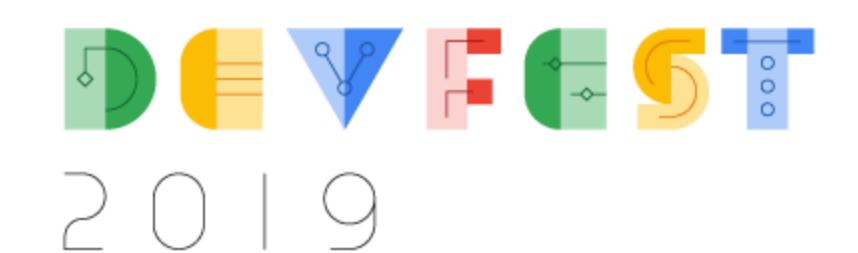
▸ Immutable Classes, PODOs (Plain Old Dart Objects)

▸ 2 types: `StatelessWidget` and `StatefulWidget`

▸ Composed together to create UI

▸ Can do anything other classes can

  ▸ Parameters `Button(title: 'Submit', color: Colors.blue)`

Widgets

▸ Immutable Classes, PODOs (Plain Old Dart Objects)

▸ 2 types: `StatelessWidget` and `StatefulWidget`

▸ Composed together to create UI

▸ Can do anything other classes can

  ▸ Parameters `Button(title: 'Submit', color: Colors.blue)`

  ▸ Functions   `_onTap() { … }`

Widgets

▸ Immutable Classes, PODOs (Plain Old Dart Objects)

▸ 2 types: `StatelessWidget` and `StatefulWidget`

▸ Composed together to create UI

▸ Can do anything other classes can

  ▸ Parameters `Button(title: 'Submit', color: Colors.blue)`

  ▸ Functions `_onTap() { … }`

  ▸ Getters `get _displayedNames ⇒ names.take(3);`

GDG

Widgets

Widgets

▸ Built-in widget packages are material, cupertino and widgets

GDG

Widgets

▶ Built-in widget packages are material, cupertino and widgets

▶ "There's a widget for that!"

GDG

Widgets

▶ Built-in widget packages are material, cupertino and widgets

▶ "There's a widget for that!"

▶ Almost all UI is composable from existing widgets

Widgets

▸ Built-in widget packages are material, cupertino and widgets

▸ "There's a widget for that!"

▸ Almost all UI is composable from existing widgets

▸ If something's missing, look for well-maintained packages
   (Badly maintained packages can cause trouble.)

GDG

Widgets

▸ Built-in widget packages are material, cupertino and widgets

▸ "There's a widget for that!"

▸ Almost all UI is composable from existing widgets

▸ If something's missing, look for well-maintained packages
  (Badly maintained packages can cause trouble.)

▸ Multiple types of widgets (visual, layout, functional)

GDG

State Management

State Management

Provider   Get It   Redux

MobX   Bloc   ChangeNotifier

State Management

Provider   Get It   Redux

MobX   Bloc   ChangeNotifier

▸ Which one is the best?

GDG

State Management

Provider   Get It   Redux

MobX   Bloc   ChangeNotifier

▸ Which one is the best?

▸ None.

State Management

Provider   Get It   Redux

MobX   Bloc   ChangeNotifier

▸ Which one is the best?

▸ None.

▸ How to choose the best one for your project:

GDG

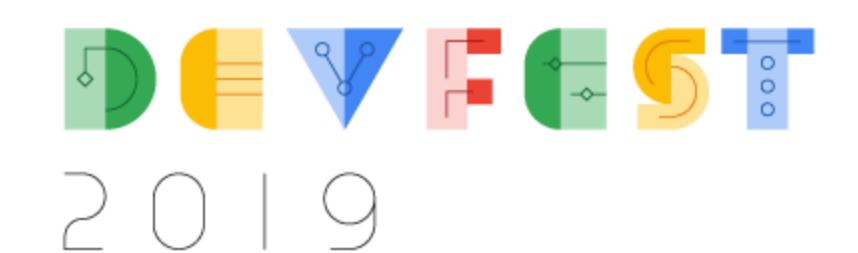# State Management

Provider   Get It   Redux

MobX   Bloc   ChangeNotifier

▸ Which one is the best?

▸ None.

▸ How to choose the best one for your project:

  ▸ Does my app need to be online all the time?

GDG

State Management

Provider   Get It   Redux

MobX   Bloc   ChangeNotifier

▸ Which one is the best?

▸ None.

▸ How to choose the best one for your project:

  ▸ Does my app need to be online all the time?

  ▸ What state management pattern does my team know?

GDG

State Management

Provider   Get It   Redux

MobX   Bloc   ChangeNotifier

▸ Which one is the best?

▸ None.

▸ How to choose the best one for your project:

  ▸ Does my app need to be online all the time?

  ▸ What state management pattern does my team know?

  ▸ How complex is my app and will it grow in the future?

DEVFEST
2019

GDG

Backend

Backend

▸ Dart supports many protocols, such as `http` and `grpc.`

GDG

Backend

▸ Dart supports many protocols, such as `http` and `grpc`.

▸ Two `http` clients

Backend

▸ Dart supports many protocols, such as `http` and `grpc`.

▸ Two `http` clients

    ▸ `package:http` for simple APIs (GET, POST, DELETE, etc.)

GDG

Backend

▸ Dart supports many protocols, such as `http` and `grpc`.

▸ Two `http` clients

   ▸ `package:http`   for simple APIs (GET, POST, DELETE, etc.)

   ▸ `HttpClient`   for fine-grain control (streams, cookies, etc.)

Backend

▸ Dart supports many protocols, such as `http` and `grpc`.

▸ Two `http` clients

   ▸ `package:http`   for simple APIs (GET, POST, DELETE, etc.)

   ▸ `HttpClient`   for fine-grain control (streams, cookies, etc.)

▸ Use existing packages (for example, `firebase`)

Backend

▸ Dart supports many protocols, such as `http` and `grpc`.

▸ Two `http` clients

   ▸ `package:http`   for simple APIs (GET, POST, DELETE, etc.)

   ▸ `HttpClient`   for fine-grain control (streams, cookies, etc.)

▸ Use existing packages (for example, `firebase`)

▸ Use whatever pattern works for you

GDG

Backend

▸ Dart supports many protocols, such as `http` and `grpc`.

▸ Two `http` clients

   ▸ `package:http`   for simple APIs (GET, POST, DELETE, etc.)

   ▸ `HttpClient`   for fine-grain control (streams, cookies, etc.)

▸ Use existing packages (for example, `firebase`)

▸ Use whatever pattern works for you

▸ Recommended: abstract API class + repository pattern

DEVFEST
2019

GDG

Packages

Packages

▸ Extend your app's functionality by importing existing code

GDG

Packages

▶ Extend your app's functionality by importing existing code

▶ Can be found on pub.dev

Packages

▸ Extend your app's functionality by importing existing code

▸ Can be found on pub.dev

▸ Are imported using the pubspec.yaml file
```
dependencies:
    camera: ^0.5.4+3
```

Packages

▸ Extend your app's functionality by importing existing code

▸ Can be found on pub.dev

▸ Are imported using the pubspec.yaml file
```
dependencies:
    camera: ^0.5.4+3
```

▸ Tip: Use Pubspec Assist to easily add and update dependencies.

Plugins

Plugins

▸ Most packages are plugins

GDG

Plugins

▸ Most packages are plugins

▸ Allows two-way communication between Dart and the native platform code

Plugins

▸ Most packages are plugins

▸ Allows two-way communication between Dart and the native platform code
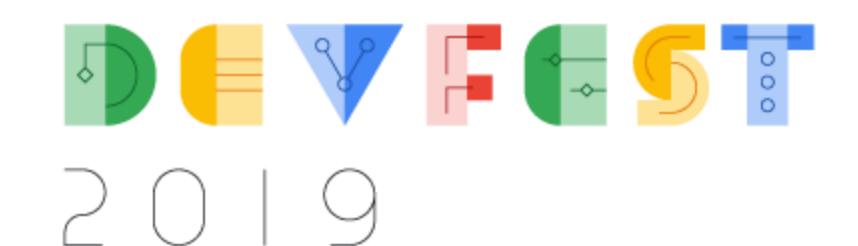
▸ Call native APIs from Flutter

Plugins

▶ Most packages are plugins

▶ Allows two-way communication between Dart and the native platform code

▶ Call native APIs from Flutter

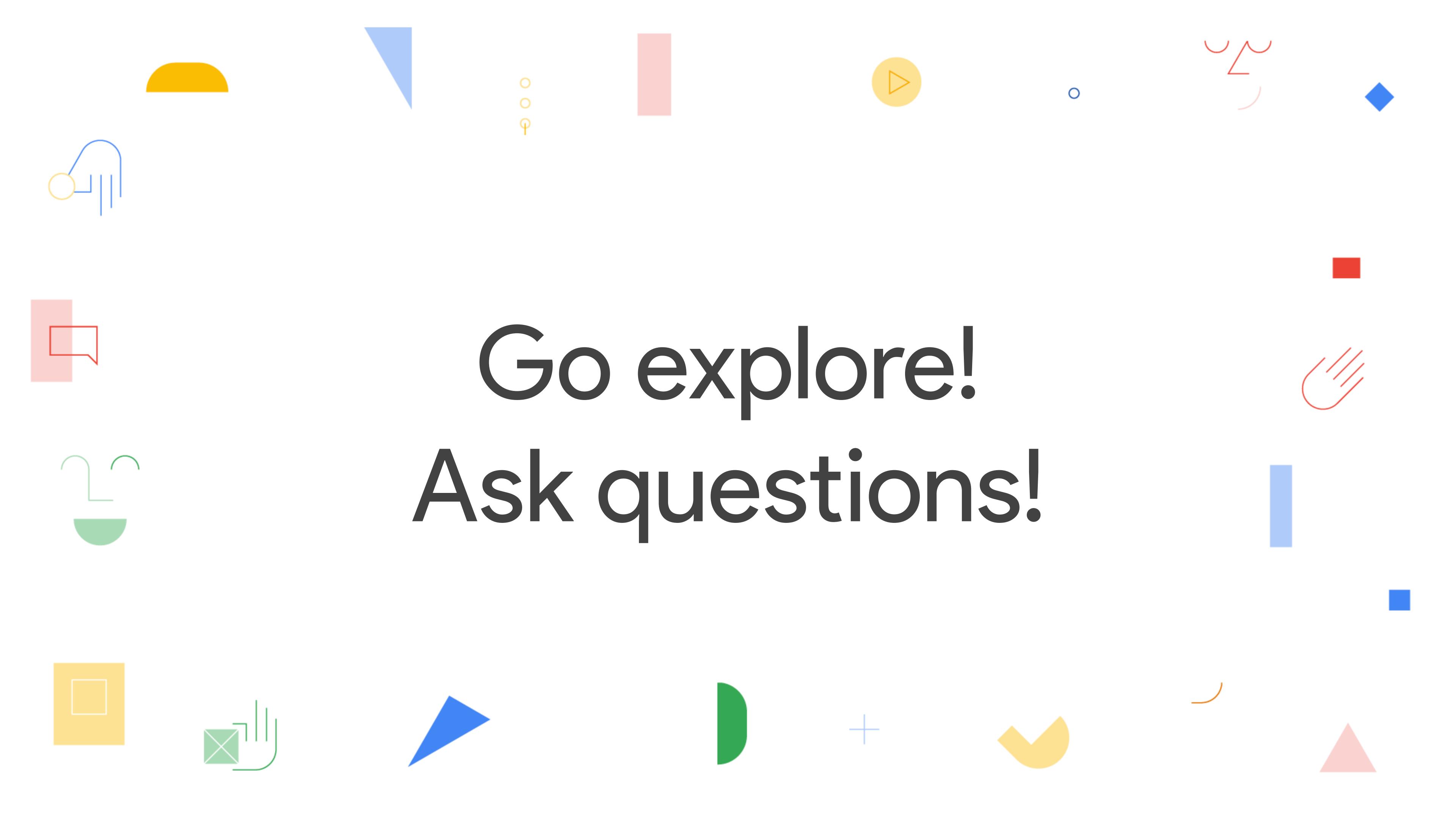▶ Wrap your existing Java/Kotlin and Objective C/Swift code.

GDG

Plugins

▸ Most packages are plugins

▸ Allows two-way communication between Dart and the native platform code

▸ Call native APIs from Flutter

▸ Wrap your existing Java/Kotlin and Objective C/Swift code.

▸ `camera`, `bluetooth`, `connectivity`, `firebase`

Go explore!
Ask questions!

DEVFEST 2019

Flutter
After Hello World

Thank you for your attention.

Jeroen Meijer (Jay)

https://jfk.dev 🌐

@JFKDEV 🐦

/jeroen-meijer

Slides and code available at: https://jfk.dev/talks

GDG