flutter

- [Changing the entry point of flutter](flutter run .\lib\my_other_main.dart)

Create .dmg

> In MacOS Run `Flutter build macos` Require Nodejs in mac We will use `node-appdmg` package

```
npm i -g appdmg
appdmg --version
or
npx appdmg --version
#  create a new file config.json
#  add the following content
```

```
{
  "title": "Flutter",
  "background": "background.png",
  "icon": "icon.icns",
  "contents": [
    {
      "x": 441,
      "y": 344,
      "type": "link",
      "path": "/Applications"
    },
    {
      "x": 441,
      "y": 434,
      "type": "file",
      "path": "Flutter.app"
    }
  ]
}
```

```
appdmg config.json Flutter.dmg
```

Packages

- Nice_intro
- Animations
- Modern Text Theming
- Fast Start
- New networking tool
- Accessibility & Internationalization
- New Dev Plugins
- Hot Reload is more resilient

Responsiveness

```
const   _kListItemExtent = 50.0;
const   _kMaxTimeWidth = 150.0;
const   _kGridViewBreakPoint = 550.0;

class AdaptiveContainer extends StatelessWidget{
    const AdaptiveContainer({Key key, this.names}):super(key:key);
    final list<String> names;

    @override
    widget build(BuildContext context){
        if(MediaQuery.of(context).size.width<_kGridViewBreakPoint){
            return ListView(
                itemExtent: _kListItemExtent,
                children: names.map<Widget>({string name} =>
AdaptedListItem(name:name)).toList(),
            );
        }else{
            return GridView.extent(
                maxCrossAxisExtent:_kMaxTimeWidth,
                children: names.map<Widget>({string name} =>
AdaptedListItem(name:name)).toList()
            );
        }
    }
}
```

```
LayoutBuilder(
    builder:(context,constraints){
        if(constraints.maxWidth>600){
            getWidgetLayout();
        }else{
            getNormalLayout();
        }
    };
)
```

```
OrientationBuilder(
    builder:(context,orientation){
        return GridView.count(
            crossAxisCount:orientation==Orientation.portrait?2:3,
        )
    }
)
```

Responsive Widget

1. AspectRatio
2. CustomSingleChildLayout
3. CustomMultiChildLayout
4. FittedBox
5. FractionallySizedBox
6. LayoutBuilder
7. MediaQuery
8. MediaQueryData
9. OrientationBuilder

Payments

- Stripe
- Razerpay
- cashfree
- paytm
- paypal

Android to iOS M1 Chip

> Add Firebase code if firebase is added in app run `flutter run` in ios go to ios/Podfile > "platform :ios, '10.0'" go to iOS folder in terminal now

```
sudo arch -x86_64 gem install ffi
arch -x86_64 pod install --repo-update
```

> run using xCode

Github Actions

```
# name: Deploy to Firebase Hosting on merge
'on':
  push:
    branches:
      - main
jobs:
  build_and_deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - run: npm ci && npm run build
      - uses: FirebaseExtended/action-hosting-deploy@v0
        with:
          repoToken: '${{ secrets.GITHUB_TOKEN }}'
          firebaseServiceAccount: '${{
secrets.FIREBASE_SERVICE_ACCOUNT_EXPERIMENTS_344B6 }}'
          channelId: live
          projectId: experiments-344b6
```

```
# name: Deploy to Firebase Hosting on PR
'on': pull_request
jobs:
  build_and_preview:
    if: '${{ github.event.pull_request.head.repo.full_name == github.repository }}'
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - run: npm ci && npm run build
      - uses: FirebaseExtended/action-hosting-deploy@v0
        with:
          repoToken: '${{ secrets.GITHUB_TOKEN }}'
          firebaseServiceAccount: '${{
secrets.FIREBASE_SERVICE_ACCOUNT_EXPERIMENTS_344B6 }}'
          projectId: experiments-344b6
```

Publish App on Play Store

> if you are using Firebase then for the signing key of play store go to App signing and paste the key in firebase General Project settings Get the new Google Services file and add it to your Flutter project

Subscriptions

RevenueCat

Docker

Simple Node serve for static files

```
web
    - index.html
    - main.js
    - style.css
Dockerfile
```

```
FROM node:lts-alpine
RUN npm -y -g install serve
```

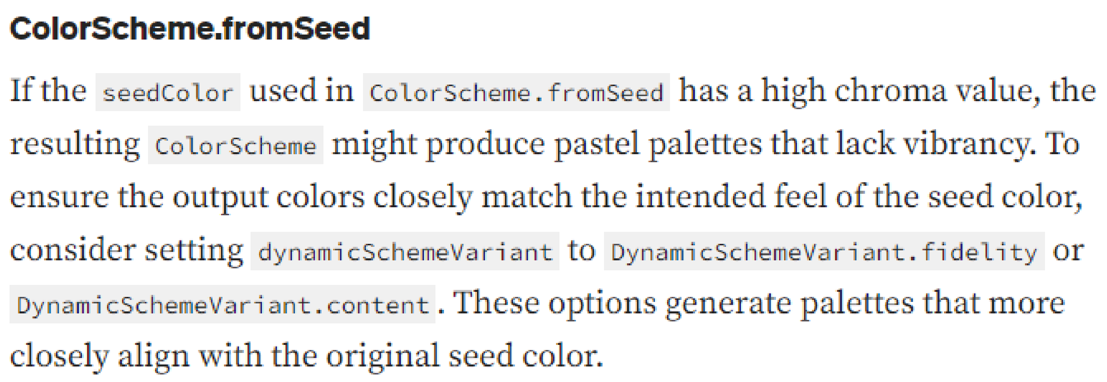
```
WORKDIR /app/
COPY . .

EXPOSE 3000
CMD ["serve", "web"]
```

- docker build -t my_docker_image -f .\Dockerfile .
- docker run -d -p 8080:8080 my_docker_image

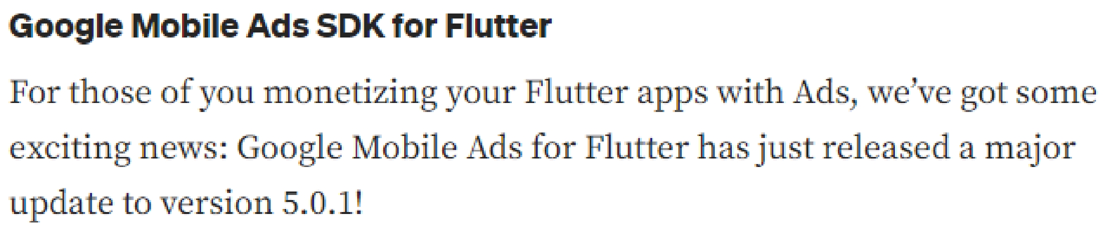
Currently, the supported version ranges are as follows:

Gradle — Fully supported 7.0.2 to current, warn otherwise AGP — Fully supported 7.0.0 to current, warn otherwise Java — Fully supported Java 11 to current, warn otherwise Kotlin — Fully supported 1.5.0 to current, warn otherwise In the next major release these warnings will become errors, which can be overridden with the flag `--android-skip-build-dependency-validation`



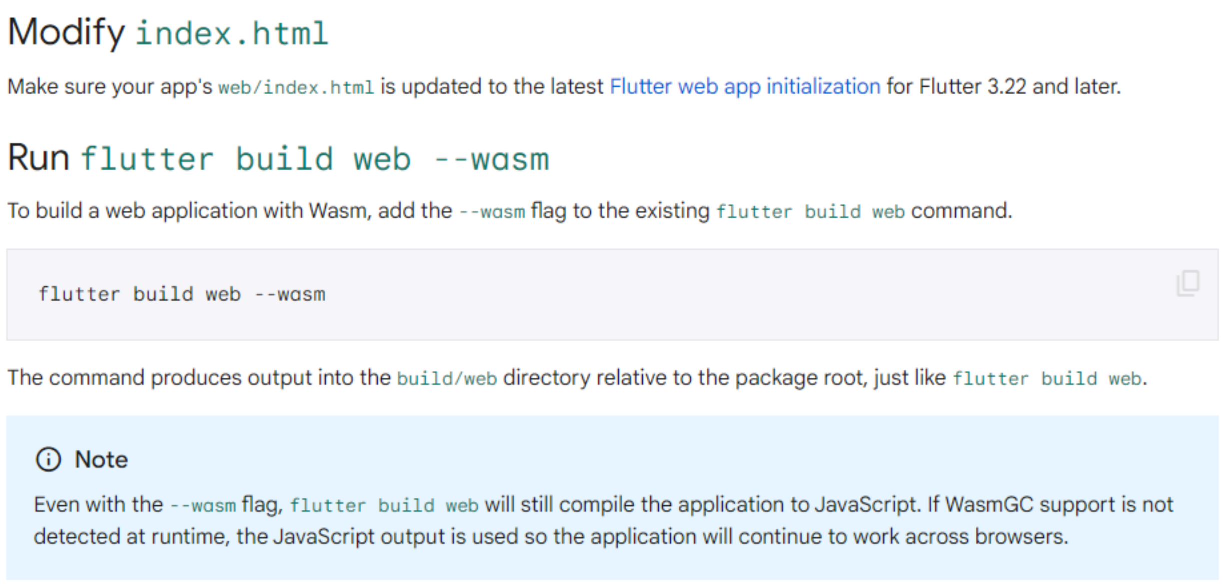
ColorScheme.fromSeed

If the `seedColor` used in `ColorScheme.fromSeed` has a high chroma value, the resulting `ColorScheme` might produce pastel palettes that lack vibrancy. To ensure the output colors closely match the intended feel of the seed color, consider setting `dynamicSchemeVariant` to `DynamicSchemeVariant.fidelity` or `DynamicSchemeVariant.content`. These options generate palettes that more closely align with the original seed color.



Google Mobile Ads SDK for Flutter

For those of you monetizing your Flutter apps with Ads, we've got some exciting news: Google Mobile Ads for Flutter has just released a major update to version 5.0.1!



Modify `index.html`

Make sure your app's `web/index.html` is updated to the latest Flutter web app initialization for Flutter 3.22 and later.

Run `flutter build web --wasm`

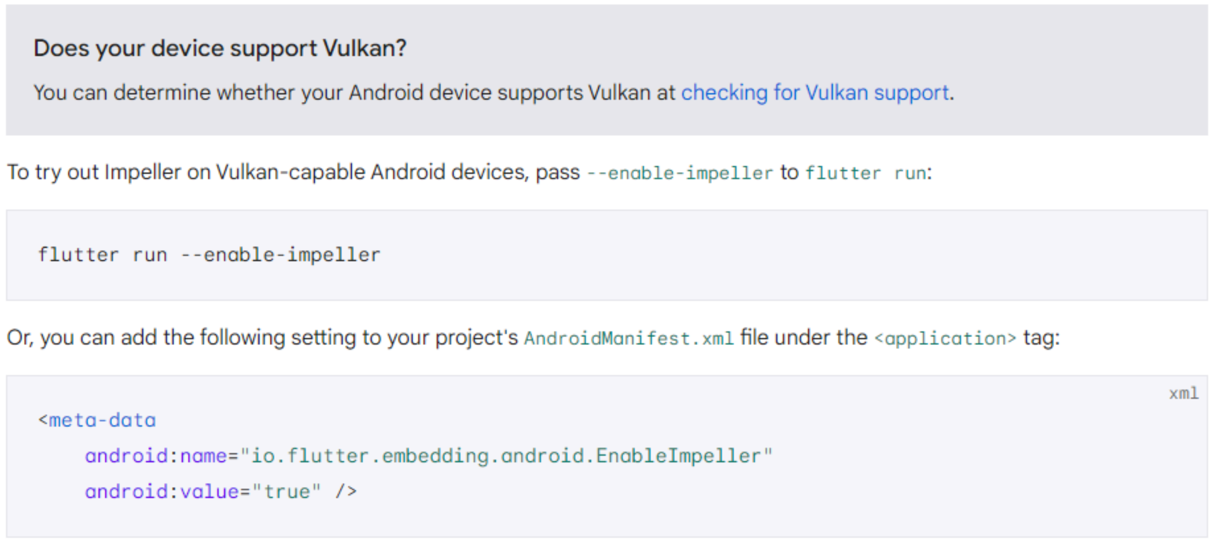
To build a web application with Wasm, add the `--wasm` flag to the existing `flutter build web` command.

```
flutter build web --wasm
```

The command produces output into the `build/web` directory relative to the package root, just like `flutter build web`.

ⓘ **Note**

Even with the `--wasm` flag, `flutter build web` will still compile the application to JavaScript. If WasmGC support is not detected at runtime, the JavaScript output is used so the application will continue to work across browsers.



Does your device support Vulkan?

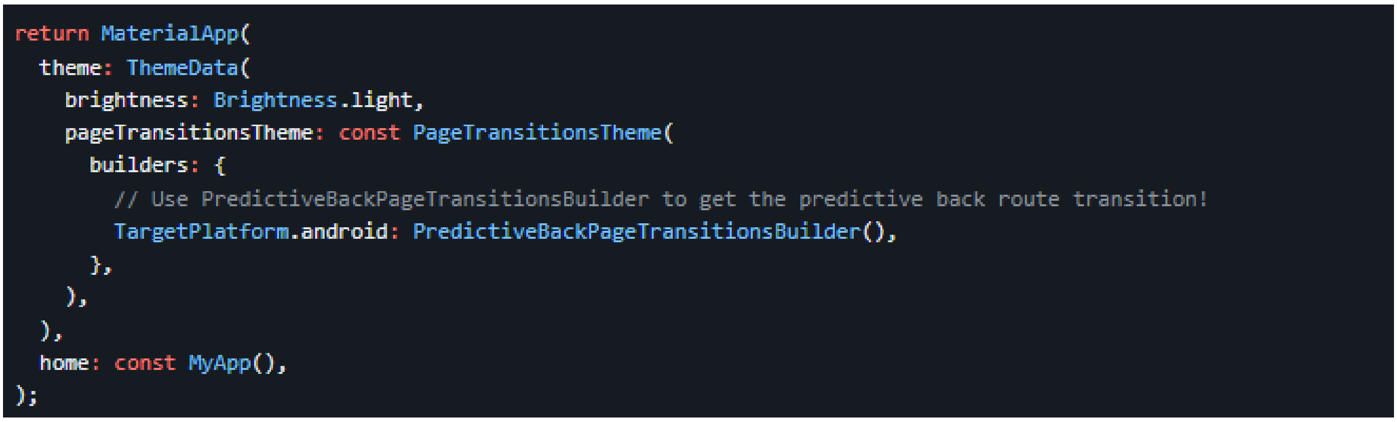
You can determine whether your Android device supports Vulkan at checking for Vulkan support.

To try out Impeller on Vulkan-capable Android devices, pass `--enable-impeller` to `flutter run`:

```
flutter run --enable-impeller
```

Or, you can add the following setting to your project's `AndroidManifest.xml` file under the `<application>` tag:

```xml
<meta-data
    android:name="io.flutter.embedding.android.EnableImpeller"
    android:value="true" />
```



```dart
return MaterialApp(
  theme: ThemeData(
    brightness: Brightness.light,
    pageTransitionsTheme: const PageTransitionsTheme(
      builders: {
        // Use PredictiveBackPageTransitionsBuilder to get the predictive back route transition!
        TargetPlatform.android: PredictiveBackPageTransitionsBuilder(),
      },
    ),
  ),
  home: const MyApp(),
);
```