



dream
IMPLEMENTATION

<http://dream-implementation.com>

info@dream-implementation.com

What's new in Flutter v1.12

- Dart 2.7
- iOS 13 dark mode completed
- macOS desktop development support (alpha)
- Web development support (beta)
- DartPad online tool with Flutter support



What's new in Flutter v1.12

- Layout Explorer in Dart Dev Tools
- HotUI in Android Studio
- Multi device simultaneous debugging in VS Code
- Android build improvements



What's new in Dart 2.7

Announcing



Dart 2.7

Extension methods

```
extension Numbers on String {  
  int parseInt() =>  
    return int.parse(this);  
}
```

Preview

characters package

```
print('I♥Dart'.  
  characters.take(2));  
// prints I♥
```

Preview

Null safety

```
String firstName;  
DateTime? birthday;  
int? year = birthday?.year;
```

Extension Methods

- add new functionality to any existing class
- auto-complete experience of regular method calls
- ParseNumbers extends on String class with `parseInt()` and `parseDouble()` methods

```
extension ParseNumbers on String {  
    int parseInt() {  
        return int.parse(this);  
    }  
}
```

```
    double parseDouble() {  
        return double.parse(this);  
    }  
}
```

```
main() {  
    int i = '42'.parseInt();  
    print(i);  
}
```

Null Safety (Preview)

- static code analysis to prevent null pointer exceptions
- declare nullable variables with ? after variable type
- available at:

<https://nullsafety.dartpad.dev/>

```
class Person {  
    String firstName;  
    DateTime? birthday;  
  
    void describe() {  
        print(firstName);  
        int? birthYear = birthday?.year;  
        if (birthYear != null) {  
            print('Born ${DateTime.now().year -  
birthYear} years ago');  
        }  
    }  
}
```

Characters Package (Preview)

- allows access to the individual characters of a string
- navigate back and forth between them using a CharacterRange
- <https://pub.dev/packages/characters>

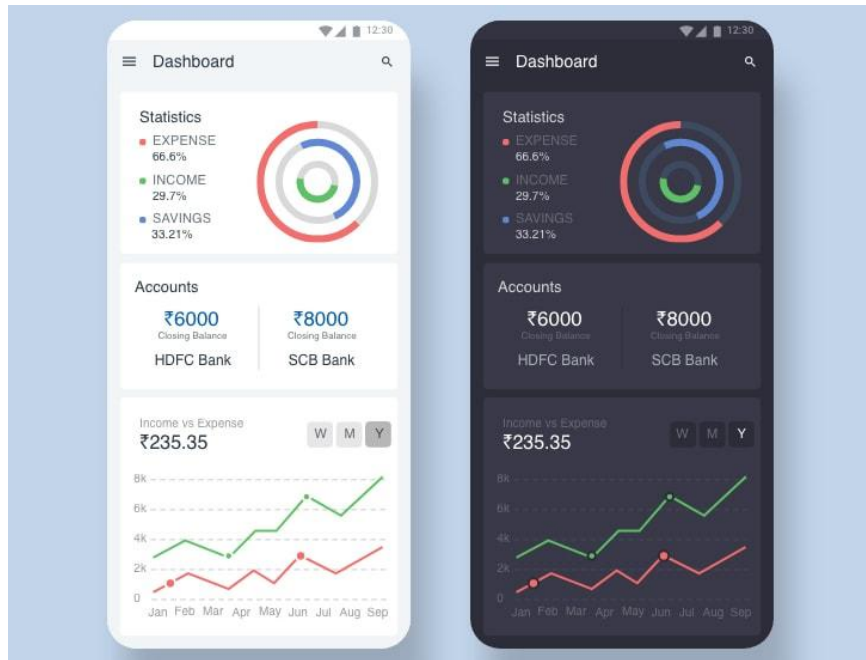
```
// prints first 3 characters of each string  
input.forEach((s) => print(s.characters.take(3)));
```



iOS 13 dark mode

- All the iOS-like widgets such as Cupertino are now available in a dark theme
- Simple implementation:

```
MaterialApp(  
  theme: ThemeData(  
    brightness: Brightness.light,  
    primaryColor: Colors.red,  
  ),  
  darkTheme: ThemeData(  
    brightness: Brightness.dark,  
    // additional settings go here  
  ),  
);
```

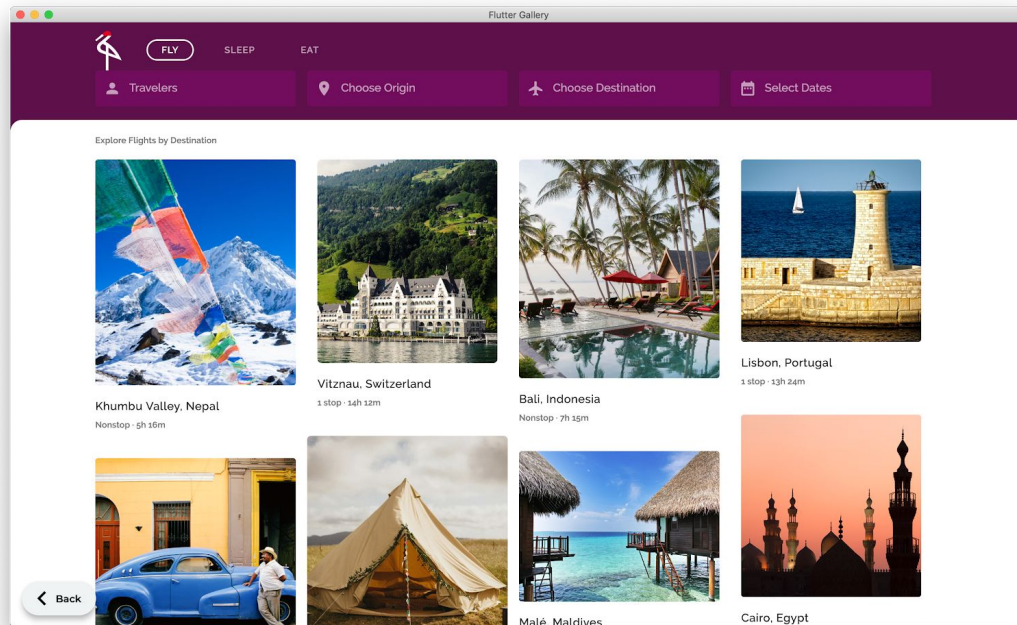


macOS support (alpha)

- available in master and dev channels
- material and cupertino design components
- keyboard navigation
- enable in terminal

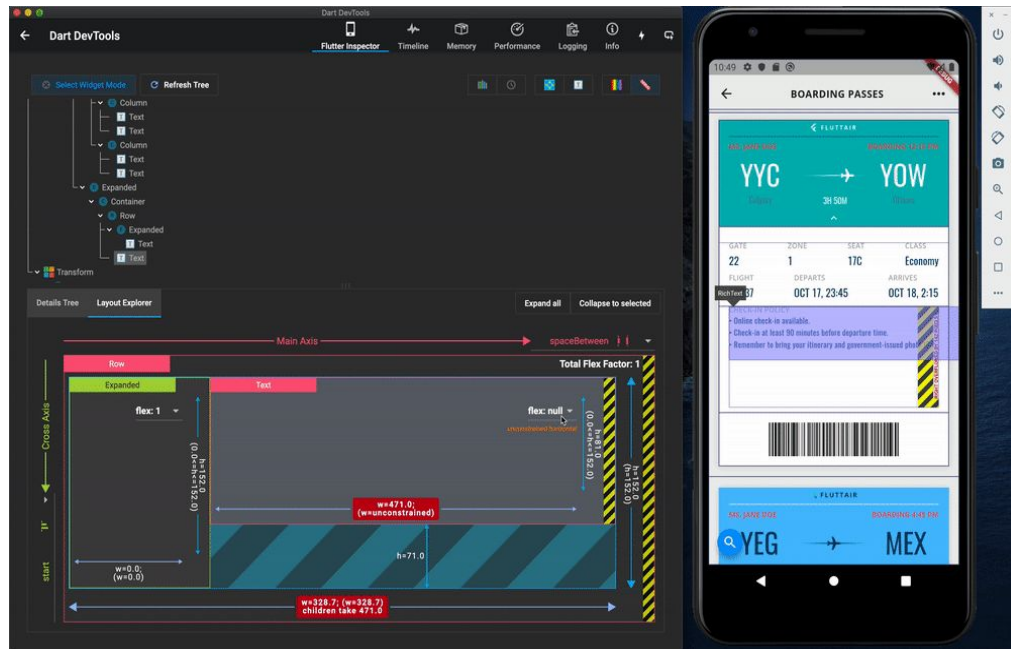
```
$ flutter config
```

```
--enable-macos-desktop
```



Layout Explorer (beta)

- Built in Flutter Inspector
- visual UI editing tool for Flex widgets (Row, Column, Flex)
- properties, like flex factor and alignment can be modified via dropdown lists in the explorer



Google Fonts Package

- https://pub.dev/packages/google_fonts
- 977 fonts (and their variants) from fonts.google.com

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

```
Text(  
  'This is Google Fonts',  
  style: GoogleFonts.oswald(  
    textStyle: TextStyle(color: Colors.blue,),  
  ),  
)
```



Android build improvements

- <https://flutter.dev/docs/development/packages-and-plugins/plugin-api-migration>
- Be careful because of third-party plugins



Android - upgrade to 1.12

1. Modify MainActivity.java / MainActivity.kt

```
class MainActivity: FlutterActivity() {  
-  override fun onCreate(savedInstanceState: Bundle?) {  
-    super.onCreate(savedInstanceState)  
-    GeneratedPluginRegistrant.registerWith(this)  
-  }  
+  override fun configureFlutterEngine(@NonNull flutterEngine: FlutterEngine) {  
+    GeneratedPluginRegistrant.registerWith(flutterEngine);  
+  }  
}
```

Android - upgrade to 1.12

2. Modify AndroidManifest.xml

application tag

- Remove the reference to `FlutterApplication` from the application tag.

Previous configuration:

```
<application
  name="io.flutter.FlutterApplication"
  >
  <!-- code omitted -->
</application>
```

New configuration:

```
<application
  >
  <!-- code omitted -->
</application>
```

Android - upgrade to 1.12

2.1. Update splash screen behavior

- Remove all `<meta-data>` tags with key

`android:name="io.flutter.app.android.SplashScreenUntilFirstFrame"`

- Add a launch theme to `styles.xml` that configures the desired launch screen as a background

Drawable:

```
<!-- You can name this style whatever you'd like -->
<style name="LaunchTheme" parent="@android:style/Theme.Black.NoTitleBar">
    <item name="android:windowBackground">@drawable/[your_launch_drawable_here]</item>
</style>
```

Android - upgrade to 1.12

- Add a normal theme that to `styles.xml` that should replace the launch screen when the Android process is fully initialized:

```
<!-- You can name this style whatever you'd like -->
<style name="NormalTheme" parent="@android:style/Theme.Black.NoTitleBar">
    <item name="android:windowBackground">@drawable/[your_normal_background_drawable]</item>
</style>
```



Android - upgrade to 1.12

2.2. Add a new `<meta-data>` tag under `<application>`

```
<meta-data
  android:name="flutterEmbedding"
  android:value="2" />
```

- allows third-party plugins to use the new Android embedding engine
- full docs at <https://github.com/flutter/flutter/wiki/Upgrading-pre-1.12-Android-projects>

WidgetsFlutterBinding.ensureInitialized()

- If you need to access the binary messenger before runApp() has been called (for example, during plugin initialization)
- <https://stackoverflow.com/q/57689492>

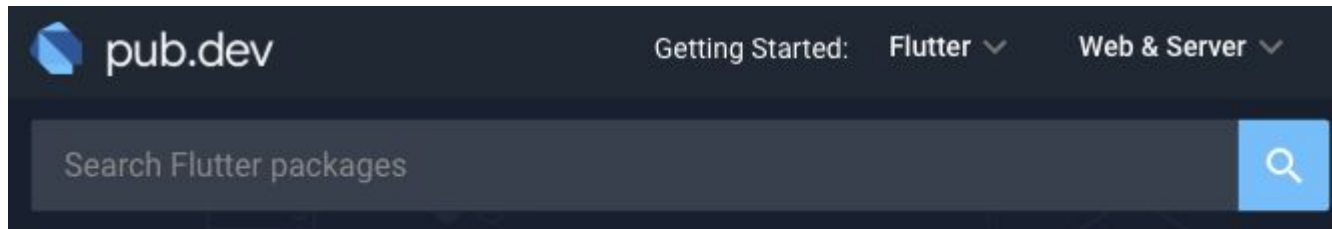


Issue on Huawei Y7 and few other devices

- Crash in v1.12.13+hotfix.5
- Lags in v1.12.13+hotfix.7
- <https://github.com/flutter/flutter/issues/47804>
- Works in v1.10.4



Liking 👍 packages on pub.dev



http 0.12.0+2

Published Apr 3, 2019 •  dart.dev



4 likes

DART

NATIVE

WEB

FLUTTER

ANDROID

IOS

WEB

Useful links

- <https://medium.com/flutter/announcing-flutter-1-12-what-a-year-22c256ba525d>
- <https://medium.com/dartlang/dart-2-7-a3710ec54e97>



Thank you

Ivan Celija: ivan@dream-implementation.com

Goran Kovač: goran@dream-implementation.com