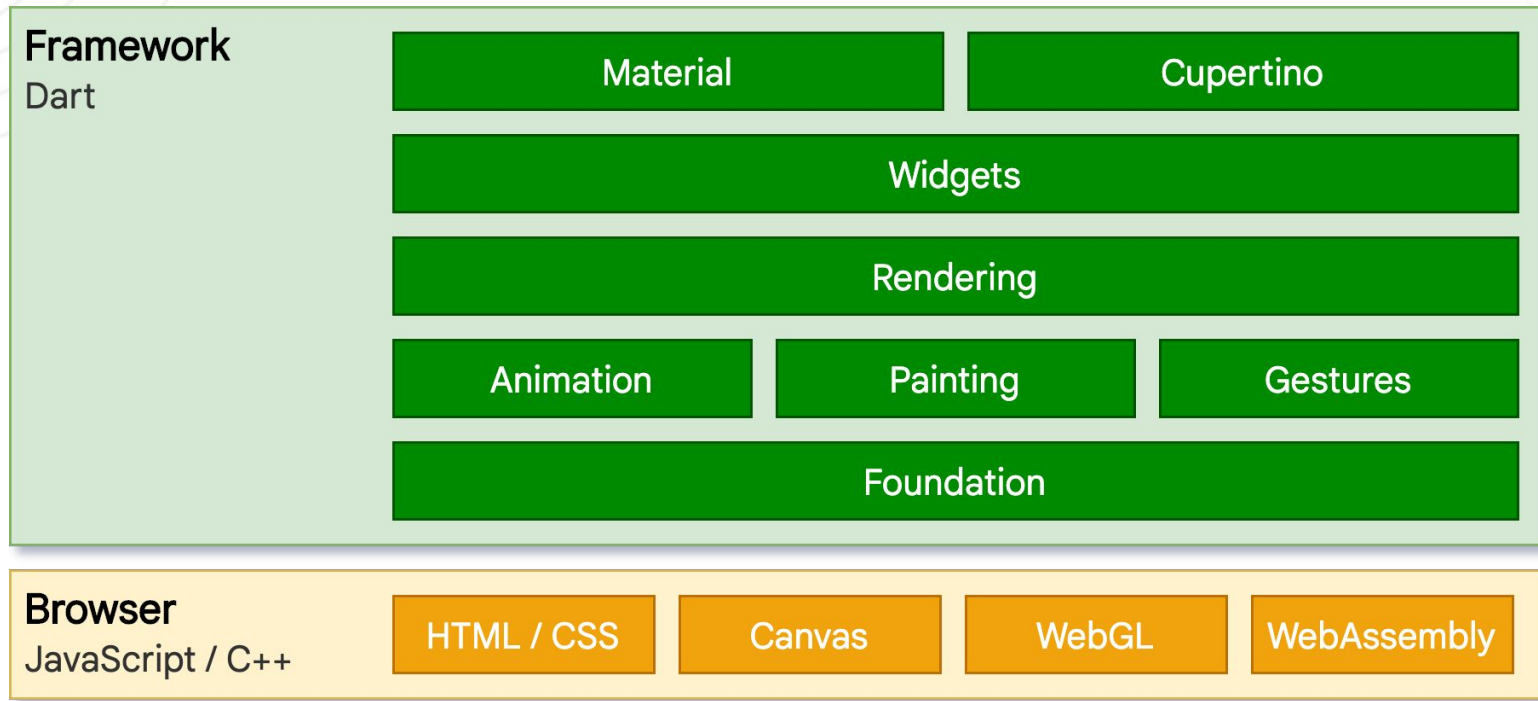


# Flutter Web & Desktop

State of web and desktop in 2022

# Flutter Web

# How it works





**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 need some workarounds)

# 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');
```

# 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,  
        ),  
      ),  
    ),  
  ),  
,
```





# Hot reload doesn't work on web



**Development toolchain**  
Fast incremental compilation  
Stateful hot reload



**Production toolchain**  
Fastest native output  
Smallest runtime



ARM32



ARM64



x86\_64



JavaScript

**Dart Native**

**Dart Web**



# Startup time

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

# **main.dart.js too large and code splitting**

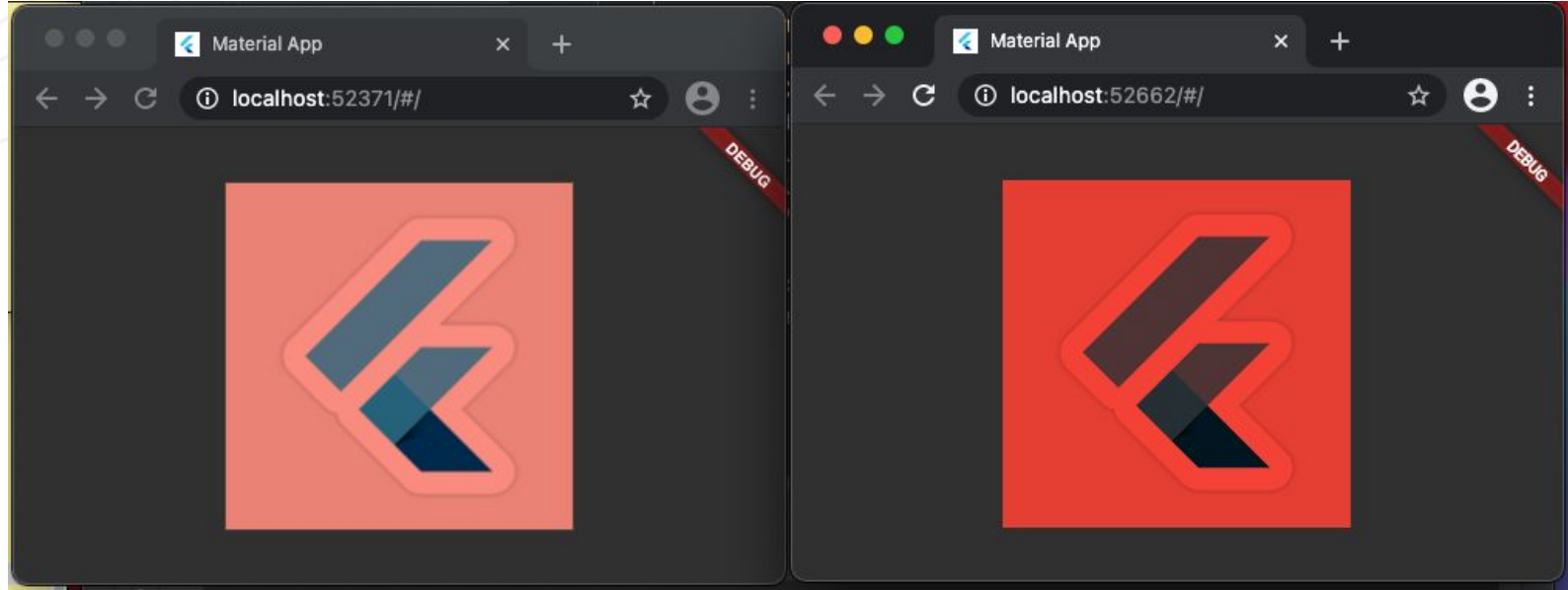
<https://github.com/flutter/flutter/issues/46589>

# Deferred loading (web “only”)

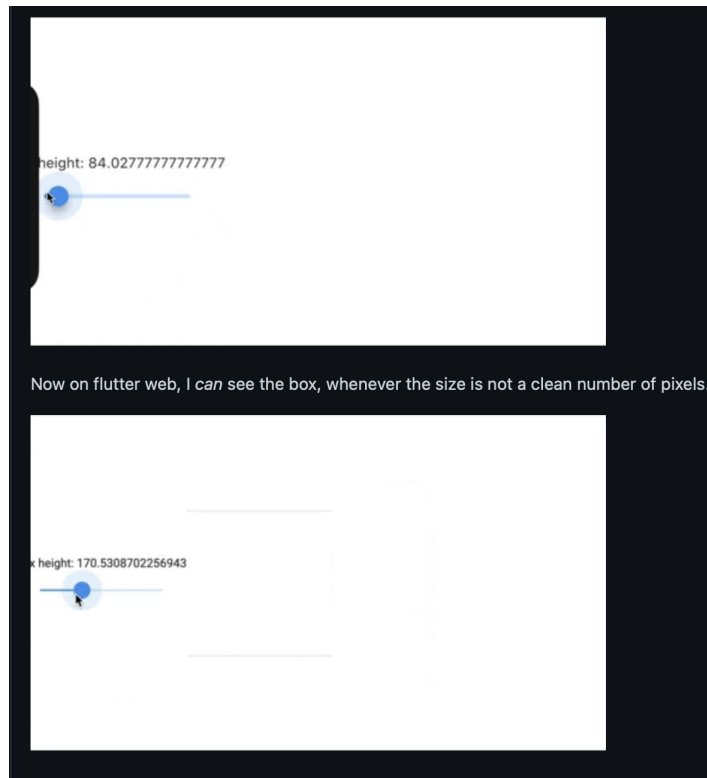
```
import 'package:greetings/hello.dart' deferred as hello;
```

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

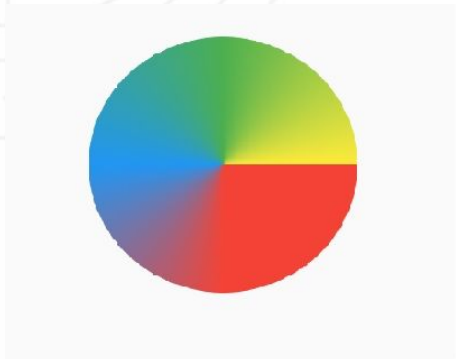
# CanvasKit vs HTML differences - color blending



# CanvasKit vs HTML differences - anti-aliasing



# CanvasKit vs HTML differences - anti-aliasing



Canvas kit



html

376 Another exception was thrown: Unsupported operation: ImageShader not implemented for web platform. [js\\_primitives.dart:30](#)

Another exception was thrown: Unsupported operation: ImageShader not implemented for web platform. [js\\_primitives.dart:30](#)

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>

# Flutter Desktop

# 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
```

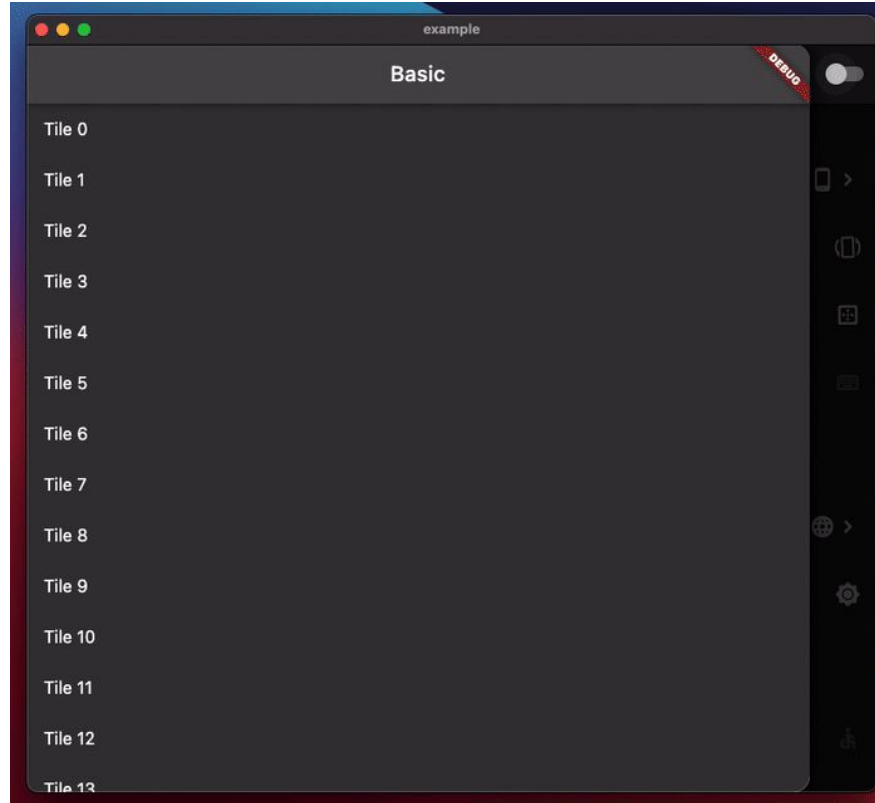
# Memory consumption issues

<https://github.com/flutter/flutter/issues/95092>

# Multiple windows support

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

# Device preview



# Adaptive layouts

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



# Adaptive layouts



# Sources

Docs / [flutter.dev](https://flutter.dev)

[github.com/flutter/flutter](https://github.com/flutter/flutter)