

Flutter

TechGuide - Alura, FIAP e PM3

Flutter

Level 1

Dart - Fundamentals:

Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.

Installing and configuring Dart

Learning the Dart keywords

Building terminal applications using Dart

Understanding how dynamism works in Dart

Understanding and learning how to use Null Safety in Dart

Contents

WebSiteBeginning with Dart (<https://dart.dev/get-dart>)

WebSiteA tour of the Dart language (<https://dart.dev/guides/language/language-tour>)

WebSiteThe Dart type system (<https://dart.dev/guides/language/type-system>)

WebSiteUnderstanding null safety (<https://dart.dev/null-safety/understanding-null-safety>)

Article	Why Flutter Uses Dart (https://medium.com/hackernoon/why-flutter-uses-dart-dd635a054ebf)
Article	Dart for JavaScript Programmers (https://medium.com/flutter-community/dart-for-javascript-programmers-24c905aadf6)
YouTube	Fireship: Dart in 100 Seconds (https://www.youtube.com/watch?v=NrO0CJCbYLA)
YouTube	Mike Dane: Dart Programming in 4 hours Full beginners tutorial (https://www.youtube.com/watch?v=5xIVP04905w)

Flutter - Fundamentals:

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase.

Flutter widgets are built using a modern framework that takes inspiration from React. The central idea is that you build your UI out of widgets. Widgets describe what their view should look like given their current configuration and state.

Installing and configuring Flutter

Learning how to use basic Flutter widgets, such as Column, Row, Scaffold, Text, Image, Container, AppBar, ElevatedButton, among other

Understanding the differences between Stateless and Stateful widgets

Developing your first Flutter application

Contents

WebSite Installing Flutter (<https://docs.flutter.dev/get-started/install>)

WebSite	Introduction to widgets (https://docs.flutter.dev/development/ui/widgets-intro)
---------	---------------------------	---	---

WebSite Widget catalog (<https://docs.flutter.dev/development/ui/widgets>)

Article	What even are Flutter widgets? (https://medium.com/super-declarative/what-even-are-flutter-widgets-ce537a048a7d)
---------	----------------------------------	---	---

Article	Flutter widgets and layouts in minutes (https://medium.com/hackernoon/flutter-widgets-and-layouts-in-minutes-ffd4d2f0d443)
---------	--	---	---

Learn About Flutter Widgets by Making a Simple Weather App (<https://betterprogramming.pub/learn-about-flutter-widgets-by-making-a-simple-weather-app-46c8b9e9b24e>)

Flutter:
Widget
of the <https://www.youtube.com/watch?v=XawP1i314WM&list=PLjxrf2q8roU23XGwz3Km7sQZFTdB996iG>)
YouTube Week (Playlist)
(

Flutter Mapp: Flutter Tutorial For Beginners in 3 Hours (<https://www.youtube.com/watch?v=CD1Y2DmL5JM>)

freeCodeCamp.org: Flutter Course for Beginners – 37-hour Cross Platform App Development Tutorial (<https://www.youtube.com/watch?v=VPvVD8t02U8>)

Object-oriented Programming Concepts:

Object-oriented programming (OOP) is a programming paradigm based on the concept of 'objects', which can contain data and code: data in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods). A common feature of objects is that procedures (or methods) are attached to them and can access and modify the object's data fields. Some of the main concepts are classes and instances, inheritance, and encapsulation.

How objects work

Creating and using constructors

What classes are

Creating and using Methods

How encapsulation works

What inheritance is

What polymorphism is

How interfaces work

What abstractions are

Contents

WebSite	MDN Web Docs: Object-oriented programming (https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_programming)
WebSite	Geeks for Geeks: Object Oriented Programming (OOPs) Concept in Java (https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/)
WebSite	W3Schools: Object-Oriented Programming Style (https://www.w3schools.blog/object-oriented-programming-style)
Article	What is object-oriented programming? OOP explained in depth (https://www.educative.io/blog/object-oriented-programming)
YouTube	Programming with Mosh: Object-oriented Programming in 7 minutes (https://www.youtube.com/watch?v=QJjY2TNyl-8)
YouTube	freeCodeCamp.org: Intro to Object Oriented Programming - Crash Course (https://www.youtube.com/watch?v=SiBw7os-_zI)
YouTube	Traversy Media: JavaScript OOP Crash Course (ES5 & ES6) (https://www.youtube.com/watch?v=vDJpGenyHaA)

Flutter - Persistence:

Persistence in computer science refers to saving data for future access. It also involves the actions of reading, updating and deleting data.

Understanding what data persistence is

Learning how to store data using SharedPreferences

Learning how to use SQLite in Flutter

Learning how to persist data reading and writing files

Contents

WebSite	Persist data with SQLite (https://docs.flutter.dev/cookbook/persistence/sqlite)
WebSite	Store key-value data on disk (https://docs.flutter.dev/cookbook/persistence/key-value)

WebSite PRead and write <https://docs.flutter.dev/cookbook/persistence/reading-writing-files> ()

WebSite W3Schools: SQL Syntax (https://www.w3schools.com/sql/sql_syntax.asp)

Article Basic SQL Commands - The List of Database Queries and Statements You Should Know (<https://www.freecodecamp.org/news/basic-sql-commands/>)

Article Using SQLite in Flutter (<https://medium.com/flutter-community/using-sqlite-in-flutter-187c1a82e8b>)

YouTube Learn Flutter with Me: Sqflite Tutorial | Flutter plugin for SQLite (<https://www.youtube.com/watch?v=noi6aYsP7Go>)

YouTube Johannes Milke: Flutter Tutorial - SQL Database Storage Using Sqlite & Sqflite CRUD | Android & iOS (<https://www.youtube.com/watch?v=noi6aYsP7Go>)

Dart - Asynchronous:

In asynchronous programming, the functions are not executed in order. We can interrupt the code to get some other information needed to continue execution. This means that the code waits for another part of the code, and while it waits, it can execute the other parts.

Understanding what asynchronism is

Learning how to use the `async`, `await` and `Future` expressions in Dart

Understanding `Future`'s `then` and `catchError` methods in Dart

Contents

WebSite Asynchrony support | Dart Language Tour (<https://dart.dev/guides/language/language-tour#asynchrony-support>)

Article Asynchronous programming: futures, `async`, `await` (<https://dart.dev/codelabs/async-await>)

Article Asynchronous programming: Streams (<https://dart.dev/tutorials/language/streams>)

Article Concurrency in Dart (<https://dart.dev/guides/language/concurrency>)

Flutter Future Tutorial — Article	Asynchronous Dart Programming (https://thinkdiff.net/flutter-future-tutorial-asynchronous-dart-programming-a0c989c0f612)
Article	When to use async, await, then and Future in Dart? (https://jelenaaa.medium.com/when-to-use-async-await-then-and-future-in-dart-5e00e64ab9b1)
Article	Dart asynchronous programming: Futures (https://medium.com/dartlang/dart-asynchronous-programming-futures-96937f831137)
YouTube	Flutter: Advanced Dart (Playlist) (https://www.youtube.com/playlist?list=PLjxrf2q8roU0Net_g1NT5_vOO3s_FR02J)

Dart - Errors and Exceptions:

Error handling refers to the procedures for responding to and recovering from error conditions present in a software application. In other words, it is the process of anticipating, detecting, and resolving application, programming, or communication errors.

Understanding what errors and exceptions are in Dart

Learning how to use the `try-on-catch-finally` structure

Contents

WebSite	Dart – Types of Exceptions (https://www.geeksforgeeks.org/dart-types-of-exceptions/)
WebSite	Null safety (https://dart.dev/null-safety)		
WebSite	Understanding null safety (https://dart.dev/null-safety/understanding-null-safety)
Article	Exceptions, Errors and asserts in Dart (https://medium.com/flutter-community/exceptions-errors-and-asserts-in-dart-bf021794cb21)
Article	Handling Exceptions in Dart & Flutter: Everything you need to know (https://quickbirdstudios.com/blog/handling-exceptions-flutter-dart/)
YouTube	Johan Jurrius: Dart Errors and Exceptions - Exceptions (try-catch-)

finally blocks) and create own Exception <https://www.youtube.com/watch?v=N5HgO7dLgc8>
class (
Reso Coder: Proper Error Handling in <https://www.youtube.com/watch?v=NoZde-wYGA4>
YouTube Flutter & Dart ()

Dart - Communication with APIs:

An API is an interface that software developers use to programmatically interact with software components or resources outside of their own code. An even simpler definition is that an API is the part of a software component that is accessible to other components.

Understanding REST APIs

Learning the basic HTTP communication commands

Learning how to use the Darthttp package tools

Learning how to make authenticated requests to Web APIs

Contents

What is a REST API? <https://www.redhat.com/en/topics/api/what-is-a-rest-api>
WebSite ()

Making Your API Calls <https://medium.com/solidmvp-africa/making-your-api-calls-in-flutter-the-right-way-f0a03e35b4b1>
ArticleIn Flutter The Right Way ()

JSON and <https://docs.flutter.dev/development/data-and-backend/json>
WebSite serialization ()

WebSitePackage http - pub.dev (<https://pub.dev/packages/http>)

Fetch data from the <https://docs.flutter.dev/cookbook/networking/fetch-data>
WebSite internet ()

Make <https://docs.flutter.dev/cookbook/networking/authenticated-requests>
WebSite authenticated requests ()

Send data to the <https://docs.flutter.dev/cookbook/networking/send-data>
WebSite internet ()

WebSite	Delete data on the internet (https://docs.flutter.dev/cookbook/networking/delete-data)
YouTube	CodeX: REST API with Flutter Step by step tutorial (https://www.youtube.com/watch?v=c09XiwOZKsl)
YouTube	Coding Orbit: Flutter Consume Rest API With Best Practices (https://youtu.be/5S9mMBEtJK4)

Level 2

Flutter - Essential Packages:

Flutter supports using shared packages contributed by other developers to the Flutter and Dart ecosystems. This allows quickly building an app without having to develop everything from scratch.

Learning how to use the pub.dev platform

Learning about the main Flutter packages

Contents

WebSite	Using packages (https://docs.flutter.dev/development/packages-and-plugins/using-packages)
WebSite	Dart and Flutter package repository pub.dev (https://pub.dev/)
WebSite	Package scores & pub points (https://pub.dev/help/scoring)
WebSite	Flutter favorites (https://pub.dev/packages?q=is%3Aflutter-favorite)
WebSite	Popular packages (https://pub.dev/packages?sort=popularity)
Article	The 7 Flutter Packages I Can't Live Without (https://levelup.gitconnected.com/the-7-flutter-packages-i-cant-live-without-9c18ac8540bd)
Article	Flutter packages which I use in every project (https://medium.com/@dawletataoglu/flutter-packages-which-i-use-in-every-project-d8bf14c43622)
Article	5 Tools/Packages for Productive Flutter Development (https://medium.com/flutter-community/5-flutter-packages-tools-for-faster-more-productive-project-development-f8adbac05616)

YouTube	Robert Brunhage: The Ultimate Flutter Starter Packages (https://www.youtube.com/watch?v=zk-TkfPr3Z0)
YouTube	Flutter: Package of the Week (Playlist) (https://www.youtube.com/playlist?list=PLjxrf2q8roU1quF6ny8oFHJ2gBdrYN_AK)

Flutter - State Management:

In Flutter, 'state' is whatever data you need in order to rebuild your UI at any moment in time. When this data change, it will trigger a redraw of the user interface.

Understanding State Management

Learning how to use the Provider

Learning about other State Management options in Flutter

Contents

WebSite	State management (https://docs.flutter.dev/development/data-and-backend/state-mgmt/intro)
Article	A List of state management approaches in Flutter (https://medium.com/flutter-community/a-list-of-state-management-approaches-in-flutter-1481ba522a51)
Article	Google's Recommended Flutter State Management (https://fredgrott.medium.com/googles-recommended-flutter-state-management-60d2803dbe81)
Article	Flutter State Management with Provider (https://medium.com/flutter-community/flutter-statemangement-with-provider-ee251bbc5ac1)
WebSite	List of state management approaches (https://docs.flutter.dev/development/data-and-backend/state-mgmt/options)
YouTube	Fireship: Flutter State Management - The Grand Tour (https://www.youtube.com/watch?v=3tm-R7ymwhc)
YouTube	Codrut Erdei: State Management! Flutter What's next? (https://www.youtube.com/watch?v=Ct9cOiuR_dY)
YouTube	Flutter: Pragmatic State Management Using Provider (https://www.youtube.com/watch?v=HrBiNHEqSYU&ab_channel=Flutter)

Flutter - Testing:

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

Understanding unit tests in Flutter

Understanding widget tests in Flutter

Understanding integration tests in Flutter

Contents

WebSiteTesting Flutter apps (<https://docs.flutter.dev/testing>)

WebSiteAn introduction to unit testing (<https://docs.flutter.dev/cookbook/testing/unit/introduction>)

WebSiteAn introduction to widget testing (<https://docs.flutter.dev/cookbook/testing/widget/introduction>)

WebSiteIntegration testing (<https://docs.flutter.dev/testing/integration-tests>)

YouTubeReso Coder: Flutter Testing Guide for Beginners - Part 1 - Unit Tests & Setup (<https://www.youtube.com/watch?v=hUAUAkIZmX0>)

YouTubeFlutter Explained: Automate your Testing on Devices with Flutter Integration Testing (<https://www.youtube.com/watch?v=WPEsnJgW99M>)

YouTubeFlutter: Testing Flutter Apps - Making Sure Your Code Works (<https://youtu.be/bj-oMYyLZEY>)

YouTubeFlutter: Flutter performance tips (<https://youtu.be/PKGguGUwSYE>)

WebSiteHow to test a Flutter app (<https://codelabs.developers.google.com/codelabs/flutter-app-testing?hl=pt-br#0>)

Flutter - Debugging:

Debugging is the process of finding and resolving bugs (defects or problems that prevent correct operation) within computer programs, software, or systems.

Learning debugging techniques and tools in Flutter

Learning how to debug Dart and Flutter code

Learning about Dart DevTools

Contents

WebSiteDebugging Flutter apps (<https://docs.flutter.dev/testing/debugging>)

YouTube Bleyl Dev: Flutter Debugging (https://www.youtube.com/watch?v=Z0_bCYlgZlw)

WebSite Debugging Flutter apps (<https://docs.flutter.dev/testing/code-programmatically>)

WebSiteUsing an OEM debugger (<https://docs.flutter.dev/testing/oem-debuggers>)

YouTubeFlutter: Debug properties | Decoding Flutter (<https://youtu.be/DnC7eT-vh1k>)

YouTube Flutter: Debug and profile your app with Flutter DevTools | Demo (<https://youtu.be/J7s5bu49k3U>)

WebSiteDart DevTools (<https://dart.dev/tools/dart-devtools>)

YouTubeFlutter: Dart DevTools (<https://youtu.be/nq43mP7hjAE>)

Flutter - Games:

Leaning how to create games with Flutter

Getting to know Flutter's Casual Game Development Kit

Learning about the Flame package as a Flutter Game Engine

Learning about the Bonfire package as a Flutter Game Engine

Contents

WebSiteFlutter Casual Games Toolkit (<https://flutter.dev/games>)

WebSiteCasual Games Toolkit (<https://docs.flutter.dev/resources/games-toolkit>)

Article Flutter & games: exploring Bonfire (<https://asgalex.medium.com/flutter-games-exploring-bonfire-1190938f3020>)

Article How I Rebuilt the OG Snake Game using Flutter (<https://medium.com/@geekyants/how-i-rebuilt-the-og-snake-game-using-flutter-6cceb009afcf>)

YouTube Flutter: Quick start to building a game in Flutter (<https://youtu.be/zGgeBNiRy-8>)

YouTube	Flutter Mapp: My First Mobile Game With Flutter and Flame (https://youtu.be/3GGcVF9p8bY)
YouTube	Mitch Koko: Flutter Games from Scratch (https://www.youtube.com/playlist?list=PLlvRDpXh1Se6kipeBLiF1xByAEmxYie6J)
YouTube	Flame Engine: Flutter Vikings 2022 - The basics of the Flame game engine (https://youtu.be/NqjaLM0TWm0)
YouTube	Majid Hajian: Building Game with Flutter using Flame engine Pair programming with authors of Flame (https://youtu.be/NqjaLM0TWm0)
YouTube	Craig Oda: Flame Game Velocity, Gravity, Basic Sprite Movement (Flutter games) (https://youtu.be/hwQpBuZoV9s=)

Level 3

Flutter - Native:

Getting to know the specifics of native development for Android

Getting to know the specifics of native development for iOS

Contents

	Android: Hosting native	
WebSite	Android views in your Flutter app with Platform Views (https://docs.flutter.dev/development/platform-integration/android/platform-views)
WebSite	Android: Supporting the new Android plugins APIs (https://docs.flutter.dev/development/platform-integration/android/plugin-api-migration)
WebSite	Android: AndroidX migration (https://docs.flutter.dev/development/platform-integration/android/androidx-migration)
WebSite	Android: Deprecated Splash Screen API Migration (https://docs.flutter.dev/development/platform-integration/android/splash-screen-migration)
WebSite	iOS: Cupertino (iOS-style) widgets (https://docs.flutter.dev/development/ui/widgets/cupertino)

	iOS: Veggie	
WebSite	Seasons - iOS	https://github.com/flutter/samples/tree/main/veggieseasons)
	with Flutter (
YouTube	iOS: Flutter: Building for	https://www.youtube.com/watch?v=ZBJa-xjZl3w&ab_channel=Flutter)
	iOS with Flutter (
WebSite	iOS: Adding an iOS App	https://docs.flutter.dev/development/platform-integration/ios/ios-app-clip)
	Clip target (
	iOS: Hosting native iOS	https://docs.flutter.dev/development/platform-integration/ios/platform-views)
WebSite	views in your Flutter app	
	with Platform Views (
WebSite	Writing custom platform-	https://docs.flutter.dev/development/platform-integration/platform-channels)
	specific code (
WebSite	C interop using dart:ffi	(https://dart.dev/guides/libraries/c-interop)
WebSite	JavaScript interoperability	(https://dart.dev/web/js-interop)
WebSite	Binding to native Android	https://docs.flutter.dev/development/platform-integration/android/c-interop)
	code using dart:ffi (
WebSite	Binding to native iOS code	https://docs.flutter.dev/development/platform-integration/ios/c-interop)
	using dart:ffi (

Data Structures:

In the context of computers, the data structure is a specific way of storing and organizing data in the computer's memory so that these data can be easily retrieved and efficiently used when needed later.

Knowing the main data structures (linked list, stack, queue, tree, etc)

Implementing the main data structures

Contents

WebSite	W3Schools: Data Structure	https://www.w3schools.in/data-structures/tutorials/)
	Tutorial (
WebSite	Geeks for Geeks: Data	https://www.geeksforgeeks.org/data-structures/)
	Structures (

WebSite	Data Structures Tutorial (https://www.javatpoint.com/data-structure-tutorial)
Article	8 Common Data Structures every Programmer must know (https://towardsdatascience.com/8-common-data-structures-every-programmer-must-know-171acf6a1a42)
YouTube	Internet Made Coder: Data Structures Explained for Beginners - How I Wish I was Taught (https://www.youtube.com/watch?v=ZdU4wMyiTSs)
YouTube	freeCodeCamp.org: Data Structures Easy to Advanced Course - Full Tutorial from a Google Engineer (https://www.youtube.com/watch?v=ZdU4wMyiTSs)
YouTube	Aaron Jack: Data Structures you MUST know (as a Software Developer) (https://www.youtube.com/watch?v=ZdU4wMyiTSs)

Flutter - Animations:

Learning the difference between Implicit and Explicit Animations in Flutter

Learning about and using Widgets for Implicit Animations

Understanding the concepts of Duration and Curves

Learning about the technique of Explicit Animations in Flutter

Contents

WebSite	Introduction to animations (https://docs.flutter.dev/development/ui/animations)
WebSite	Implicit animations (https://docs.flutter.dev/codelabs/implicit-animations)
YouTube	Flutter: Animation Basics with Implicit Animations (https://youtu.be/IVTjpW3W33s)
WebSite	Animation and motion widgets (https://docs.flutter.dev/development/ui/widgets/animation)
YouTube	Johannes Milke: AnimatedContainer - Deep Dive (https://www.youtube.com/watch?v=Mj-NGe7svAg)
Article	Flutter Animations Cheat Sheet (https://medium.com/flutter-community/flutter-animations-cheat-sheet-7f8cebfb850c)

Take Your Flutter Animations <https://medium.com/@roaakdm/take-your-articles-to-the-next-level-flutter-vikings-talk> ()

Article to The Next Level — Flutter [flutter-animations-to-the-next-level-flutter-vikings-talk-c7130939a4c1](https://medium.com/@roaakdm/take-your-articles-to-the-next-level-flutter-vikings-talk-c7130939a4c1)

Vikings Talk ()

WebSite Fade a widget in and out (<https://docs.flutter.dev/cookbook/animation/opacity-animation>)

WebSite Animate the properties of a container (<https://docs.flutter.dev/cookbook/animation/animated-container>)

WebSite Animate a page route transition (<https://docs.flutter.dev/cookbook/animation/page-route-animation>)

WebSite Animate a widget using a physics simulation (<https://docs.flutter.dev/cookbook/animation/physics-simulation>)

WebSite Explicit animations (<https://docs.flutter.dev/codelabs/explicit-animations>)

WebSite Hero animations (<https://docs.flutter.dev/development/ui/animations/hero-animations>)

WebSite Staggered animations (<https://docs.flutter.dev/development/ui/animations/staggered-animations>)

YouTube Flutter: Animations in Flutter done right (<https://youtu.be/wnARLByOtKA>)

Flutter - Multiplatform - Desktop:

Learning about and using Flutter's multiplatform creation options

Contents

WebSite Flutter on Desktop (<https://flutter.dev/multi-platform/desktop>)

WebSite Desktop support for Flutter (<https://docs.flutter.dev/development/platform-integration/desktop>)

WebSite Building Linux apps with Flutter (<https://docs.flutter.dev/development/platform-integration/linux/building>)

WebSite Building Windows apps with Flutter (<https://docs.flutter.dev/development/platform-integration/windows/building>)

Building macOS apps with Flutter (<https://docs.flutter.dev/development/platform-integration/macos/building>)

Flutter - Multiplatform - Web:

Getting to know the specifics of creating Flutter apps for the Web

Contents

Flutter Web FAQ (<https://docs.flutter.dev/development/platform-integration/web/faq>)

Web renderers (<https://docs.flutter.dev/development/platform-integration/web/renderers>)

Customizing web app initialization (<https://docs.flutter.dev/development/platform-integration/web/initialization>)

Displaying images on the web (<https://docs.flutter.dev/development/platform-integration/web/web-images>)

Creating a Progressive Web App (PWA) with Flutter (<https://blog.codemagic.io/pwa-in-flutter/>)

Flutter - Deployment:

Software deployment includes all of the steps, processes, and activities that are required to make a software system or update available to its intended users.

Learning about Dart code obfuscation techniques

Learning know how to deploy Flutter applications on various platforms

Understanding Flavors in Flutter

Learning about Continuous Delivery in Flutter

Contents

Obfuscating Dart code (<https://docs.flutter.dev/deployment/obfuscate>)

Build and release an Android app (<https://docs.flutter.dev/deployment/android>)

Build and release an iOS app (<https://docs.flutter.dev/deployment/ios>)

WebSite	Build and release a macOS app (https://docs.flutter.dev/deployment/macos)
WebSite	Build and release a Linux app to the Snap Store (https://docs.flutter.dev/deployment/linux)
WebSite	Build and release a Windows desktop app (https://docs.flutter.dev/deployment/windows)
WebSite	Build and release a web app (https://docs.flutter.dev/deployment/web)
WebSite	Creating flavors for Flutter (https://docs.flutter.dev/deployment/flavors)
WebSite	Continuous delivery with Flutter (https://docs.flutter.dev/deployment/cd)
YouTube	MJSD Coding: Build and Release Flutter App to App Store (https://youtu.be/akFF1uJWZck)
Article	Publishing Flutter App To PlayStore (https://medium.flutterdevs.com/publishing-flutter-app-to-playstore-fa7543b61a7b)
YouTube	Johannes Milke: How To Publish Flutter App On Play Store (https://youtu.be/PM-wILFbPbg)

Flutter - Architecture:

In programming, there are several architectural patterns for project files and folders, that provide developers with ease in finding, researching, and understanding the written code.

Getting to know the main architecture patterns used by the Flutter community, such as MVC, MVVM, MVP etc

Contents

Article	An Introduction to Flutter Clean Architecture (https://medium.com/ruangguru/an-introduction-to-flutter-clean-architecture-ae00154001b0)
Article	Flutter: MVVM Architecture (https://medium.com/flutterworld/flutter-mvvm-architecture-f8bed2521958)
Article	Design Patterns in Flutter- Part 1(MVC) (https://medium.flutterdevs.com/design-patterns-in-flutter-part-1-c32a3ddb00e2)

Article	Flutter Project Structure: Feature-first or Layer-first? (https://codewithandrea.com/articles/flutter-project-structure/)
YouTube	CodeCave: MVC Pattern in Flutter (https://www.youtube.com/watch?v=RQf5VVuGJh8)
YouTube	Mobile Programmer: MVVM in Flutter using Providers (https://www.youtube.com/watch?v=W1e7d3cvmRo)

Flutter - Packages and Plugins:

Learning how to develop plugins for Flutter

Contents

YouTubeFlutter: Packages versus Plugins? (<https://youtu.be/Y9WifT8aN6o>)

YouTubeLara Martín: Flutter Plugin Development (<https://youtu.be/PLPhmsDovwQ>)

WebSite Developing packages & plugins (<https://docs.flutter.dev/development/packages-and-plugins/developing-packages>)

WebSite Federated plugins (<https://docs.flutter.dev/development/packages-and-plugins/developing-packages#federated-plugins>)

Article Writing a good Flutter plugin (<https://medium.com/flutter/writing-a-good-flutter-plugin-1a561b986c9c>)

WebSite Contributing to Flutter (<https://github.com/flutter/flutter/blob/master/CONTRIBUTING.md>)

Auxiliary Skill: Infrastructure and Back-end

Git & GitHub - Fundamentals:

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

GitHub is a hosting service for software development and version control using Git.

Creating a repository

Cloning a repository

Committing, pushing and pulling to and from the repository

Reversing a commit

Creating branches and pull requests

Handling merge and conflicts

Contents

WebSiteGit Reference Book (<https://git-scm.com/book/en/v2>)

WebSiteGitHub Documentation (<https://docs.github.com/en>)

WebSiteGitHub Pages Documentation (<https://docs.github.com/en/pages/getting-started-with-github-pages/about-github-pages>)

WebSiteW3Schools: Git Tutorial (<https://www.w3schools.com/git/default.asp?remote=github>)

WebSiteGit School - Visualizing Git (<https://git-school.github.io/visualizing-git/>)

WebSiteDangit, Git!?! (<https://dangitgit.com/>)

ArticleGit Tutorial - Explore The Commands And Operations In Git (<https://medium.com/edureka/git-tutorial-da652b566ece>)

ArticleGit and Github Quickstart Tutorial (<https://medium.com/@prashantramnyc/git-and-github-quickstart-tutorial-654a71594dca>)

ArticleGetting Started with Git and GitHub: A Complete Tutorial for Beginner (<https://towardsdatascience.com/learn-basic-git-commands-for-your-data-science-works-2a75396d530d>)

YouTubeProgramming with Mosh: Git Tutorial for Beginners - Learn Git in 1 Hour (<https://www.youtube.com/watch?v=8JJ101D3knE>)

YouTubefreeCodeCamp.org: Git and GitHub for Beginners - Crash Course (<https://www.youtube.com/watch?v=RGQj5yH7evk>)

YouTubeKevin Stratvert: Git and GitHub for Beginners Tutorial (<https://www.youtube.com/watch?v=tRZGeaHPoaw>)

YouTubeTech With Tim: Git Tutorial for Beginners - Git & GitHub Fundamentals In Depth (<https://www.youtube.com/watch?v=DVRQoVRzMIY>)

HTTP - Fundamentals:

HTTP stands for Hyper Text Transfer Protocol. Communication between client computers and web servers is done by sending HTTP Requests and receiving

HTTP Responses.

Understanding the difference between HTTP verbs

Testing requests and checking the status codes in the browser

Learning how to make a HTTP request on the command line with WGET

Downloading an image with WGET

Performing a POST

Contents

WebSite	W3Schools: What is HTTP? (https://www.w3schools.com/whatis/whatis_http.asp)
WebSite	MDN Web Docs: An overview of HTTP (https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview)
WebSite	MDN Web Docs: HTTP request methods (https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods)
WebSite	HTTP Cats (https://http.cat/)
WebSite	HTTP Dogs (https://http.dog/)
Article	HTTP codes as Valentine's Day comics (https://medium.com/@hanilim/http-codes-as-valentines-day-comics-8c03c805faa0)
Article	Here Are the most popular ways to make an HTTP request in JavaScript (https://medium.com/free-code-camp/here-is-the-most-popular-ways-to-make-an-http-request-in-javascript-954ce8c95aaa)
YouTube	Traversy Media: HTTP Crash Course & Exploration (https://www.youtube.com/watch?v=iYM2zFP3Zn0)
YouTube	Curious Code: HTTP Request Methods - GET, POST, PUT, DELETE (https://www.youtube.com/watch?v=tkfVQK6UxDI)
YouTube	freeCodeCamp.org: Postman Beginner's Course - API Testing (https://www.youtube.com/watch?v=VywxIQ2ZXw4)

JSON:

JSON stands for JavaScript Object Notation. It is a text format for storing and transporting data.

Creating an object

Transforming an object into a string

Transforming a string into an object

Manipulating an object

Contents

WebSiteW3Schools: JSON (https://www.w3schools.com/js/js_json_intro.asp)

WebSite MDN Web Docs: JSON (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON)

WebSite MDN Web Docs: Working with JSON (<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>)

YouTube Web Dev Simplified: Learn JSON in 10 Minutes (<https://www.youtube.com/watch?v=iiADhChRriM>)

YouTube Traversy Media: JSON Crash Course (<https://www.youtube.com/watch?v=wI1CWzNtE-M>)

Design Patterns:

In software engineering, a Design Pattern is a general, reusable solution to a commonly occurring problem within a given context in software design. It is a description or template for how to solve a problem that can be used in many different situations. Design Patterns are formalized best practices that the programmer can use to solve common problems when designing an application or system.

Getting familiarized with and applying the main Design Patterns

Contents

WebSite What's a design pattern? (<https://refactoring.guru/design-patterns/what-is-pattern>)

Article Design Patterns — Introduction (<https://henriquesd.medium.com/design-patterns-introduction-220f811db857>)

Article What Are Design Patterns and Do I Need Them? (<https://www.developer.com/design/what-are-design-patterns-and-do-i-need-them/>)

The 3 Types of Design Patterns Article	https://www.freecodecamp.org/news/the-basic-design-patterns-all-developers-should-know/	(with code examples of each) (need-to-know/)
10 Design Patterns every Software Architect and Software Engineer must know Article	https://ravindraelicherla.medium.com/10-design-patterns-every-software-architect-must-know-b33237bc01c2)
Modern-Day Architecture Article	https://betterprogramming.pub/modern-day-architecture-design-patterns-for-software-professionals-9056ee1ed977	Design Patterns for Software Professionals ()
Web Dev Simplified: Design Patterns (Playlist) YouTube	https://www.youtube.com/playlist?list=PLZIA0Gpn_vH_CthENcPCM0Dww6a5XYC7f	()
Programming with Mosh: Design Patterns in Plain English YouTube	https://www.youtube.com/watch?v=NU_1StN5Tkk)
Fireship: 10 Design Patterns Explained in 10 Minutes YouTube	https://www.youtube.com/watch?v=tv-1er1mWl)
Traversy Media: 5 Design Patterns Every Engineer Should Know YouTube	https://www.YOUTUBE.com/watch?v=AWOf6Wo6gtg)

Command Line - Fundamentals:

CLI is a command line program that accepts text input to execute operating system functions.

Knowing the most important commands

Contents

W3Schools: What is Command Line Interface (CLI)? WebSite	https://www.w3schools.com/whatis/whatis_cli.asp	
Microsoft Docs: Using command line arguments for Windows Terminal WebSite	https://docs.microsoft.com/en-us/windows/terminal/command-line-arguments?tabs=windows	
Advanced CLI: Commands You Should Know as a Article	https://betterprogramming.pub/advanced-cli-commands-you-should-know-as-a-developer-)

Developer (7bc48c752a5e)
freeCodeCamp.org: Command Line	https://www.youtube.com/watch?v=yz7nYlnXLfE)
Crash Course ()
Traversy Media: Command Line Crash	https://www.youtube.com/watch?v=uwAqEzhyjtw)
Course For Beginners - Terminal)
Commands ()

Cloud - Fundamentals:

Cloud, or cloud computing, is the distribution of computing services over the Internet using a pay-as-you-go pricing model. A cloud is composed of various computing resources, ranging from the computers themselves (or instances, in cloud terminology) to networks, storage, databases, and everything around them. In other words, everything that is normally needed to set up the equivalent of a server room, or even a complete data center, will be ready to use, configured, and run.

Knowing the difference between IaaS, PaaS and SaaS

Knowing the largest cloud providers

Specializing in a specific provider of your choice

Contents

WebSite	Microsoft Azure: What is cloud computing? (https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing/)
WebSite	Amazon AWS: What is cloud computing? (https://aws.amazon.com/en/what-is-cloud-computing/)
Article	A beginner's guide to the basics of what cloud computing is about (https://scientya.com/a-beginners-guide-to-the-basics-of-what-cloud-computing-is-about-e8b3b7f25a30/)
Article	Cloud Computing for Beginners (https://medium.com/hackernoon/cloud-computing-for-beginners-85d168959afb/)
Article	What are Cloud Computing Services [IaaS, CaaS, PaaS, FaaS, SaaS] (https://medium.com/@nnilesh7756/what-are-cloud-computing-services-iaas-caas-paas-faas-saas-ac0f6022d36e)

YouTube	Simplilearn: Cloud Computing Tutorial for Beginners (https://www.youtube.com/watch?v=RWgW-CgdIk0)
YouTube	Amazon Web Services: What is Cloud Computing? - Amazon Web Services (https://www.youtube.com/watch?v=mxT233EdY5c)
YouTube	Ecourse Review: Cloud Computing Services Models - IaaS PaaS SaaS Explained (https://www.youtube.com/watch?v=36zducUX16w/)

SOLID:

SOLID has five principles that are considered best practices in software development that help programmers write cleaner code by separating responsibilities, reducing coupling, easing refactoring, and encouraging code reuse.

Contents

Article	The S.O.L.I.D Principles in Pictures (https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898)
Article	SOLID Principles every Developer Should Know (https://blog.bitsrc.io/solid-principles-every-developer-should-know-b3bfa96bb688)
YouTube	Fireship: Solid in 100 Seconds (https://www.youtube.com/watch?v=q0BGgQJcp7w)
YouTube	A Dev' Story: SOLID Design Principles Explained in a Nutshell (https://www.youtube.com/watch?v=69sfWNzxTMc)
YouTube	Web Dev Simplified: Single Responsibility Principle Explained - SOLID Design Principles (https://www.youtube.com/watch?v=UQqY3_6Epbq)

Clean Architecture:

Clean architecture is a way of developing software, such that just by looking at the source code of a program, you should be able to tell what the program does.

Contents

Article	The Clean Architecture —Beginner's Guide (https://betterprogramming.pub/the-clean-architecture-beginners-guide-e4b7058c1165)
Article	Thoughts on Clean Architecture (https://medium.com/android-news/thoughts-on-clean-architecture-b8449d9d02df)
Article	Clean Architecture, the right way (https://medium.com/gdg-vit/clean-architecture-the-right-way-d83b81ecac6)
Article	Why use a Clean Architecture (https://www.mytaskpanel.com/the-5-advantages-of-using-a-clean-architecture-all-you-need-to-know/)
Article	Clean Architecture – Build Software like an Artisan (https://flexiple.com/developers/clean-architecture-build-software-like-an-artisan/)
Article	React Clean Architecture (https://kpiteng.medium.com/react-clean-architecture-e4144a0788b6)
Article	Clean Architecture with Java 11 (https://medium.com/slalom-build/clean-architecture-with-java-11-f78bba431041)
Article	A detailed guide on developing Android apps using the Clean Architecture pattern (https://medium.com/@dmilicic/a-detailed-guide-on-developing-android-apps-using-the-clean-architecture-pattern-d38d71e94029)
YouTube	CodeOpinion: Clean Architecture Example & Breakdown (https://www.youtube.com/watch?v=Ys_W6MyWOCw)
YouTube	Women Who Code: Clean Architecture Concepts (https://www.youtube.com/watch?v=ouBSPdzbzvw)
YouTube	CodandoTV(Rods) - Simplifying Clean Architecture +MVVM in your mobile application - COMPLETE GUIDE (https://youtu.be/8ehIZfyN1S0?si=I7-I5l4zsnYLIjD)

Firestore:

Firestore is a Backend-as-a-Service (BaaS) app development platform that provides hosted backend services such as a realtime database, cloud storage, authentication, crash reporting, machine learning, remote configuration, and hosting for your static files.

Understanding how to install Firestore

Getting acquainted with Firebase documentation

Learning about the Firebase tools available

Contents

WebSite	Flutter:	https://docs.flutter.dev/development/data-and-backend/firebase)
Article	Firebase — Flutter (https://medium.com/google-developer-experts/firebase-70fd12309fd1)
Article	Firebase Android Series: Learning Firebase from zero to hero (https://proandroiddev.com/firebase-android-series-learning-firebase-from-zero-to-hero-3bacbdf8e048)
WebSite	Firebase - Case studies (https://firebase.google.com/use-cases)
WebSite	Add Firebase to your Flutter app (https://firebase.google.com/docs/flutter/setup?platform=android)
WebSite	Add Firebase to your Android project (https://firebase.google.com/docs/android/setup)
YouTube	Firebase: Getting started with Firebase on Flutter - Firecasts (https://www.youtube.com/watch?v=EXp0gg9kGxl&ab_channel=Firebase)
YouTube	Firebase: Getting started with Firebase on Android - Firecasts (https://www.youtube.com/watch?v=dRYnm_k3w1w)
YouTube	Johannes Milke: Flutter Firebase CRUD (https://www.youtube.com/watch?v=ErP_xomHKTW)
YouTube	GeeksforGeeks: User Authentication and CRUD Operation with Firebase Realtime Database in Android (https://www.youtube.com/watch?v=-Gvpf8tXpbc)
WebSite	Firebase - Pricing plans (https://firebase.google.com/pricing)

Auxiliary Skill: UX and Design

Design Systems:

A design system is a collection of reusable components, guided by clear standards, that can be assembled together to build applications.

Creating and maintaining libraries that will be consumed and used as a standard for building a project

Contents

Article	Everything you need to know about Design Systems (https://uxdesign.cc/everything-you-need-to-know-about-design-systems-54b109851969)
Article	A comprehensive guide to design systems (https://www.invisionapp.com/inside-design/guide-to-design-systems/)
Article	What is a UX Design System? (https://fuzzymath.com/blog/what-is-ux-design-system/)
Articles	Creating a design system in Figma — Tutorial 1 (https://medium.com/@johan_ronsse/creating-a-design-system-in-figma-the-not-so-definite-guide-tutorial-1-8aa6801101cb)
YouTube	DesignerUp: 5 Best Design Systems and How to Learn (and Steal) From Them (https://www.youtube.com/watch?v=BISC15OPeGA)
YouTube	DesignCourse: What is a Design System? Design Systems 101 for Designers (https://www.youtube.com/watch?v=wc5krC28ynQ)
YouTube	vaexperience: Design Systems - What are They and How to Get Started (https://www.youtube.com/watch?v=I6YuGE6EjJA)

Figma - Fundamentals:

Figma is a collaborative web application for interface design. The feature set of Figma focuses on user interface and user experience design, with an emphasis on real-time collaboration, utilising a variety of vector graphics editor and prototyping tools.

Creating page layouts and components

Contents

Article	Getting Started with Figma (https://levelup.gitconnected.com/getting-started-with-figma-637f2c868017)
Article	Introducing Figma to React (https://medium.com/figma-design/introducing-figma-to-react-d2d545cba3cc)

YouTube	Adrian Twarog: Figma Crash Course (https://www.youtube.com/watch?v=lg7w3Ntfqy0)
YouTube	AJ&Smart: Figma UI Design Tutorial - Get Started in Just 24 Minutes (https://www.youtube.com/watch?v=FTFaQWZBqQ8)
YouTube	Jesse Showalter: Intro to Figma - Beginners guide to Figma Basics (https://www.youtube.com/watch?v=jk1T0CdLxwU)
YouTube	Figma: Figma Tutorial - Components - The Basics (https://www.youtube.com/watch?v=k74lrUNaJVk)
YouTube	Figma: Figma Components 101 (https://www.youtube.com/watch?v=jk1T0CdLxwU)
Course	LearnUX: Figma Course (https://learnux.io/course/figma)	

Design components:

Knowing the components that describe a layout or interface

Contents

Article	Text fields & Forms design — UI components series (https://uxdesign.cc/text-fields-forms-design-ui-components-series-2b32b2beebd0)
Article	Design Components for Emails (https://blog.prototypr.io/design-components-for-emails-587f6267951d)
YouTube	GCFLearnFree.org: Beginning Graphic Design - Layout & Composition (https://www.youtube.com/watch?v=a5KYIHnkQB8)
YouTube	uxable - learning ux together: UI Elements/Components - Types and Importance of UI Elements (https://www.youtube.com/watch?v=roF6ocfArK0)
YouTube	Dennis Cortes: How to Design Powerful Components and Buttons with Variants and Auto Layout in Figma (https://www.youtube.com/watch?v=ODOR5EniJNg)
YouTube	Tutorial Tim: Let's Build a Design System - Understanding Layout (https://www.youtube.com/watch?v=nn3tO7gAFR4)

Color systems:

Defining a color palette that makes sense for a given interface

Contents

Article	Color Theory: There is more than one set of universal Primary Colors (https://medium.com/upskilling/color-theory-there-is-more-than-one-set-of-universal-primary-colors-debunking-the-myths-28140a7866c9)
Article	The Science of Color (https://medium.com/100-days-of-product-design/the-science-of-color-bd0f057c08ea)
Article	RGB vs. CMYK: A guide to color systems for designers (https://medium.com/envato/rgb-vs-cmyk-a-guide-to-color-systems-for-designers-6be8c1ed8554)
Article	Color in UI Design: A (Practical) Framework (https://medium.com/@erikdkennedy/color-in-ui-design-a-practical-framework-e18cacd97f9e)
Article	Basic UI color guide (https://blog.prototypr.io/basic-ui-color-guide-7612075cc71a)
YouTube	Rachel How: How to pick the right colors for your website or app (https://www.youtube.com/watch?v=ewRYw4pnKQU)
YouTube	Flux Academy: How to Choose Colors (Easy 3-Step Process) (https://www.youtube.com/watch?v=KMS3VwGh3HY)
YouTube	Jesse Showalter: 60-30-10 Color Rule (https://www.youtube.com/watch?v=UWwNIMHFdW4)
YouTube	DesignerUp: Super Practical Guide to Color Theory, Color Models and Perfect Color Palettes (https://www.youtube.com/watch?v=GyVMoejbGFg)

How to use Fonts:

Choosing the most appropriate font for a given project

Contents

Article	Top 10 UI Fonts for Web & Mobile (https://blog.prototypr.io/top-10-ui-fonts-for-web-mobile-a8488e561ce3)
Article	Best Fonts for UI Design, I use Daily - Best Typefaces and Font Resources for UI Designers (https://uxplanet.org/best-fonts-for-ui-design-i-use-daily-4a7bcffb966c)

DesignerUp: Best Practices for Choosing
YouTubeFonts and Font Pairing in UI and Web Design (<https://www.youtube.com/watch?v=30IoKUGaPWQ>)

DesignerUp: The Difference Between
YouTubeFonts, Typefaces and Typography for UI Designers (<https://www.youtube.com/watch?v=OKGTkLqgm58>)

DesignerUp: Choosing and Pairing
YouTubeFonts - UI Design (<https://www.youtube.com/watch?v=uWCWTq1cPW0>)

Mizko: The ONLY 8 Fonts UI Designers
YouTubeNeed. Forget The Rest. (<https://www.youtube.com/watch?v=mEAmAFgzQd4>)

Flutter - Accessibility:

Web accessibility is the inclusive practice of ensuring there are no barriers that prevent interaction with, or access to, websites by people with physical disabilities, situational disabilities, and socio-economic restrictions on bandwidth and speed.

Writing code with accessibility in mind

Contents

WebSiteW3: Accessibility (<https://www.w3.org/standards/webdesign/accessibility>)

Flutter - Documentation
WebSite- Accessibility (<https://docs.flutter.dev/development/accessibility-and-localization/accessibility>)

A deep dive into
ArticleFlutter's accessibility widgets (<https://medium.com/flutter-community/a-deep-dive-into-flutters-accessibility-widgets-eb0ef9455bc>)

The Single Most Useful
ArticleFlutter Accessibility Trick (<https://levelup.gitconnected.com/the-single-most-useful-flutter-accessibility-trick-ecf547cd71ad>)

Accessibility in
ArticleFlutter? (<https://tonyowen.medium.com/accessibility-in-flutter-592f2e760149>)

Flutter: Building in Accessibility
YouTubewith Flutter (<https://www.youtube.com/watch?v=bWbBgbmAdQs>)

monterail: Accessibility features and
YouTube how to implement them in Flutter (<https://www.youtube.com/watch?v=sO97YsPxwsw>)

Responsive Design:

Responsive web design (RWD) or responsive design is an approach to web design that aims to make web pages render well on a variety of devices and window or screen sizes from minimum to maximum display size to ensure usability and satisfaction.

Adjusting your pages to the user's screen size

Learning about Media queries

Knowing the concept of Mobile first

Contents

W3Schools: HTML
WebSite Responsive Web https://www.w3schools.com/html/html_responsive.asp
Design (

MDN Web Docs: https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design
WebSite Responsive design (Responsive Web

Design: What It Is <https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/>
Article And How To Use It (

The Beginner's Guide to Responsive
Article Web Design (Code Samples & Layout <https://kinsta.com/blog/responsive-web-design/>
Examples) (

Jesse Showalter: Responsive Web <https://www.youtube.com/watch?v=zF6Vsky4Slc>
YouTube Design | 10 Basics (

freeCodeCamp.org: Introduction To <https://www.youtube.com/watch?v=srvUrASNj0s>
YouTube Responsive Web Design - HTML & CSS Tutorial (

Web Dev Simplified: Learn CSS Media <https://www.youtube.com/watch?v=yU7jJ3NbPdA>
YouTube Query In 7 Minutes (

TechGuide - Alura
Alura, PM3 e FIAP
O Techguide.sh é um projeto open source