LeanCode

Flutter Web & Desktop

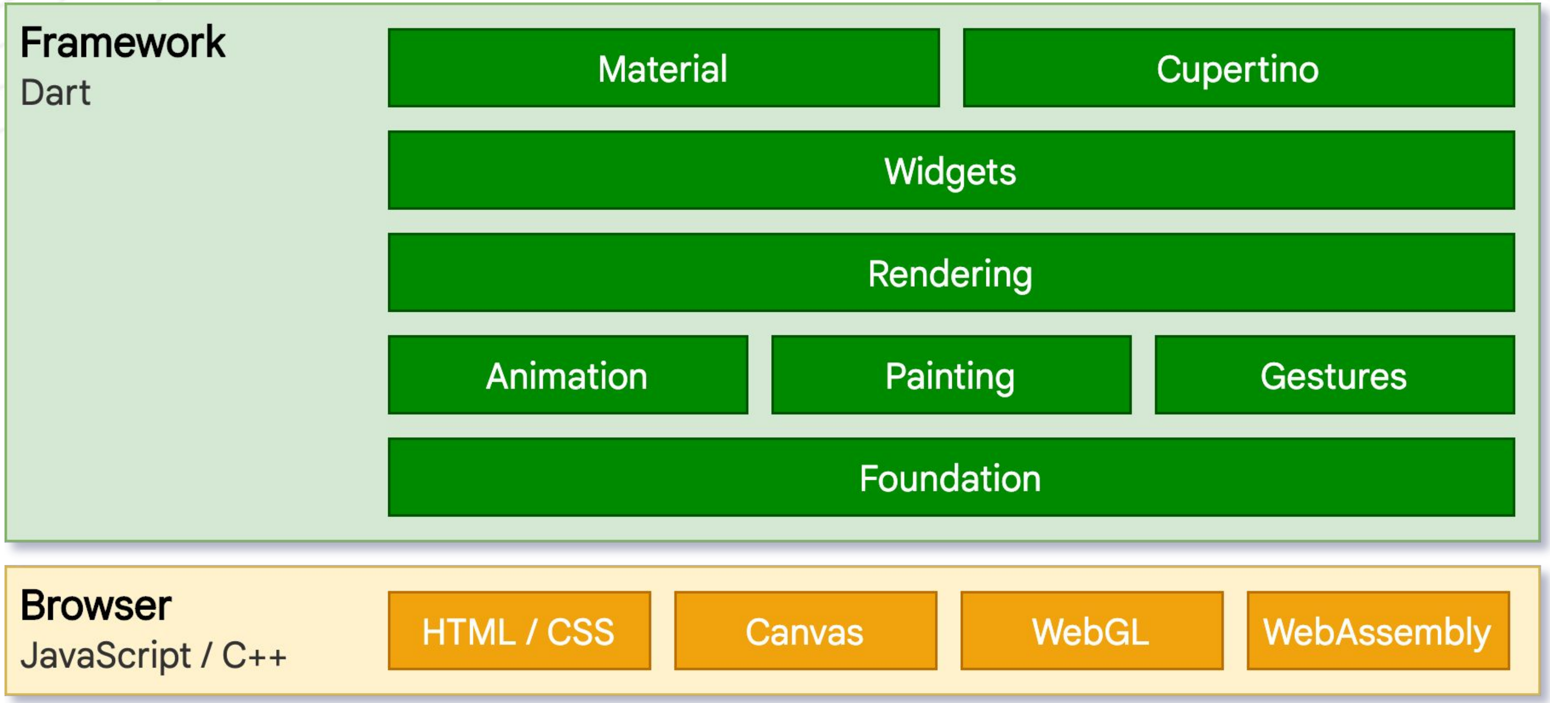
State of web and desktop in 2024

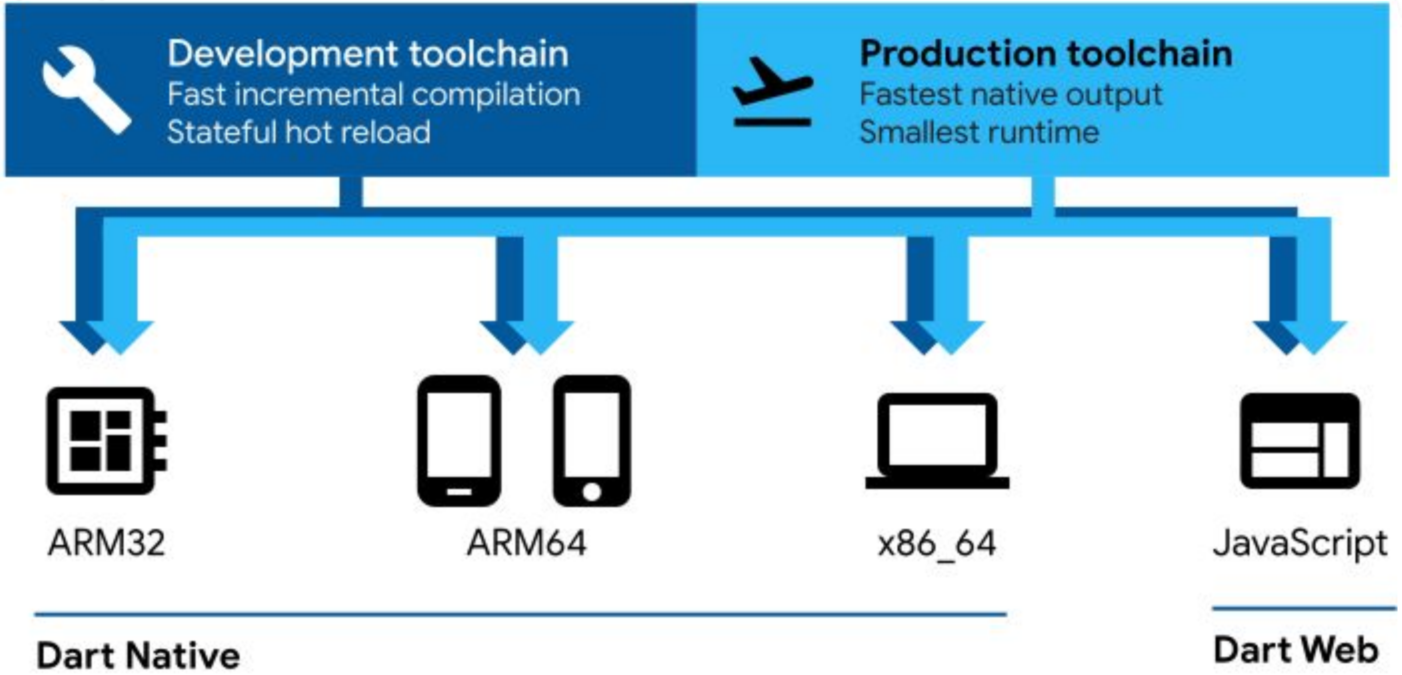
Flutter Web

We build digital products.

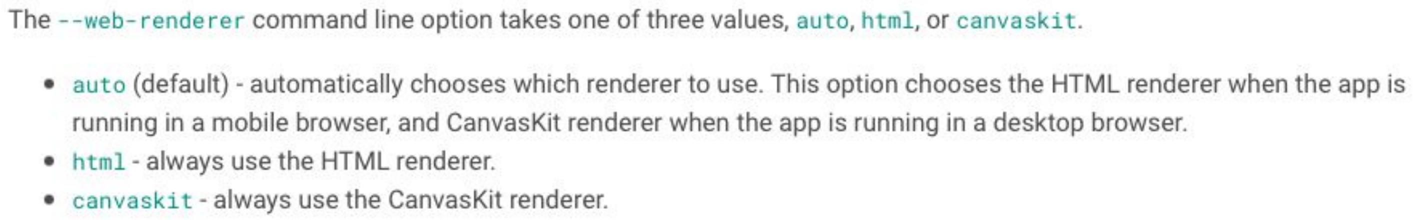
LeanCode

How it works


Framework
Dart
Material
Cupertino
Widgets
Rendering
Animation
Painting
Gestures
Foundation
Browser
JavaScript / C++
HTML / CSS
Canvas
WebGL
WebAssembly


Development toolchain
Fast incremental compilation
Stateful hot reload
Production toolchain
Fastest native output
Smallest runtime
ARM32
ARM64
x86_64
JavaScript
Dart Native
Dart Web

How it works



The `--web-renderer` command line option takes one of three values, `auto`, `html`, or `canvaskit`.

- `auto` (default) - automatically chooses which renderer to use. This option chooses the HTML renderer when the app is running in a mobile browser, and CanvasKit renderer when the app is running in a desktop browser.
- `html` - always use the HTML renderer.
- `canvaskit` - always use the CanvasKit renderer.

Use cases

1. Add web support for existing Flutter mobile app
2. SPA (Single Page Application)
3. PWA (but you might need some workarounds)

Things that can’t be used on web:

1. dart:io (you can use conditional imports or packages like universal_io)
2. dart:isolate
3. dart:ffi

Things work only on web:

1. dart:html
2. package:js
3. dart:js_util
4. dart:indexed_db
5. dart:svg
6. dart:web_audio
7. dart:web_gl

See this lint rule

Conditional imports

```
export 'src/hw_none.dart' // Stub implementation
    if (dart.library.io) 'src/hw_io.dart' // dart:io implementation
    if (dart.library.html) 'src/hw_html.dart'; // dart:html implementation
```

```
import 'dart:io';

void alarm([String? text]) {
  stderr.writeln(text ?? message);
}

String get message => 'Hello World from the VM!';
```

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

void main() {
  print(message);
}
```

```
void alarm([String? text]) => throw UnsupportedError('hw_none alarm');

String get message => throw UnsupportedError('hw_none message');
```

Conditional imports - dart:ui case

```dart
// You can export conditionally so that you don't have to
// conditionaly import later
export 'dart_ui_fake.dart' if (dart.library.html) 'dart:ui';
```

```dart
// dart_ui_fake.dart
class SomeClassFromDartUiThatWeMock {}
```

LayoutBuilder and other responsive widgets

```
body: LayoutBuilder(
  builder: (BuildContext context, BoxConstraints constraints) {
    if (constraints.maxWidth > 600) {
      return _buildWideContainers();
    } else {
      return _buildNormalContainer();
    }
  },
```

```
child: FractionallySizedBox(
  widthFactor: 0.5,
  heightFactor: 0.5,
  alignment: FractionalOffset.center,
  child: DecoratedBox(
    decoration: BoxDecoration(
      border: Border.all(
        color: Colors.blue,
        width: 4,
      ),
    ),
  ),
```

Image support on web

HTML renderer

- Uses <img> tag
- Delegates to the browser
- Can display images from arbitrary sources
- No control over image memory
- No support for ImageShader

CanvasKit renderer

- Uses WebGL to render images
- Subject to CORS policy
- Not optimized by the browser

https://docs.flutter.dev/platform-integration/web/web-images

Caveats

Hot reload doesn't work on web

https://github.com/flutter/flutter/issues/53041

Startup time (resolved)

https://github.com/flutter/flutter/issues/76009

main.dart.js too large and code splitting

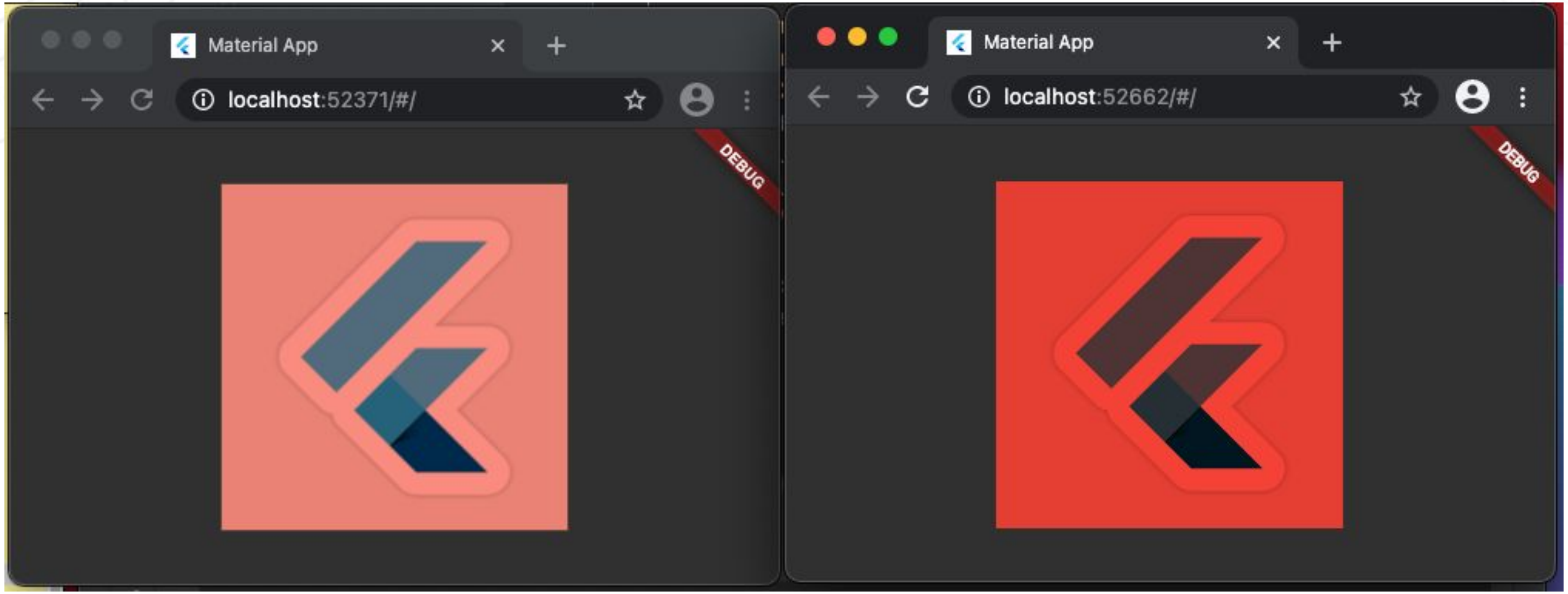
https://github.com/flutter/flutter/issues/46589

Deferred loading (web “only”)

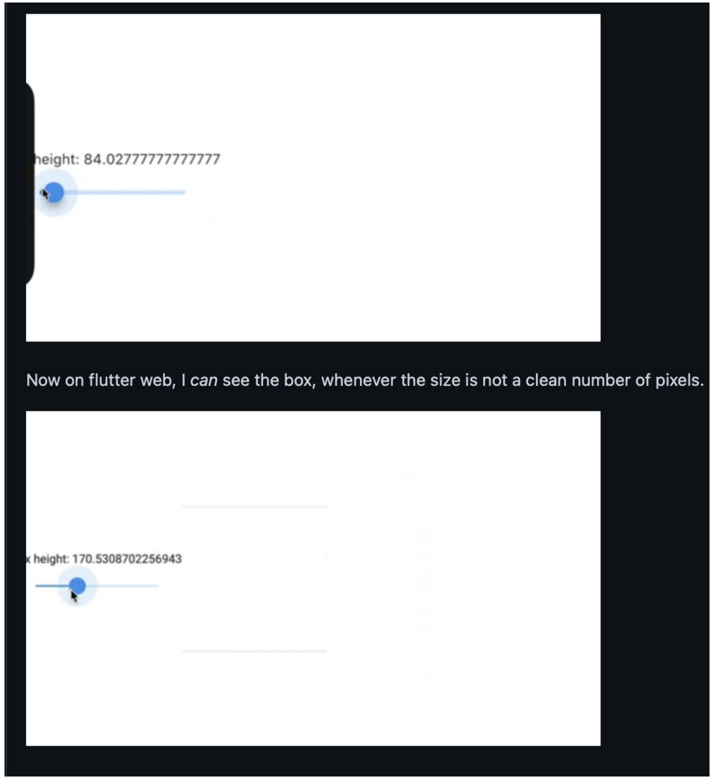
```
import 'package:greetings/hello.dart' deferred as hello;
```

```
Future<void> greet() async {
  await hello.loadLibrary();
  hello.printGreeting();
}
```

CanvasKit vs HTML differences - color blending

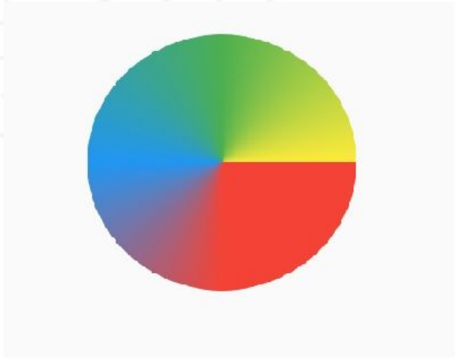


CanvasKit vs HTML differences - anti-aliasing

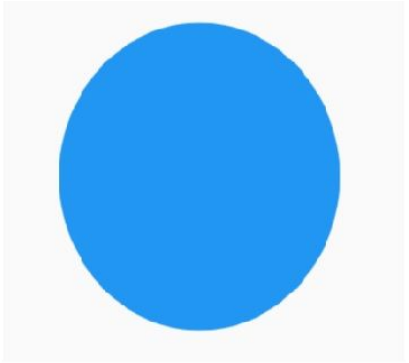


Now on flutter web, I *can* see the box, whenever the size is not a clean number of pixels.

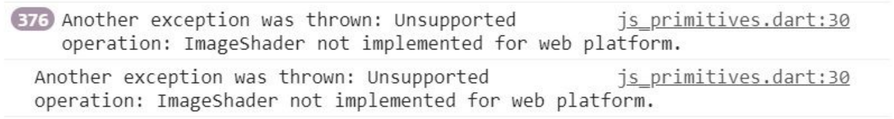
CanvasKit vs HTML differences - anti-aliasing



Canvas kit

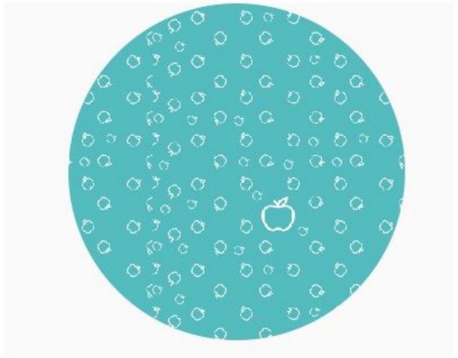


html



```
376 Another exception was thrown: Unsupported          js_primitives.dart:30
    operation: ImageShader not implemented for web platform.
  Another exception was thrown: Unsupported            js_primitives.dart:30
  operation: ImageShader not implemented for web platform.
```

html



canvas kit

https://github.com/flutter/flutter/issues/84984

SEO issues

https://github.com/flutter/flutter/issues/46789

Caching issues

https://github.com/flutter/flutter/issues/63500

Element embedding

https://docs.flutter.dev/deployment/web#embedding-a-flutter-app-into-an-html-page

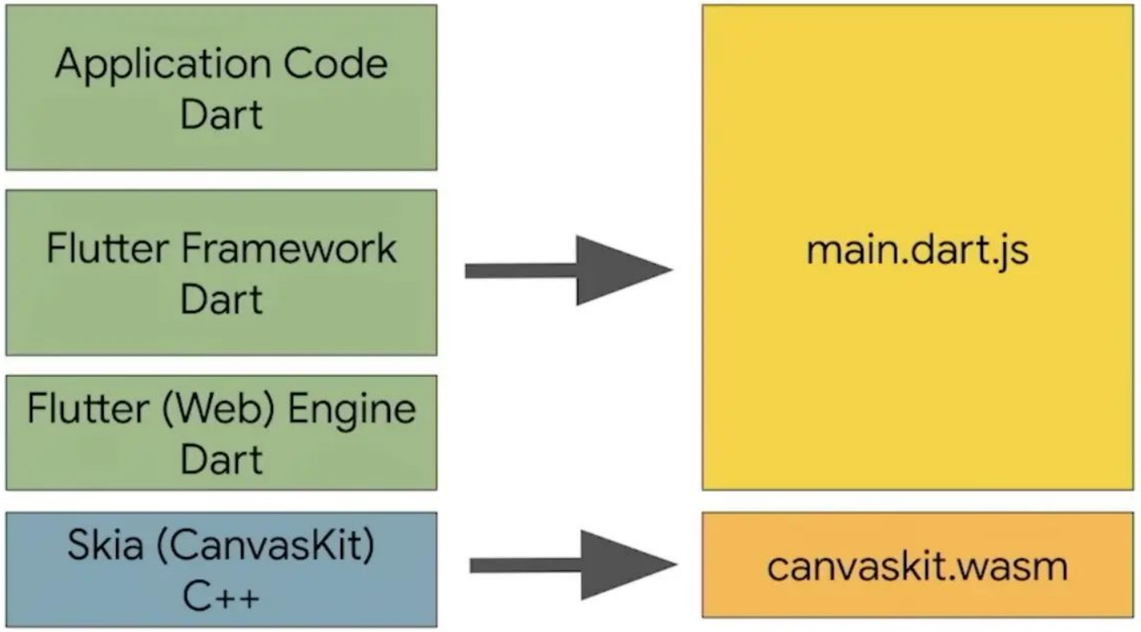
Multi-view element embedding not supported yet

https://github.com/flutter/flutter/issues/118481

What's next for Flutter on the Web?

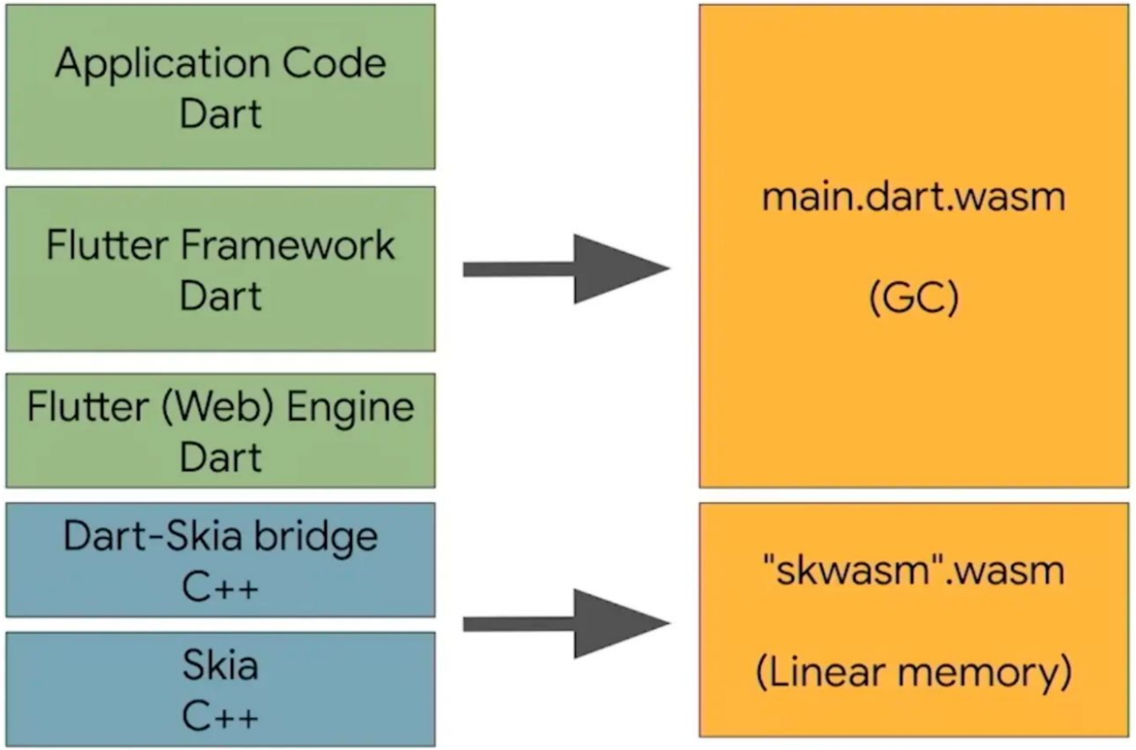
Starting with WebAssembly

State as of March 2023

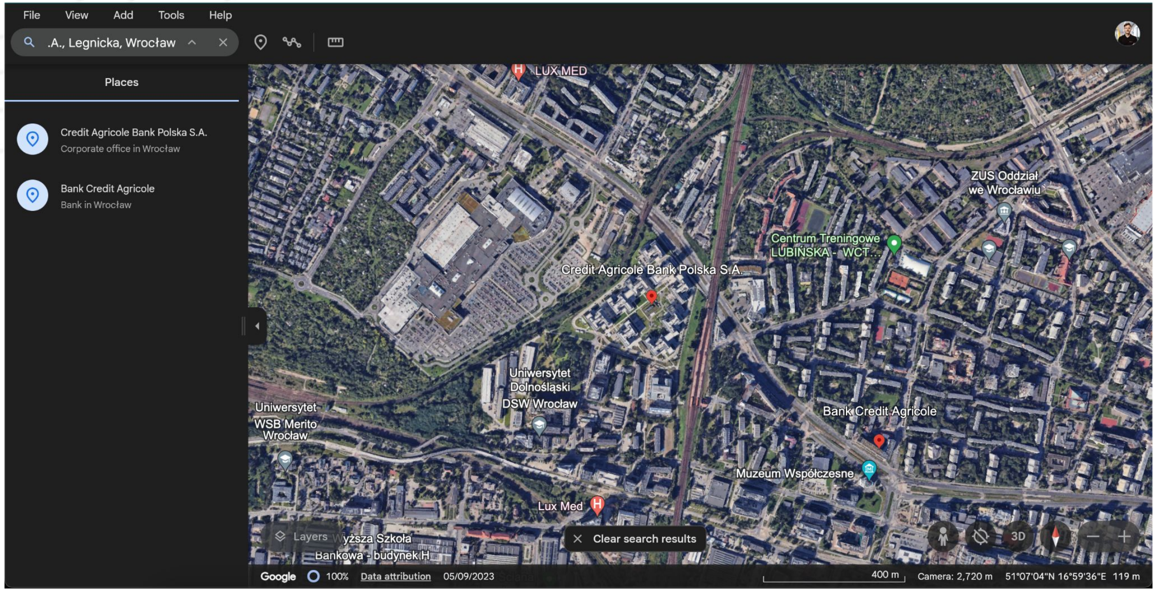


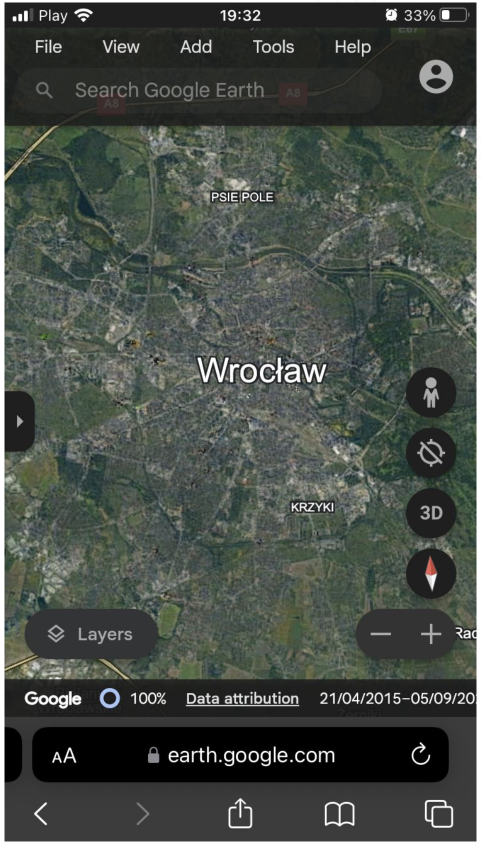
All-in on WebAssembly

Our wasm **plan** as of March 2023



Google Earth - earth.google.com





Uses Dart WebAssembly support

We build digital products.

LeanCode

Flutter Desktop

Support tiers

We define three tiers of support for the platforms on which apps built with Flutter might be deployed:

1. **Supported**
   Google-tested platforms that are automatically tested on every commit by continuous integration testing.
2. **Best effort**
   Platforms that we intend to support through coding practices, but are only tested on an ad-hoc basis.
3. **Unsupported**
   Platforms that we don't test or support.

Platform version	Supported	Best effort	Unsupported
Android SDK	21-34	19-20	18-
iOS	16	11-15, 17	10-, arm7v 32-bit
Linux Debian	10-12	9-	any 32-bit
Linux Ubuntu	20.04 LTS	20.10-23.04	any 32-bit
macOS	Ventura (13)	Mojave (10.14) to Monterey (12), Sonoma (14)	High Sierra (10.13-)
web - Chrome	latest 2 releases	96+	
web - Firefox	latest 2 releases	99+	
web - Safari	latest 2 releases	14+	
web - Edge	latest 2 releases	96+	
Windows	10	7, 8, and 11	Vista-, any 32-bit

Enable desktop support

```
$ flutter config --enable-windows-desktop # for the Windows runner
$ flutter config --enable-macos-desktop   # for the macOS runner
$ flutter config --enable-linux-desktop   # for the Linux runner
```

No PlatformView implementation

https://github.com/flutter/flutter/issues/41722
https://github.com/flutter/flutter/issues/31713

Multiple windows support

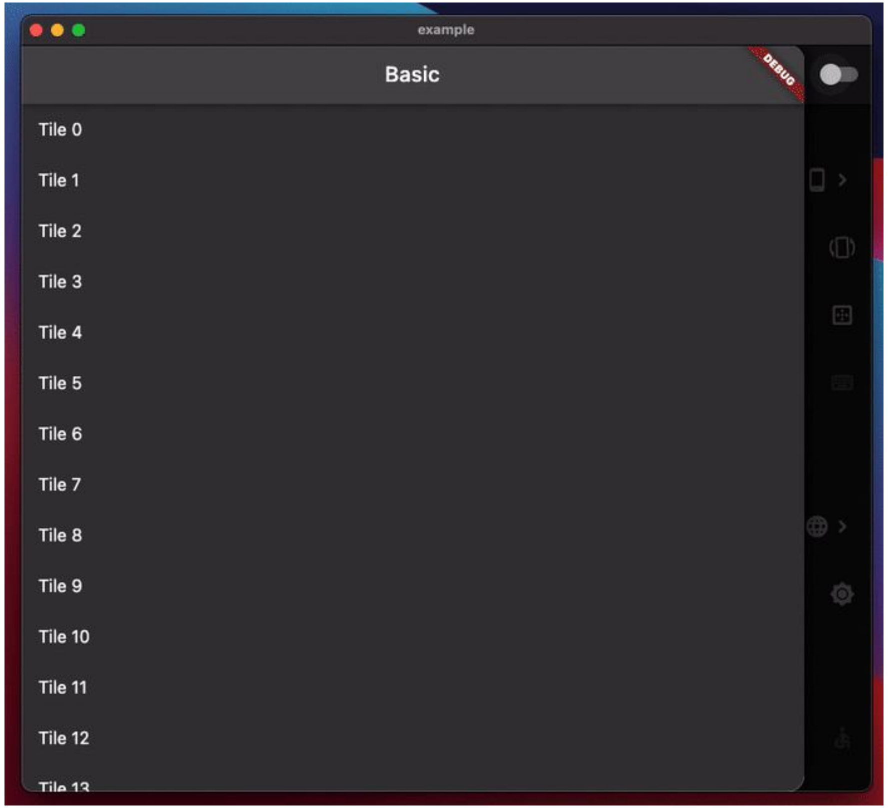
https://github.com/flutter/flutter/issues/30701

Check out all of the issues here

We build digital products.

LeanCode

device_preview


example

Basic

DEBUG

Tile 0

Tile 1

Tile 2

Tile 3

Tile 4

Tile 5

Tile 6

Tile 7

Tile 8

Tile 9

Tile 10

Tile 11

Tile 12

Tile 13

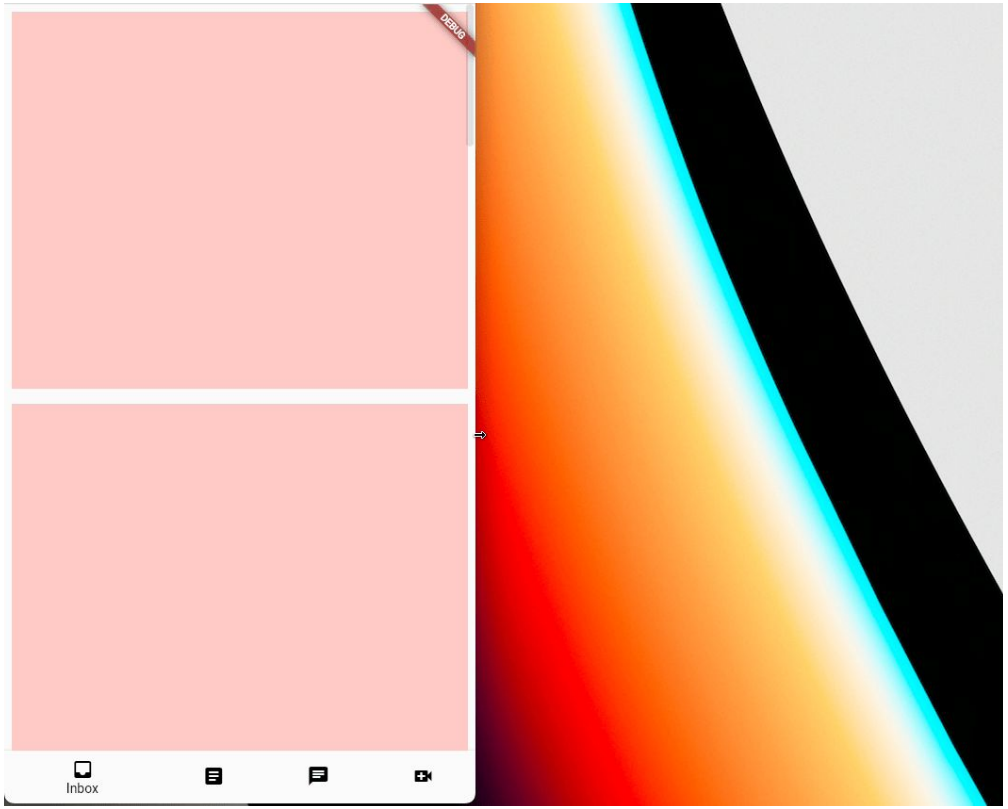
We build digital products.

LeanCode

Adaptive layouts

- `AspectRatio`
- `CustomSingleChildLayout`
- `CustomMultiChildLayout`
- `FittedBox`
- `FractionallySizedBox`
- `LayoutBuilder`
- `MediaQuery`
- `MediaQueryData`
- `OrientationBuilder`

flutter_adaptive_scaffold



Custom embedders

https://github.com/flutter/flutter/wiki/Custom-Flutter-Engine-Embedders

We build digital products.

Sources

Docs / flutter.dev

github.com/flutter/flutter

Kevin Moore @ Wasm I/O 2023

youtube.com/@romanjustcodes/