

Flutter アプリ開発の第一歩

何ができる、どう作る。



自己紹介

名前

robo (兼高理恵)

@cch_robo



好きなもの

モバイル端末

おしごと

アプリの要件定義と仕様策定、設計から実装 & 改修まで

所属

GDG Kyoto (主催者) WTM Kyoto (主催者)

Flutter Osaka (staff) FlutterKaigi (staff)

おわび

この講座は、アプリ開発未経験の方を対象に
Flutterで何かしてみたい と思ってもらうためのものです。

このため技術的な話よりも、
Flutterがどこで使われているのか や

アプリ開発に踏み出す前に、**知っておくと良いこと**や、
アプリ開発現場の**苦勞と奮闘**を ~~笑って~~ 知ってもらうことで、

気負いなく取り組んでもらおうという企画です。

はじめに

Flutter は、世界中の製品で使われています。

Google I/O 2021 での正式報告によれば、
Google Play 新規アプリの20万件を占めるそうです。

同一コードでのマルチプラットフォーム対応と、
高い生産性 (コード量や開発期間の削減) により、
品質向上にリソースを割けられることが採用理由だそうです。

Google I/O 2021 Flutter 全体報告

<https://www.slideshare.net/cch-robo/google-io-2021-flutter>

<https://drive.google.com/file/d/1Hw14gA-9IbBXxRK0fUZgS83HHgeZ5Udn/view?usp=sharing>

Flutterとは

Flutterはアプリ開発を変革するフレームワーク

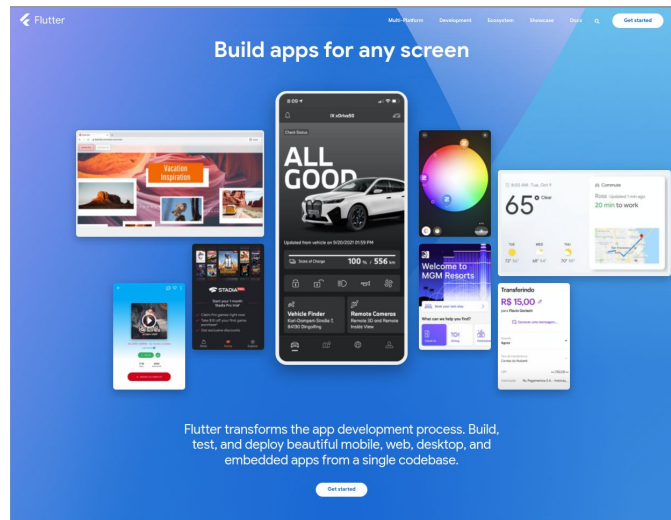
Flutter 公式サイトより

Flutter transforms the app development process.
Build, test, and deploy beautiful mobile, web, desktop,
and embedded apps from a single codebase.

Flutterはアプリ開発プロセスを変革するフレームワークです。

単一のコードベースから美しいアプリ（モバイル、ウェブ、
デスクトップ、組み込み）の構築とテストやデプロイを担います。

注: デプロイについては、各プラットフォームの知識が必要です。



Flutter - Build apps for any screen

<https://flutter.dev/>

Flutterの特徴

- **マルチプラットフォーム対応の向上**

単一コードから多環境のアプリ(iOS/Android、Web、Windows/Mac/Linux、IoT) が作れます。

- **処理速度の向上**

アプリは、プラットフォームのネイティブコードにコンパイルされます。
実装ロジックは、ARM や Intel のマシンコードとなり高速処理されます。

- **開発体験の向上**

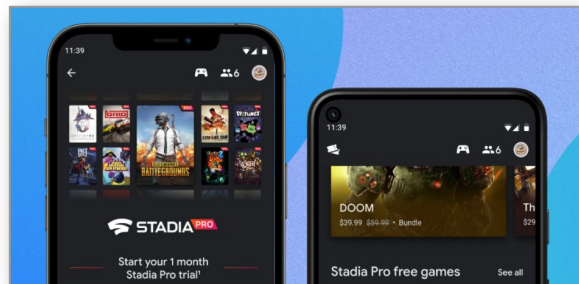
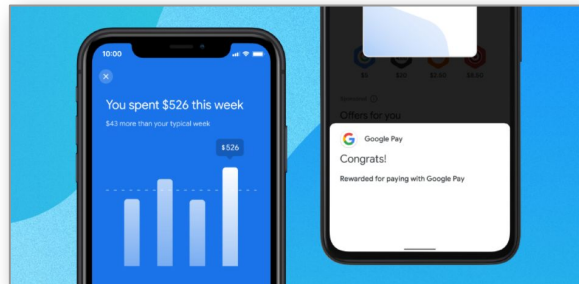
Hot reload (コード変更しても数秒で再実行可能)や メモリ使用量や実行時間を計測する **プロファイラ** と **デバッグツール**、**自動テスト**、**コード静的解析**、**コードフォーマッター**、IDEや各種エディタ(Android Studio/ IntelliJ/ VSCODE/ Emacs)での開発にも対応しています。

flutter は、オープンソースで開発されています。

Flutter採用例

flutter.dev Showcase より

- Google Pay
<https://flutter.dev/showcase/google-pay>
- Stadia
<https://flutter.dev/showcase/stadia>
- eBay
<https://flutter.dev/showcase/ebay>
- Nubank
<https://flutter.dev/showcase/nubank>



Google 内部でも利用されています。

flutter.dev Showcase より

- Alibaba Group
<https://flutter.dev/showcase/alibaba-group>
- ByteDance
<https://flutter.dev/showcase/bytedance>
- BMW
<https://flutter.dev/showcase/bmw>
- Toyota
<https://flutter.dev/showcase/toyota>



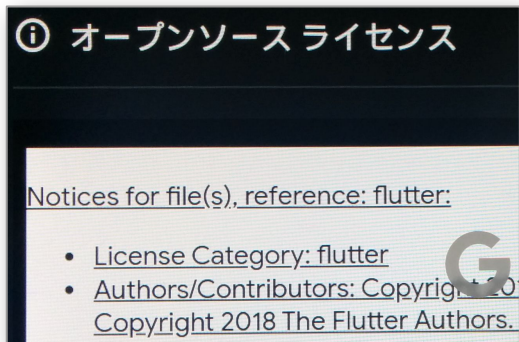
世界的企業でも利用されています。

Flutter製 UI を持つ組込デバイス

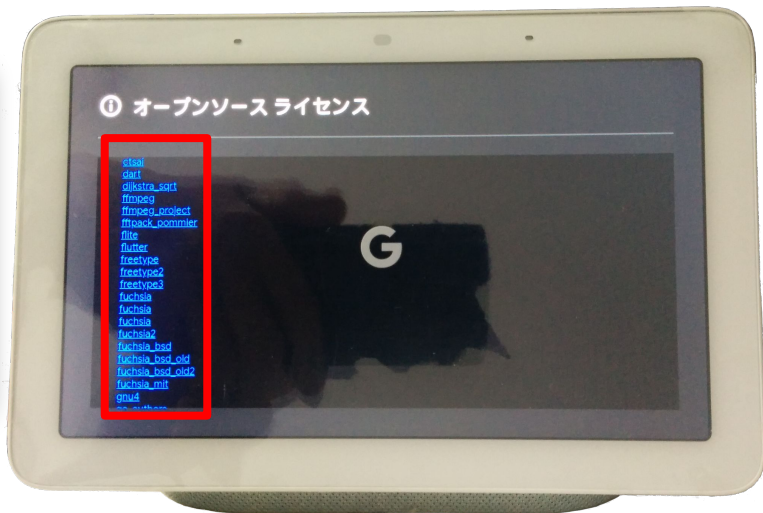
- Nest Hub

Googleの“第三のOS”「Fuchsia」、初代「Nest Hub」へ

<https://www.itmedia.co.jp/news/articles/2105/26/news054.html>



Flutterは、家庭内でも既に使われています。



[ctsai](#)
[dart](#)
[dijkstra_sqrt](#)
[ffmpeg](#)
[ffmpeg_project](#)
[fftpack_pommier](#)
[flite](#)
[flutter](#)
[freetype](#)
[freetype2](#)
[freetype3](#)
[fuchsia](#)
[fuchsia](#)
[fuchsia2](#)
[fuchsia_bsd](#)
[fuchsia_bsd_old](#)
[fuchsia_bsd_old2](#)
[fuchsia_mit](#)
[gnu4](#)

Flutter製のWebアプリ

- Google I/O 2021 - I/O Photo Booth

<https://photobooth.flutter.dev/>

<https://github.com/flutter/photobooth>

How It's Made: I/O Photo Booth

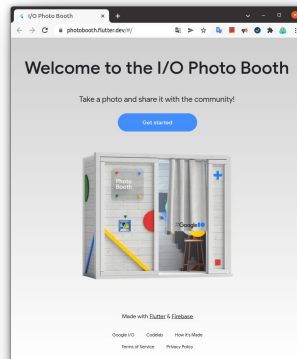
<https://medium.com/flutter/how-its-made-i-o-photo-booth-3b8355d35883>

- FlutterKaigi 2021 official website

<https://flutterkaigi.jp/>

<https://github.com/FlutterKaigi/confwebsite2021>

これらはオープンソースです。



Flutter製プロダクト

国内編

Flutterは、既に国内でも実用段階にあります。

Flutter製の国内プロダクト

アプリは、ユーザとサービスとの**フロントエンド事業ツール**です。
ユーザの関心事はサービス内容であり、その裏側ではありませんし、
事業者の責務はその課題解決ですので手段の主役化は失礼でしょう。

このためここで紹介するアプリは、
事業者から開発者に向けて Flutterを採用していることを公表した
ごく一部の事例紹介であり、参考程度の情報と見てください。

Flutter製の国内プロダクト

- GO Dine (App Store, Google Play)
Flutterで作るタクシーデリバリーアプリ「GO Dine」
<https://lab.mo-t.com/blog/mot-online-techtalk-5>
- NewsDigest (App Store, Google Play)
爆速開発を目指して NewsDigest を Flutter にリプレイスします
<https://tech.jxpress.net/entry/2021/12/03/121639>
- YOUTRUST (App Store, Google Play)
YOUTRUSTアプリの構造、暴露してみた
<https://speakerdeck.com/daiki1003/youtrustapurifalsegou-zao-bao-lu-sitemita>
【YOUTRUST】Flutterアプリエンジニア募集！

Flutter製の国内プロダクト

- Chompy (App Store, Google Play)
Chompy でFlutter採用するまでと、その振り返り
<https://speakerdeck.com/yagitatsu/chompytefluttercai-yong-sita-zhen-rifan-ri>
- youbride (App Store, Google Play)
歴史ある婚活サービスyoubrideがFlutterを採用しました
<https://developer.diverse-inc.com/entry/2019/07/04/173000>
- NOIN (App Store, Google Play)
Flutterほぼ未経験チームがネイティブアプリを半年でFlutterにリプレイスした話
<https://drive.google.com/file/d/1NmUTwjvs9aebcKz9awwoEU-Re11kVcPj/view>
<https://droidkaigi.jp/2021/timetable/277232/?day=2>

Flutter製の国内プロダクト

- イトーヨーカドーネットスーパー (App Store, Google Play)
ライフネットスーパー (App Store, Google Play)
「なぜDartなのか」CTO石川が技術選定に込めた想いとその背景
<https://10x.co.jp/articles/posts/?id=2020-06-22>
- Amebaマンガ (App Store, Google Play)
WRESTLE UNIVERSE (App Store, Google Play)
WINTICKET (App Store, Google Play)
9プロジェクトが同時に走る サイバーエージェントのクロスプラットフォームアプリ開発 2021
<https://developers.cyberagent.co.jp/blog/archives/33996/>

クロスプラットフォームのメリットと課題も添えられています

Flutterでアプリを作る心得

アプリ開発に必要なもの。

Flutterでのアプリ開発に必要なもの

Flutter は、アプリ開発のためのフレームワークです。

プラットフォーム iOS/ Android/ Web/ Windows... の違いを吸収してくれますが、
アプリ開発はそれだけで完結しません。

アプリ開発を行うには、

SDK という開発キット、**IDE** というコード編集 & 開発支援ツールの
インストールや使い方、

Widget という アプリコンポーネントと 宣言的 UI の使い方を学び、

Dart というプログラム言語で やりたいことのロジックをコードにする
総合的な知識と論理力が必要です。

学びの基本は、Flutterと Dartの公式サイト

Flutterと Dart言語には、公式サイトがあります。

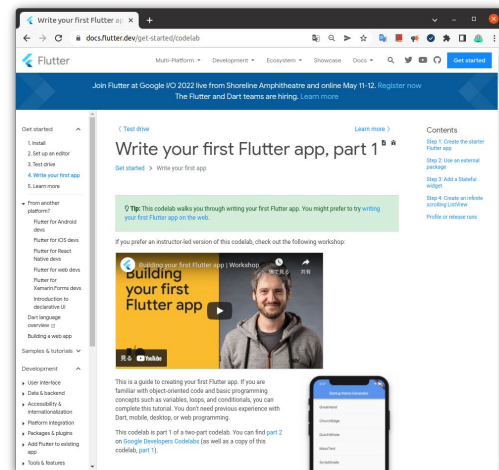
Flutter 公式サイト ⇒ <https://flutter.dev> には、**SDK** や **IDE** のインストールから **Widget** の使い方、**テストの実行**や、iOS / Android 固有の**リリース方法**...などアプリの作成に必要な**基礎知識**が説明されています。

Dart 言語公式サイト ⇒ <https://dart.dev> には、言語仕様やコアライブラリの紹介と、簡単な使い方が説明されています。

Codelabs では、アプリの作り方やコードの書き方がハンズオン形式で学べます。

Google Codelabs

<https://codelabs.developers.google.com/?product=flutter>



Flutter 入門第一歩のページ

Flutter公式サイトには、

Flutter SDK と **IDE** のインストールから、IDE でのアプリ雛形の作成やアプリの実行ならびに **Hot reload** でのコード変更の素早い確認の仕方の紹介。さらに実際にコードを書いてアプリを段階的に作成していくハンズオンという、

Flutter 入門者用のページが用意されています。

- install ⇒ <https://docs.flutter.dev/get-started/install>
- Setup an editor ⇒ <https://docs.flutter.dev/get-started>
- Test drive ⇒ <https://docs.flutter.dev/get-started/test-drive>
- Write your first Flutter app ⇒ <https://docs.flutter.dev/get-started/codelab>
- Learn more ⇒ <https://docs.flutter.dev/get-started/learn-more>

こちらのページで Flutter 入門してみましょう。

コミュニティサポート

Flutter は、コミュニティサポートも厚く、
Flutter Meetup Network という公式の 地域密着型ユーザグループがあります。
日本も 東京/大阪で、定期的に**勉強会**や**ハンズオン**、**もくもく会**を開いています。

- **Flutter Meetup Tokyo**

<https://www.meetup.com/ja-JP/flutter-meetup-tokyo/>

- **Flutter Meetup Osaka**

<https://www.meetup.com/ja-JP/flutter-meetup-osaka/>



Flutter Meetup では、ノウハウの共有や知見の公開が行われています。
アプリの作り方に悩んだときは、参加されてみてはいかがでしょうか。

コミュニティサポート

Flutter Meetup だけでなく、
GDG Japan でも Flutter に関する講座や、
ハンズオンを行っています。

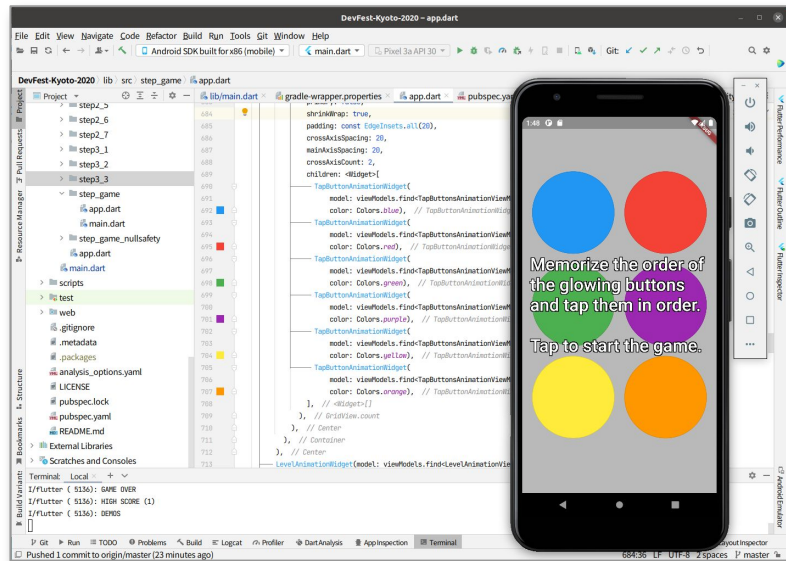
段階的にタイマーやアニメーションを学び
右図のようなミニゲームを作る講座資料も
ありますので御参照ください。

DevFest-Kyoto-2020 リポジトリ

<https://github.com/cch-robo/DevFest-Kyoto-2020>

<https://dartpad.dev/embed-flutter.html?id=77ee97e93bc43e3a91c1e78a4015ea87&split=80&theme=dark>

こちらのリンクを開くと DartPad でミニゲームが遊べます。



アプリ開発は、支援者がいるマラソンです。

アプリの開発は、総合的な知識と論理力、時には忍耐が必要です。
コンピュータは、コンピュータにできることを言われた通りにしかできないからです。
やりたいことを**アプリ**(コンピュータ)にやらせるには、
コンピュータ(Flutter)にできることと、コンピュータ(Flutter)への伝え方を学び、
やりたいことを自分自身が **理路整然とまとめあげ**(ロジック化) **最適化**して、
コンピュータが判る言葉(Dart 言語コード)にして伝え続けなければなりません。

ここで紹介したことは、長い道のりの第一歩を踏み出すための情報です。
幸いなことに Flutter 日本語資料は、ネット上にたくさんたまって来ました。

自分で手を動かしてコードを学び、コミュニティに参加してみてください。

Flutter開発現場のこぼれ話

アプリ開発は大変ですが楽しいものです。

業務としてのアプリ開発

事業者から見れば、アプリは事業を継続発展させるためのツールでしょう。このため事業の成長によるスケールアップ、信頼性確保のための高品質化、ニーズの変化による方針変更から、**継続改修**できることが求められます。

解りやすく変更や拡張を容易にする設計が大切なのはこのためです。

設計の前段階として、願望ではない**要件を仕様として定義**します。**要件と願望**の線引きは、**求めることと対策**が見えているか否かでしょう。

- 願望や観念的な要求は、取り組むべき対象が具体化できません。
- 1つ1つは具体化できても、全体で矛盾するものは実現できません。

メンバー全員
との対話が大切

システムや業務に関する総合的な知識と、
対策を理路整然と導出できる論理力が求められます。

理想論です...

Flutter アプリ開発業務の辛いところ

- プラットフォームに詳しいメンバーやアドバイザーの確保が必要
成果物は、iOS/Android アプリとして公開されます。(Flutterは黒子です)
各プラットフォーム提供のサービスは、取り扱いや考え方が異なります。
例) App Store/Google Play へのリリース、アプリ内購入、Push通知...etc
- 中庸的なアプリは作りやすいが、強い拘りがある場合は大変
プラグインや対策が確立している**一般的なこと**は簡易に実現できます。
iOS/Android の新機能や、iOS/Android 専用サードパーティライブラリは、
自分や誰かが**「頑張ってプラグインを作る」**ことがないと対応できません。
ネイティブ機能を駆使する必要のある**「強い拘り」**がある場合も然りです。

Flutter アプリ開発業務の辛いところ

- 進化が速いので、時々システムやライブラリに**破壊的変更**がある。
これは安全や品質向上のため、iOS / Android でも発生する宿命ですね。

- プラットフォーム都合が強いと、ロジック共通化メリットが少ない

App Store/Google Play 都合でリリース方法が異なることを紹介したように、コア事業において、プラットフォームごとの外部都合の影響が強い場合、UI コード共通化の他に、**ロジック共通化コスト**の意味が見いだせません。

プラットフォームごとにユーザ対策を替えたい場合もありますでしょうし、
各アプリを同一プロジェクトにするか否かは場合によるでしょう。

Flutter アプリ開発業務の希望的観測

...とはいえ、

Flutterは、

プラットフォームごとの事情を吸収する基盤として、
事業のコアロジックを浮かび上がらせ、集中させてくれます。

たとえ別プロジェクトでになっても、
UI Widgetや 表示ロジックの共有はできるでしょう。

そして各プラットフォームについても
私は学び続けたいと思っています。

プロダクトの作り方は自由です。

アプリを作る理由は、
個人的な利便性(シラバスを手元で確認したいなあ)や、
事業のためユーザとのサービス・フロントエンドを作りたい
...など様々です。

このため**正式な方法論**は場合によりけりでしょう。

事業のための開発なら、
企画, 市場調査, 目標設定, 設計, 実装, テスト, リリース, レトロスペクティブ
...のサイクルを繰り返すのですが
個人開発なら 自由に作りましょう。

ご清聴、
ありがとうございました。

情報修正があれば更新します
最新版スライドは [こちらから](#)

