

互联网应用开发技术

Web Application Development

第17课

WEB移动端-FLUTTER & DART

Episode Seventeen
Flutter & Dart

陈昊鹏
chen-hp@sjtu.edu.cn



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



- Dart
 - is a client-optimized language for fast apps on any platform



- Get the Flutter SDK
 - Download and unzip the SDK package

```
flutter_macos_v1.12.13+hotfix.9-stable.zip
```

- Or check it from the Flutter repo on GitHub

```
git clone https://github.com/flutter/flutter.git -b stable
```

- Add the flutter tool to your path

```
export PATH="$PATH:[PATH_TO_FLUTTER_GIT_DIRECTORY]/flutter/bin"
```

- Run flutter doctor
 - flutter doctor

```
[HAOPENGdeiMac:bin haopengchen$ flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, v1.12.13+hotfix.9, on Mac OS X 10.15.4 19E266,
    locale zh-Hans-CN)

[✓] Android toolchain - develop for Android devices (Android SDK version 29.0.3)
[✓] Xcode - develop for iOS and macOS (Xcode 11.4)
[✓] Android Studio (version 3.6)
[✓] IntelliJ IDEA Ultimate Edition (version 2019.3.3)
[✓] VS Code (version 1.43.2)
[✓] Connected device (2 available)

• No issues found!
```

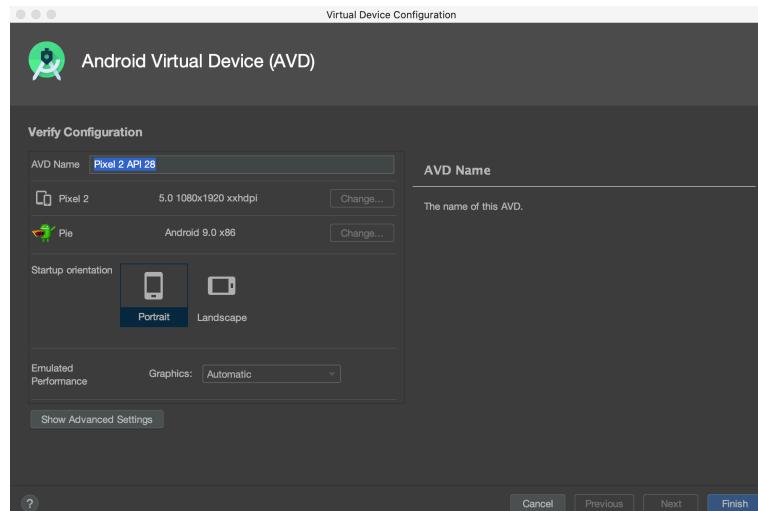
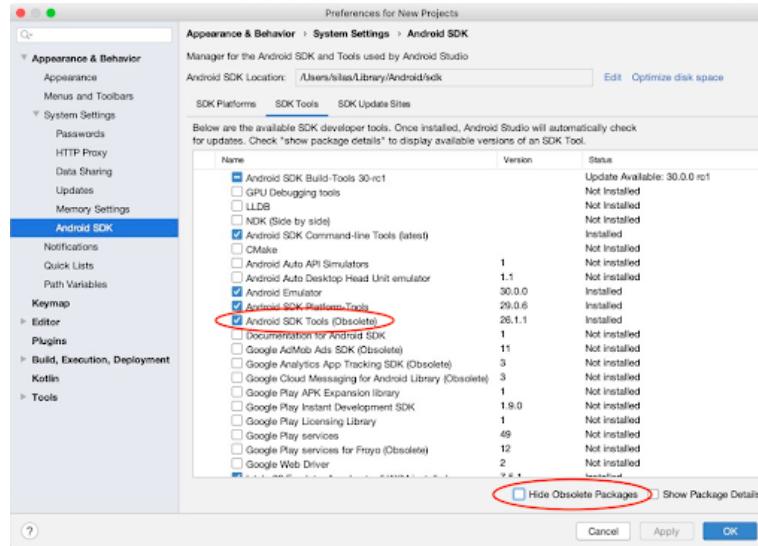
- iOS setup
 - Install Xcode
 - Configure the Xcode command-line tools

```
sudo xcode-select --switch /Applications/Xcode.app/Contents/Developer  
sudo xcodebuild -runFirstLaunch
```
 - Make sure the Xcode license agreement is signed by either opening Xcode once and confirming or running `sudo xcodebuild -license` from the command line.
- Set up the iOS simulator

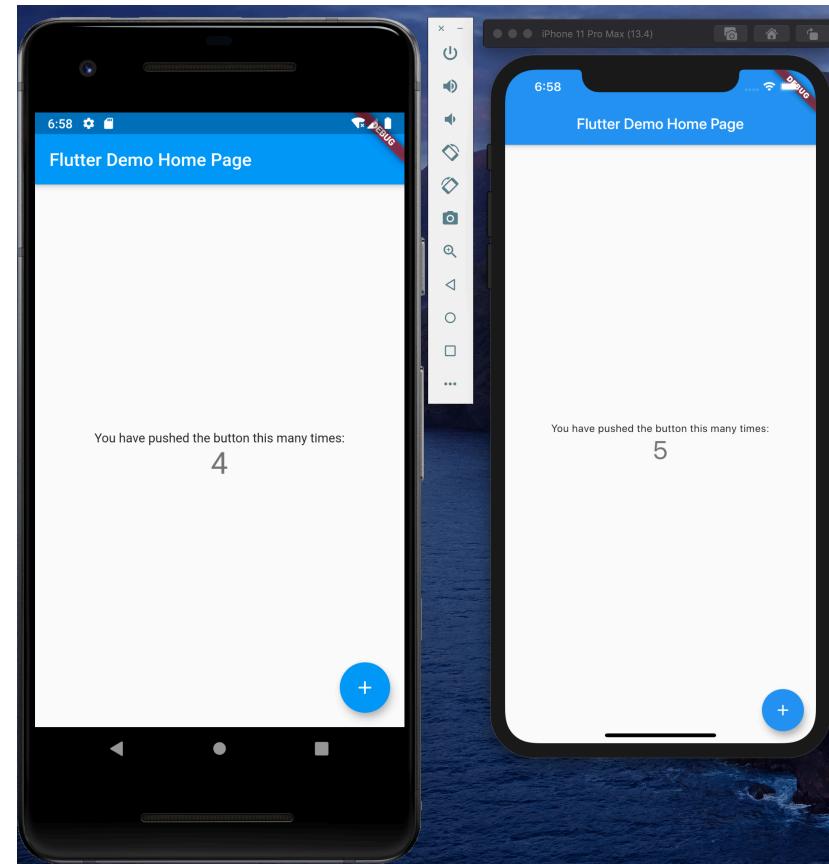
```
open -a Simulator
```

Install

- Android setup
 - Install Android Studio
 - Set up your Android device
 - Set up the Android emulator
 - Accept licenses
- \$ cd [sdk path]/tools/bin
\$./sdkmanager --licenses



- Create and run a simple Flutter app
 - Create a new Flutter app by running the following from the command line:
 - `$ flutter create my_app`
 - A `my_app` directory is created, containing Flutter's starter app. Enter this directory:
 - `$ cd my_app`
 - To launch the app in the Simulator, ensure that the Simulator is running and enter:
 - `$ flutter run -d all`



Install



REliable, INtelligent & Scalable Systems

Observatory > vm@ws://127.0.0.1:53312/C6gVUUMkyq0=/ws Refresh

VM

name	vm@ws://127.0.0.1:53312/C6gVUUMkyq0=/ws
version	2.7.2 (Mon Mar 23 22:11:27 2020 +0100) on "android_ia32"
embedder	Flutter
started at	2020-04-07 18:19:14.206
uptime	0:22:26.710000
refreshed at	2020-04-07 18:41:40.916
pid	7810
peak memory	220.6MB
current memory	214.4MB
native zone memory	0B
native heap memory	unavailable
native heap allocation count	unavailable

[see flags](#) [view timeline](#) [view native memory profile](#)

Isolates (1)

Isolate 2613210942277799 (main)

Activity	Percentage
VM	25.39%
DartCompile	40.68%
LoadBytecode	0.00%
LoadWait	0.00%
CompileOptimized	0.00%
ClassLoading	0.00%
CompileParseR...	0.00%
Idle	0.00%

idle [debug]

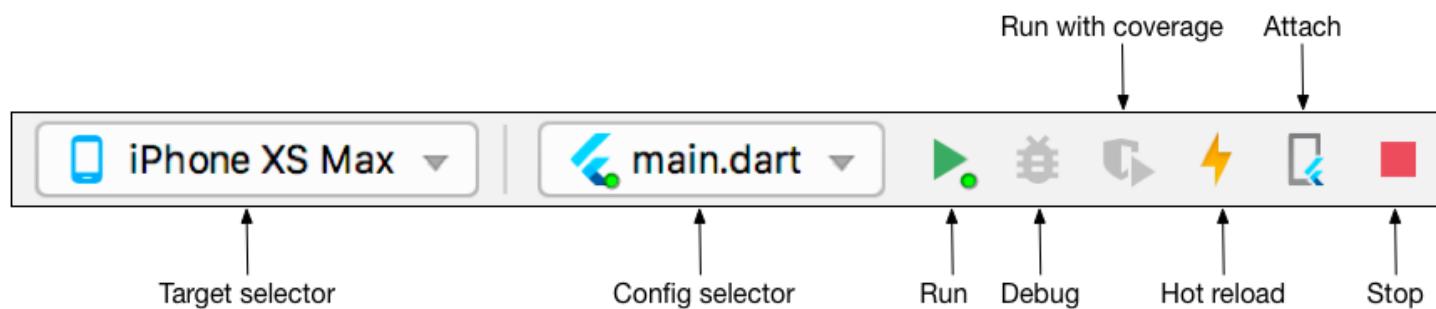
Activity	Percentage	Notes
new heap	1.0MB of 4.0MB	
old heap	33.6MB of 35.4MB	

[see debug](#) [see class hierarchy](#) [see cpu profile](#) [see cpu profile \(table\)](#) [see allocation profile](#) [see heap snapshot](#) [see heap map](#) [see metrics](#) [see persistent handles](#) [see ports](#) [see logging](#)

[View documentation](#) [File a bug report](#)

Create the app with IDEA

- Create the app
 - Open the IDE and select **Start a new Flutter project**.
 - Select **Flutter Application** as the project type. Then click **Next**.
 - Verify the Flutter SDK path specifies the SDK's location (select **Install SDK...** if the text field is blank).
 - Enter a project name (for example, myapp). Then click **Next**.
 - Click **Finish**.
 - Wait for Android Studio to install the SDK and create the project.
- Run the app



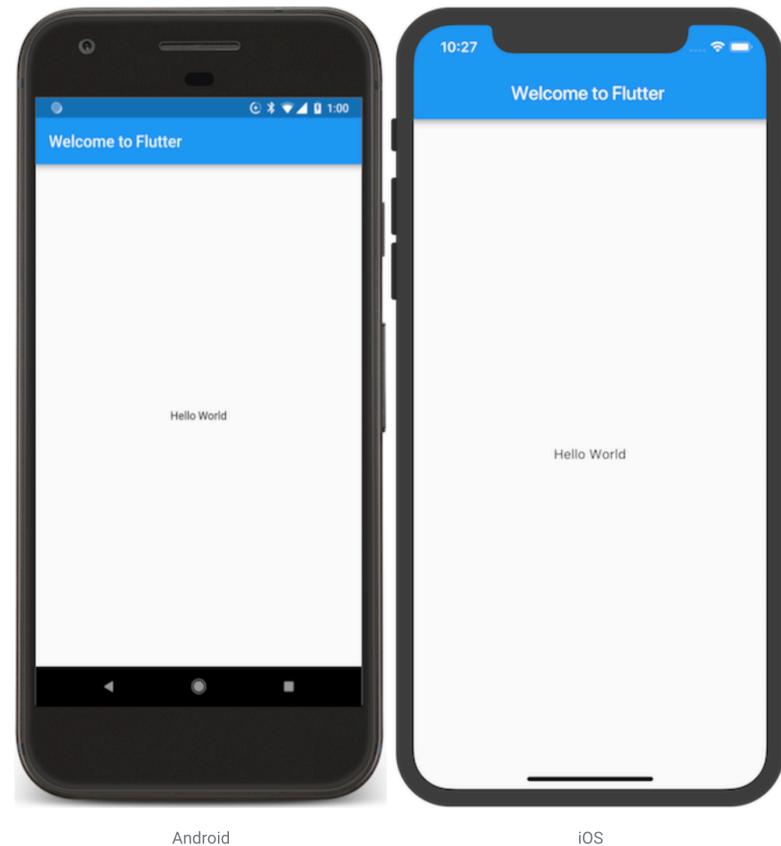
Write your first Flutter app

- Create the starter Flutter app

- lib/main.dart

```
// Copyright 2018 The Flutter team. All rights reserved.  
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.
```

```
import 'package:flutter/material.dart';  
  
void main() => runApp(MyApp());  
  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Welcome to Flutter',  
      home: Scaffold(  
        appBar: AppBar(  
          title: Text('Welcome to Flutter'),  
        ),  
        body: Center(  
          child: Text('Hello World'),  
        ),  
      ),  
    );  
  }  
}
```



Write your first Flutter app



REliable, INtelligent & Scalable Systems

- Use an external package

- lib/main.dart

```
// Copyright 2018 The Flutter team. All rights reserved.  
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.  
import 'package:flutter/material.dart';  
import 'package:english_words/english_words.dart';  
void main() => runApp(MyApp());  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    final wordPair = WordPair.random();  
    return MaterialApp(  
      title: 'Welcome to Flutter',  
      home: Scaffold(  
        appBar: AppBar(  
          title: Text('Welcome to Flutter'),  
        ),  
        body: Center(  
          child: Text(wordPair.asPascalCase),  
        ),  
      ),  
    );  
  }  
}
```

- pubspec.yaml

```
name: startup_namer  
description: A startup-namer app.  
version: 1.0.0+1
```

dependencies

```
dependencies:  
  flutter:  
    sdk: flutter  
  cupertino_icons: ^0.1.2  
  english_words: ^3.1.0
```

end dependencies

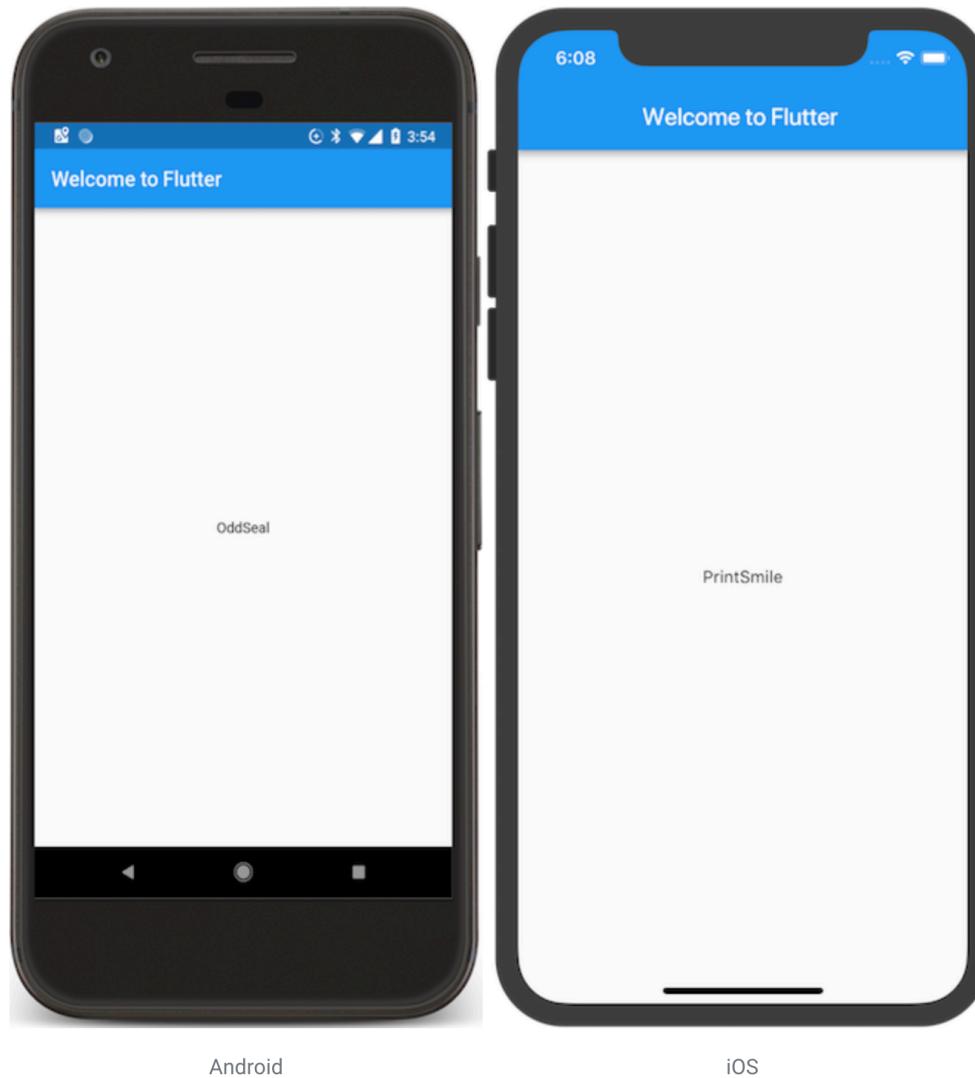
```
dev_dependencies:  
  flutter_test:  
    sdk: flutter  
  pedantic: ^1.4.0
```

```
flutter:  
  uses-material-design: true
```

Write your first Flutter app



REliable, INtelligent & Scalable Systems



Android

iOS

Write your first Flutter app



REliable, INtelligent & Scalable Systems

- Add a Stateful widget

```
// Copyright 2018 The Flutter team. All rights reserved.  
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.
```

```
import 'package:flutter/material.dart';  
import 'package:english_words/english_words.dart';  
  
void main() => runApp(MyApp());  
  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Welcome to Flutter',  
      home: Scaffold(  
        appBar: AppBar(  
          title: Text('Welcome to Flutter'),  
        ),  
        body: Center(  
          child: RandomWords(),  
        ),  
      ),  
    );  
  }  
}  
  
// #docregion RandomWordsState, RWS-class-only  
class RandomWordsState extends State<RandomWords> {  
  // #enddocregion RWS-class-only  
  @override  
  Widget build(BuildContext context) {  
    final wordPair = WordPair.random();  
    return Text(wordPair.asPascalCase);  
  }  
  // #docregion RWS-class-only  
}  
  // #enddocregion RandomWordsState, RWS-class-only  
  
// #docregion RandomWords  
class RandomWords extends StatefulWidget {  
  @override  
  RandomWordsState createState() => RandomWordsState();  
}
```

Write your first Flutter app



REliable, INtelligent & Scalable Systems

- Create an infinite scrolling ListView

```
// Copyright 2018 The Flutter team. All rights reserved.  
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.  
  
import 'package:flutter/material.dart';  
import 'package:english_words/english_words.dart';  
  
void main() => runApp(MyApp());  
  
// #docregion MyApp  
class MyApp extends StatelessWidget {  
  // #docregion build  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Startup Name Generator',  
      home: RandomWords(),  
    );  
  }  
  // #enddocregion build  
}  
// #enddocregion MyApp  
  
// #docregion RWS-var  
class RandomWordsState extends State<RandomWords> {  
  final _suggestions = <WordPair>[];  
  final _biggerFont = const TextStyle(fontSize: 18.0);  
  // #enddocregion RWS-var  
  
  // #docregion _buildSuggestions  
  Widget _buildSuggestions() {  
    return ListView.builder(  
      padding: const EdgeInsets.all(16.0),  
      itemBuilder: /*1*/ (context, i) {  
        if (i.isOdd) return Divider(); /*2*/  
  
        final index = i ~/ 2; /*3*/  
        if (index >= _suggestions.length) {  
          _suggestions.addAll(generateWordPairs().take(10)); /*4*/  
        }  
        return _buildRow(_suggestions[index]);  
      },  
    );  
  }  
  // #enddocregion _buildSuggestions
```

Write your first Flutter app

- Create an infinite scrolling ListView

```
// #docregion _buildRow
Widget _buildRow(WordPair pair) {
  return ListTile(
    title: Text(
      pair.asPascalCase,
      style: _biggerFont,
    ),
  );
}
```

```
// #enddocregion _buildRow
```

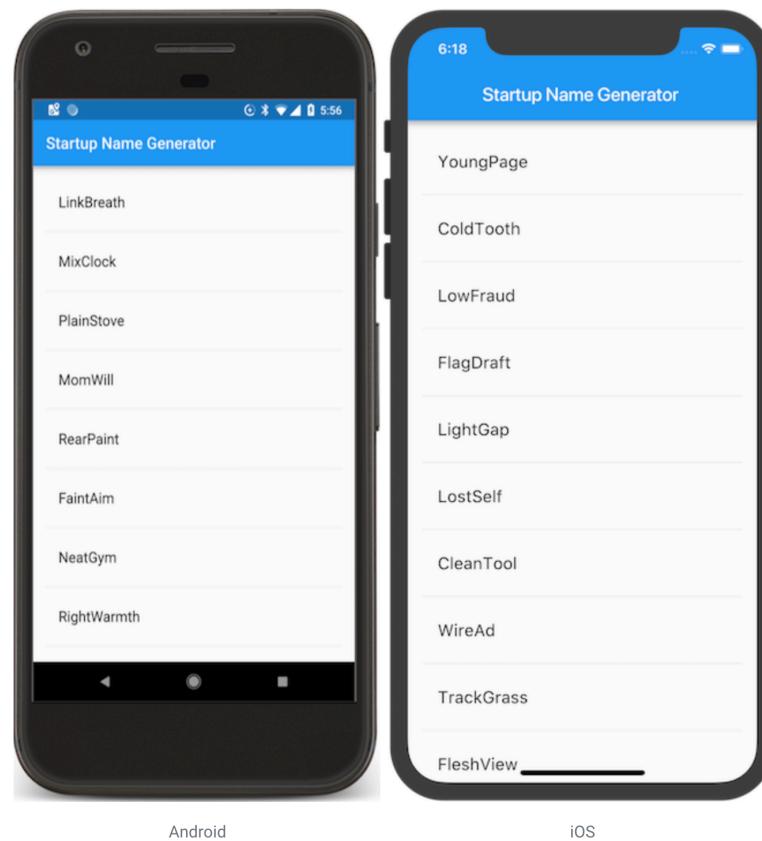
```
// #docregion RWS-build
@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Startup Name Generator'),
    ),
    body: _buildSuggestions(),
  );
}
```

```
// #enddocregion RWS-build
```

```
// #docregion RWS-var
}
```

```
// #enddocregion RWS-var
```

```
class RandomWords extends StatefulWidget {
  @override
  RandomWordsState createState() => RandomWordsState();
}
```



Write your first Flutter app



REliable, INtelligent & Scalable Systems

- Add icons and interactivity to the list

```
// Copyright 2018 The Flutter team. All rights reserved.  
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.  
  
import 'package:flutter/material.dart';  
import 'package:english_words/english_words.dart';  
  
void main() => runApp(MyApp());  
  
// #docregion MyApp  
class MyApp extends StatelessWidget {  
  // #docregion build  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Startup Name Generator',  
      home: RandomWords(),  
    );  
  }  
  // #enddocregion build  
}  
// #enddocregion MyApp  
  
// #docregion RWS-var  
class RandomWordsState extends State<RandomWords> {  
  final _suggestions = <WordPair>[];  
  final Set<WordPair> _saved = <WordPair>{};  
  final _biggerFont = const TextStyle(fontSize: 18.0);  
  // #enddocregion RWS-var  
  
  // #docregion _buildSuggestions  
  Widget _buildSuggestions() {  
    return ListView.builder(  
      padding: const EdgeInsets.all(16.0),  
      itemBuilder: /*1*/ (context, i) {  
        if (i.isOdd) return Divider(); /*2*/  
  
        final index = i ~/ 2; /*3*/  
        if (index >= _suggestions.length) {  
          _suggestions.addAll(generateWordPairs().take(10)); /*4*/  
        }  
        return _buildRow(_suggestions[index]);  
      },  
    );  
  }  
  // #enddocregion _buildSuggestions
```

Write your first Flutter app



REliable, INtelligent & Scalable Systems

- Add icons and interactivity to the list

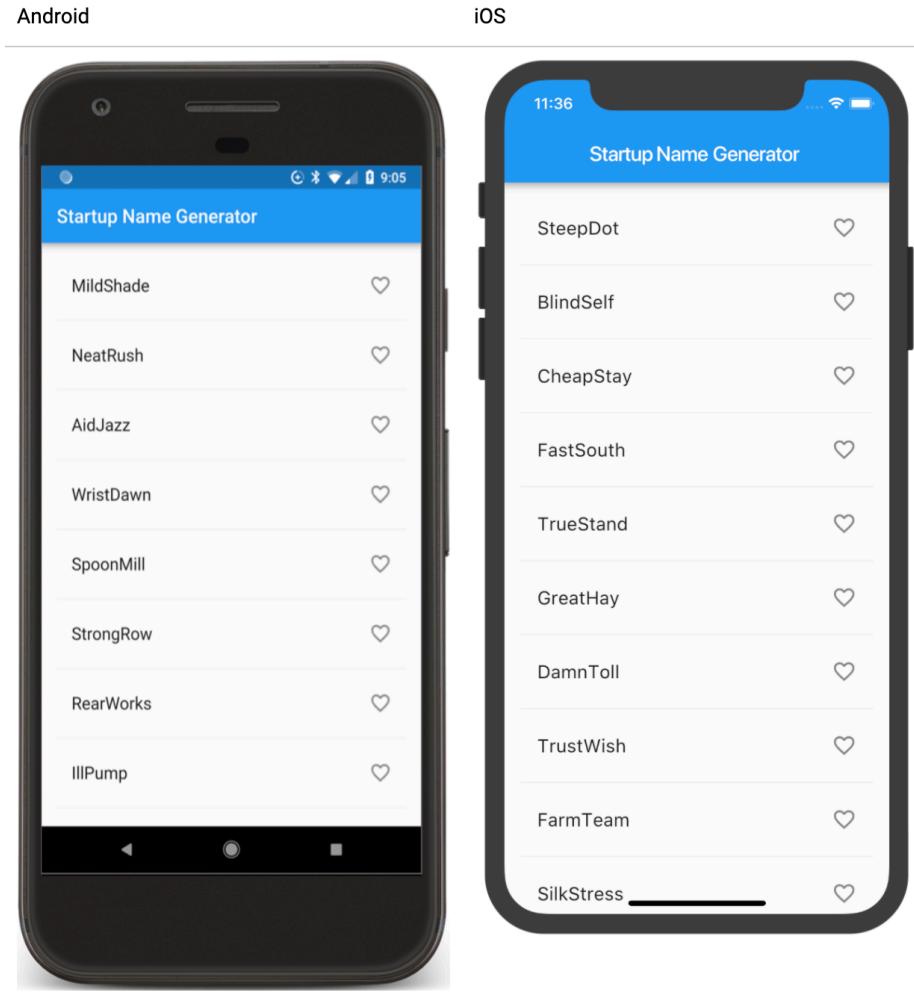
```
// #docregion _buildRow
Widget _buildRow(WordPair pair) {
    final alreadySaved = _saved.contains(pair);
    return ListTile(
        title: Text(
            pair.asPascalCase,
            style: _biggerFont,
        ),
        trailing: Icon(
            alreadySaved ? Icons.favorite : Icons.favorite_border,
            color: alreadySaved ? Colors.red : null,
        ),
        onTap: () {
            setState(() {
                if (alreadySaved) {
                    _saved.remove(pair);
                } else {
                    _saved.add(pair);
                }
            });
        },
    );
}
// #enddocregion _buildRow

// #docregion RWS-build
@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text('Startup Name Generator'),
        ),
        body: _buildSuggestions(),
    );
}
// #enddocregion RWS-build
// #docregion RWS-var
}
// #enddocregion RWS-var

class RandomWords extends StatefulWidget {
    @override
    RandomWordsState createState() => RandomWordsState();
}
```

Write your first Flutter app

- Add icons and interactivity to the list



Write your first Flutter app

- Navigate to a new screen

```
// Copyright 2018 The Flutter team. All rights reserved.
```

```
// Use of this source code is governed by a BSD-style license that can be  
// found in the LICENSE file.
```

```
import 'package:flutter/material.dart';  
import 'package:english_words/english_words.dart';  
  
void main() => runApp(MyApp());
```

```
// #docregion MyApp  
class MyApp extends StatelessWidget {  
  // #docregion build  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Startup Name Generator',  
      home: RandomWords(),  
    );  
  }  
  // #enddocregion build  
}  
// #enddocregion MyApp
```

```
// #docregion RWS-var  
class RandomWordsState extends State<RandomWords> {  
  final _suggestions = <WordPair>[];  
  final Set<WordPair> _saved = <WordPair>{};  
  final _biggerFont = const TextStyle(fontSize: 18.0);  
  // #enddocregion RWS-var
```

```
// #docregion _buildSuggestions  
Widget _buildSuggestions() {  
  return ListView.builder(  
    padding: const EdgeInsets.all(16.0),  
    itemBuilder: /*1*/ (context, i) {  
      if (i.isOdd) return Divider(); /*2*/  
  
      final index = i ~/ 2; /*3*/  
      if (index >= _suggestions.length) {  
        _suggestions.addAll(generateWordPairs().take(10)); /*4*/  
      }  
      return _buildRow(_suggestions[index]);  
    },  
  );  
  // #enddocregion _buildSuggestions
```

Write your first Flutter app



REliable, INtelligent & Scalable Systems

- Navigate to a new screen

```
// #docregion _buildRow
Widget _buildRow(WordPair pair) {
    final alreadySaved = _saved.contains(pair);
    return ListTile(
        title: Text(
            pair.asPascalCase,
            style: _biggerFont,
        ),
        trailing: Icon(
            alreadySaved ? Icons.favorite : Icons.favorite_border,
            color: alreadySaved ? Colors.red : null,
        ),
        onTap: () {
            setState(() {
                if (alreadySaved) {
                    _saved.remove(pair);
                } else {
                    _saved.add(pair);
                }
            });
        },
    );
}
// #enddocregion _buildRow
```

```
// #docregion RWS-build
@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text('Startup Name Generator'),
            actions: <Widget>[
                IconButton(icon: Icon(Icons.list), onPressed: _pushSaved),
            ],
        ),
        body: _buildSuggestions(),
    );
}
// #enddocregion RWS-build
```

Write your first Flutter app



REliable, INtelligent & Scalable Systems

- Navigate to a new screen

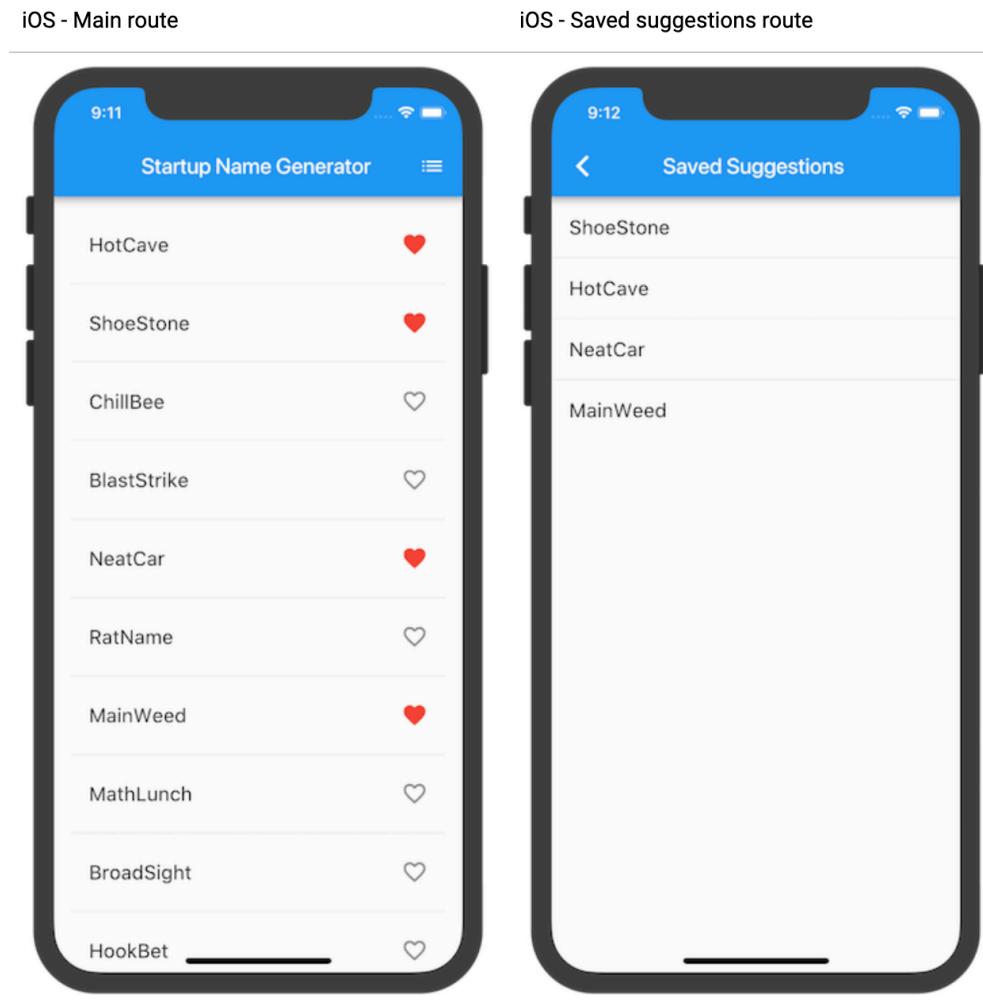
```
void _pushSaved() {
  Navigator.of(context).push(
    MaterialPageRoute<void>(
      // Add 20 lines from here...
      builder: (BuildContext context) {
        final tiles = _saved.map(
          (WordPair pair) {
            return ListTile(
              title: Text(
                pair.asPascalCase,
                style: _biggerFont,
              ),
            );
          },
        );
        final divided = ListTile.divideTiles(
          context: context,
          tiles: tiles,
        ).toList();
      }
    )
  );
}
```

```
return Scaffold(
  appBar: AppBar(
    title: Text('Saved Suggestions'),
  ),
  body: ListView(children: divided),
);
},
);
}
}
// #docregion RWS-var
}
// #enddocregion RWS-var

class RandomWords extends StatefulWidget {
  @override
  RandomWordsState createState() => RandomWordsState();
}
```

Write your first Flutter app

- Navigate to a new screen



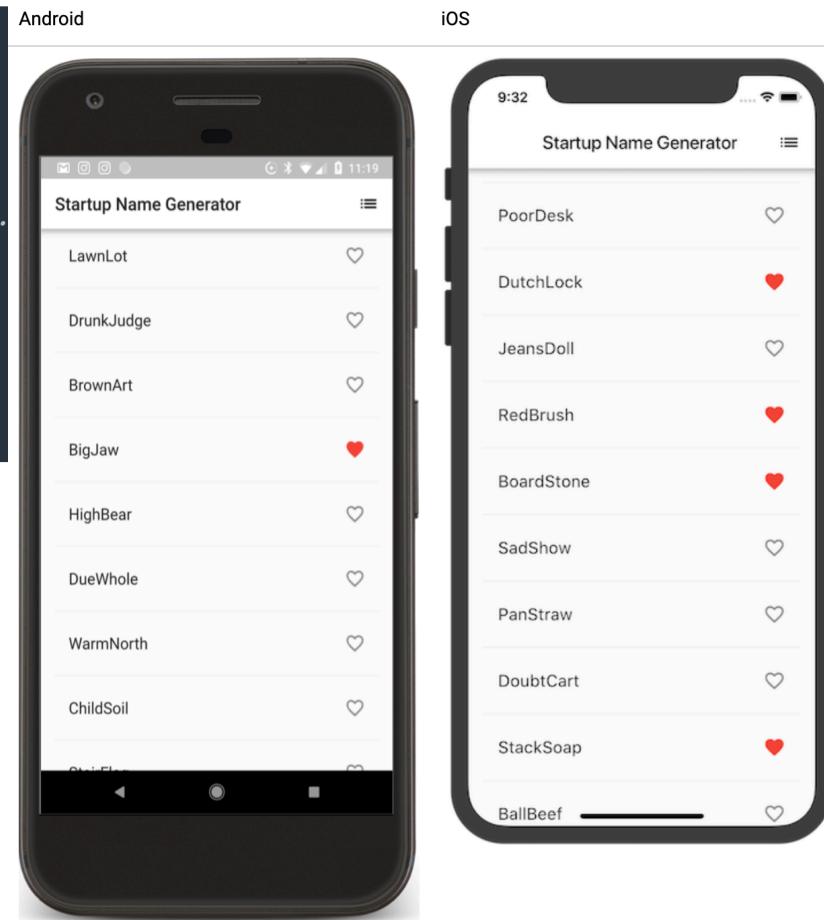
Write your first Flutter app



REliable, INtelligent & Scalable Systems

- Change the UI using Themes

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Startup Name Generator',  
      theme: ThemeData(          // Add the 3 lines from here...  
        primaryColor: Colors.white,  
      ),                      // ... to here.  
      home: RandomWords(),  
    );  
  }  
}
```



A tour of the Dart language



REliable, INtelligent & Scalable Systems

- Dart is a class-based, single-inheritance, pure object-oriented programming language.
- A basic Dart program
 - The following code uses many of Dart's most basic features:

```
// Define a function.  
printInteger(int aNumber) {  
  print('The number is $aNumber.');// Print to console.  
}
```

```
// This is where the app starts executing.  
main() {  
  var number = 42; // Declare and initialize a variable.  
  printInteger(number); // Call a function.  
}
```

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- Hello World
 - Every app has a `main()` function. To display text on the console, you can use the top-level `print()` function:

```
void main() {  
  print('Hello, World!');  
}
```

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- Variables
 - Even in type-safe Dart code, most variables don't need explicit types, thanks to type inference:

```
var name = 'Voyager I';
var year = 1977;
var antennaDiameter = 3.7;
var flybyObjects = ['Jupiter', 'Saturn', 'Uranus',
'Neptune'];
var image = {
  'tags': ['saturn'],
  'url': '//path/to/saturn.jpg'
};
```

A tour of the Dart language

- Control flow statements

- Dart supports the usual control flow statements:

```
if (year >= 2001) {  
    print('21st century');  
} else if (year >= 1901) {  
    print('20th century');  
}
```

```
for (var object in flybyObjects) {  
    print(object);  
}
```

```
for (int month = 1; month <= 12; month++) {  
    print(month);  
}
```

```
while (year < 2016) {  
    year += 1;  
}
```

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- Functions
 - We recommend specifying the types of each function's arguments and return value: :

```
int fibonacci(int n) {  
  if (n == 0 || n == 1) return n;  
  return fibonacci(n - 1) + fibonacci(n - 2);  
}  
  
var result = fibonacci(20);
```

- A shorthand => (arrow) syntax is handy for functions that contain a single statement

```
flybyObjects.where((name) => name.contains('turn')).forEach(print);
```

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- Comments
 - Dart comments usually start with //

// This is a normal, one-line comment.

*/// This is a documentation comment, used to document libraries,
/// classes, and their members. Tools like IDEs and dartdoc treat
/// doc comments specially.*

/* Comments like these are also supported. */

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- Imports
 - To access APIs defined in other libraries, use `import`

```
// Importing core libraries
import 'dart:math';
```

```
// Importing libraries from external packages
import 'package:test/test.dart';
```

```
// Importing files
import 'path/to/my_other_file.dart';
```

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- ## Classes

- Here's an example of a class with three properties, two constructors, and a method. One of the properties can't be set directly, so it's defined using a getter method (instead of a variable).

```
class Spacecraft {  
  String name;  
  DateTime launchDate;  
  
  // Constructor, with syntactic sugar for assignment to members.  
  Spacecraft(this.name, this.launchDate) {  
    // Initialization code goes here.  
  }  
  
  // Named constructor that forwards to the default one.  
  Spacecraft.unlaunched(String name) : this(name, null);  
  
  int get launchYear =>  
    launchDate?.year; // read-only non-final property  
  
  // Method.  
  void describe() {  
    print('Spacecraft: $name');  
    if (launchDate != null) {  
      int years =  
        DateTime.now().difference(launchDate).inDays ~/  
          365;  
      print('Launched: $launchYear ($years years ago)');  
    } else {  
      print('Unlaunched');  
    }  
  }  
}
```

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- Classes
 - You might use the Spacecraft class like this:

```
var voyager = Spacecraft('Voyager I', DateTime(1977, 9, 5));  
voyager.describe();
```

```
var voyager3 = Spacecraft.unlaunched('Voyager III');  
voyager3.describe();
```

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- Inheritance
 - Dart has single inheritance

```
class Orbiter extends Spacecraft {  
    num altitude;  
    Orbiter(String name, DateTime launchDate, this.altitude)  
        : super(name, launchDate);  
}
```

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- ## Mixins

- Mixins are a way of reusing code in multiple class hierarchies. The following class can act as a mixin:

```
class Piloted {  
    int astronauts = 1;  
    void describeCrew() {  
        print('Number of astronauts: $astronauts');  
    }  
}
```

- To add a mixin's capabilities to a class, just extend the class with the mixin.

```
class PilotedCraft extends Spacecraft with  
Piloted {  
    // ...  
}
```

- **PilotedCraft** now has the `astronauts` field as well as the `describeCrew()` method.

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- Interfaces and abstract classes
 - Dart has no interface keyword. Instead, all classes implicitly define an interface. Therefore, you can implement any class.

```
class MockSpaceship implements Spacecraft {  
  // ...  
}
```

- You can create an abstract class to be extended (or implemented) by a concrete class. Abstract classes can contain abstract methods (with empty bodies).

```
abstract class Describable {  
  void describe();  
  void describeWithEmphasis() {  
    print('=====');  
    describe();  
    print('=====');  
  }  
}
```

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- **Async**
 - Avoid callback hell and make your code much more readable by using `async` and `await`.

```
const oneSecond = Duration(seconds: 1);
// ...
Future<void> printWithDelay(String message) async {
  await Future.delayed(oneSecond);
  print(message);
}
```

- The method above is equivalent to:

```
Future<void> printWithDelay(String message) {
  return Future.delayed(oneSecond).then((_) {
    print(message);
  });
}
```

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- **Async**

- `async` and `await` help make asynchronous code easy to read

```
Future<void> createDescriptions(Iterable<String> objects) async {
    for (var object in objects) {
        try {
            var file = File('$object.txt');
            if (await file.exists()) {
                var modified = await file.lastModified();
                print(
                    'File for $object already exists. It was modified on $modified.');
                continue;
            }
            await file.create();
            await file.writeAsString('Start describing $object in this file.');
        } on IOException catch (e) {
            print('Cannot create description for $object: $e');
        }
    }
}
```

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- **Async**
 - You can also use `async*`, which gives you a nice, readable way to build streams.

```
Stream<String> report(Spacecraft craft, Iterable<String> objects) async* {  
  for (var object in objects) {  
    await Future.delayed(oneSecond);  
    yield '${craft.name} flies by $object';  
  }  
}
```

A tour of the Dart language



REliable, INtelligent & Scalable Systems

- Exceptions

- To raise an exception, use `throw`:

```
if (astronauts == 0) {  
    throw StateError('No astronauts.');//  
}
```

- To catch an exception, use a `try` statement with `on` or `catch` (or both):

```
try {  
    for (var object in flybyObjects) {  
        var description = await File('$object.txt').readAsString();  
        print(description);  
    }  
} on IOException catch (e) {  
    print('Could not describe object: $e');//  
} finally {  
    flybyObjects.clear();  
}
```

- Google Flutter
 - <http://www.flutter.dev>
- Flutter Gallery
 - <https://flutter.github.io/gallery/#/>
- Flutter Samples
 - <https://flutter.github.io/samples/#>
- Android Studio: License for package Android SDK Build-Tools 28.0.2 not accepted.
 - <https://blog.csdn.net/didah/article/details/87894670>
- A tour of the Dart language
 - <https://dart.dev/guides/language/language-tour#class-variables-and-methods>
- Dart Language Specification v2.2
 - <https://dart.dev/guides/language/specifications/DartLangSpec-v2.2.pdf>
- DartPad
 - <https://dartpad.dev/>
- Language samples
 - <https://dart.dev/samples>



- *Web*开发技术
- *Web Application Development*

Thank You!