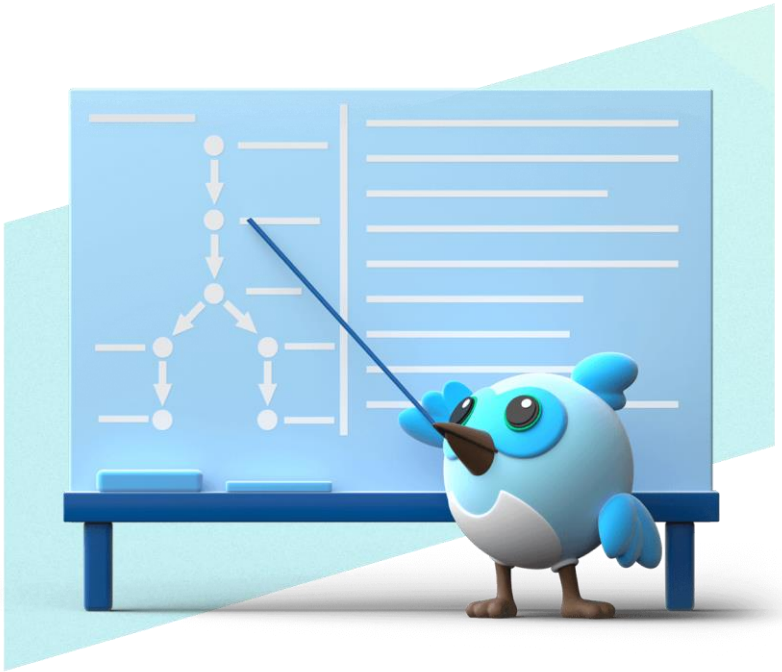
Mobile Application Development

Roshan David Jathanna
roshan.jathanna@manipal.edu

Flutter

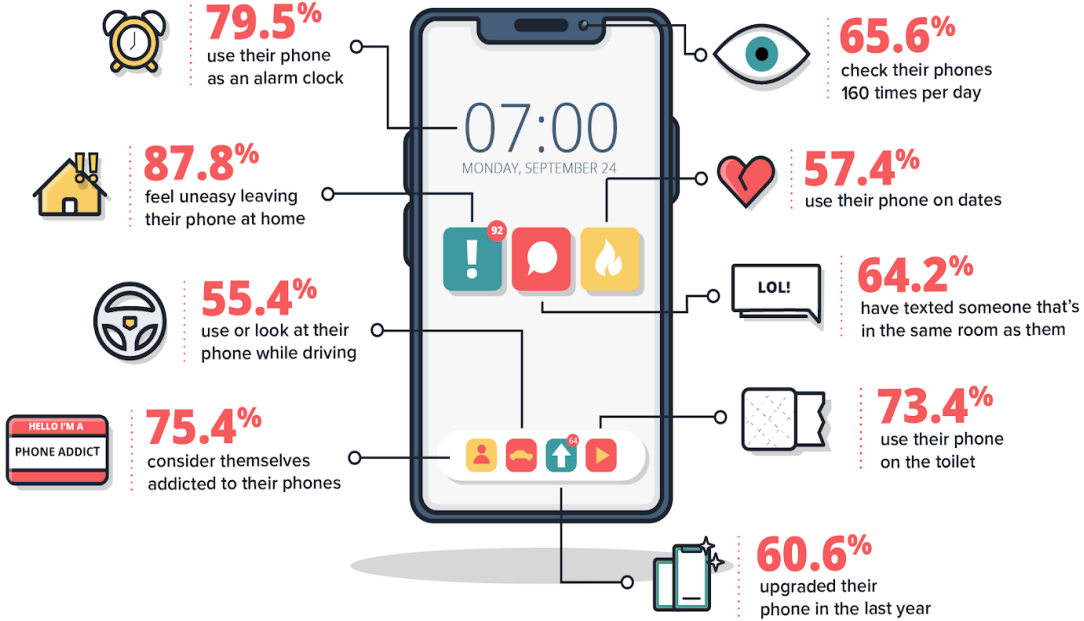


References

- Miola, A., 2020. Flutter Complete Reference: Create beautiful, fast and native apps for any device.
- https://flutter.dev/learn

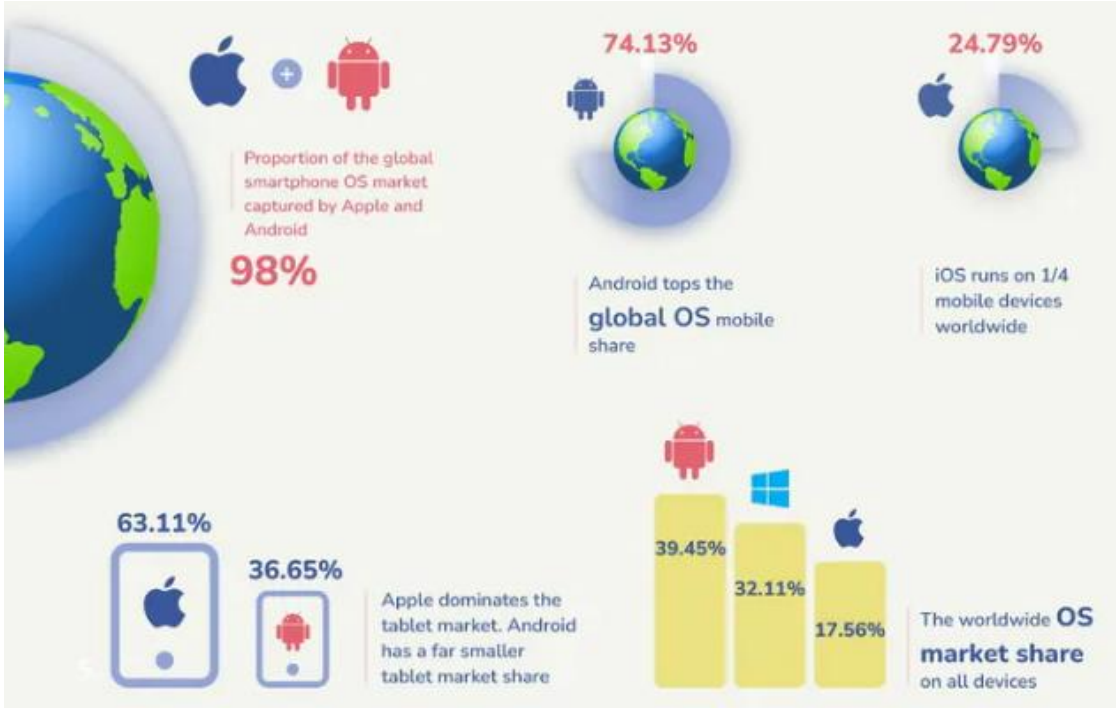
Can you guess?

American Survey 2022



Ref: https://www.digitalinformationworld.com/2020/03/smartphone-addiction-statistics.html

Mobile Platforms & OSs



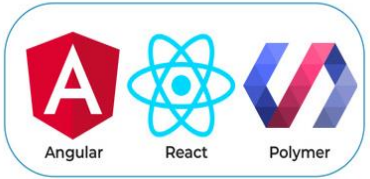
Ref: https://root-nation.com/en/news-en/en-android-ios-market-share/

App Development Approaches

1. Native

2. Cross Platform

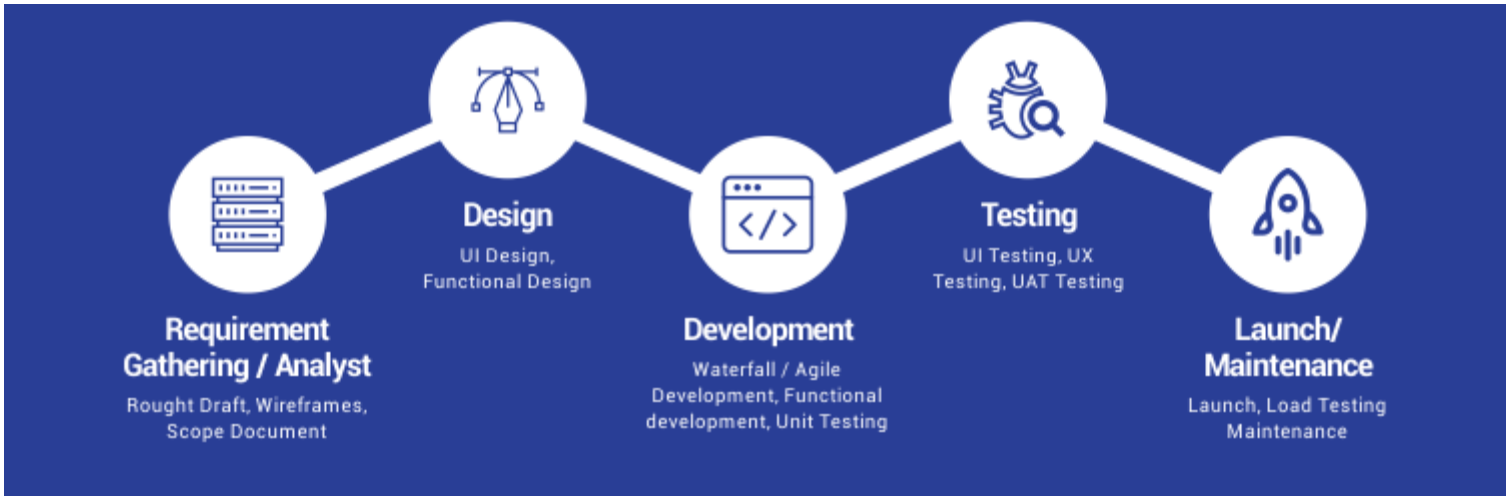
3. Hybrid



4. Progressive Web

Ref: https://railsware.com/blog/native-vs-hybrid-vs-cross-platform/amp/

App Development Lifecycle



Ref: https://www.itrobes.com/mobile-app-development-lifecycle

Installation – VS Code

Windows

macOS

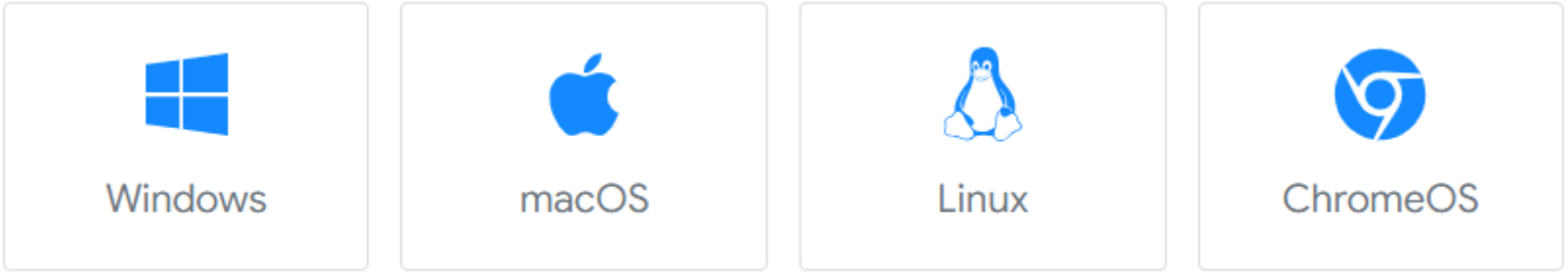
Linux

https://docs.flutter.dev/tools/vs-code

Installation – VS Code Extensions

- Flutter
- Dart
- Awesome Flutter Snippets
- Pubspec Assist
- Error Lens
- Dart Data Class Generator
- Material Icon Theme - Set File Icon Theme

Installation



https://docs.flutter.dev/get-started/install

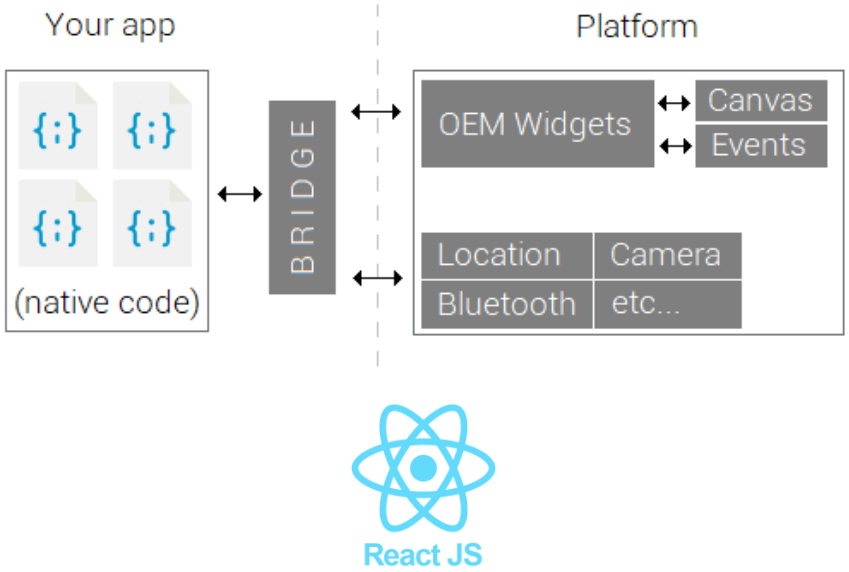
Why Flutter

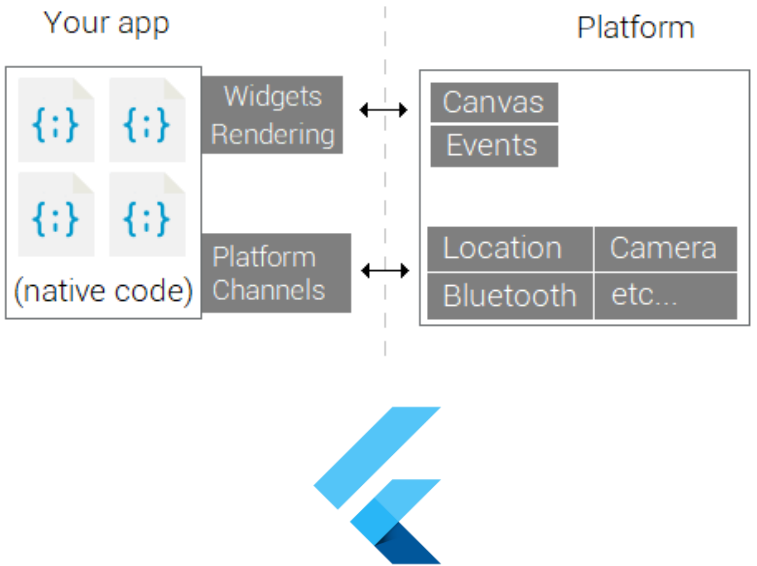
Flexible

Productive

Fast

How does it work





Why Dart

- Object-Oriented Programming (OOP) Style
- Performance
- Productivity
- Both Flutter and Dart are developed by Google

Dart

- int, double, bool, String
- final, late, const
- parse, tryParse, toString
- enum
- List<>
- Nullable type
- If else, switch, for, for-in, while
- function, anonymous

- List.ForEach
- Named, Positional parameter
- Import
- Classes
- Encapsulation
- Named constructor
- Cloning
- Extends, implements, mixins
- Exceptions
- Map, Spread
- Asynchronous

Tutorial 1

- Find the number in the list
- Find the shortest string in the list
- Create gender as an enumerated list and show its usage
- Create a Person class in Dart with the following properties:
  - firstName, lastName, age

Include a constructor to initialize these properties and a method named printDetails that prints the full name and age of the person.

Then, create two instances of the Person class and call the printDetails method on each instance.

Tutorial 2

You are building a basic game engine in Dart for a platformer game. You have various types of game objects, each with their own unique behaviors. Implement the following scenario using class inheritance, mixins, and interface implementation.

You have three types of game objects:

- GameObject: The base class that all game objects inherit from. It has properties like x and y coordinates.
- Movable: A mixin that provides the ability to move a game object. It has methods like moveLeft(), moveRight(), moveUp(), and moveDown().
- Drawable: An interface that defines the draw() method.

Tutorial 2 (Contd)

Additionally, there are two types of game objects that inherit from GameObject and use the Movable mixin:

- Player: A game object representing the player character. It implements the Drawable interface and has an additional method, jump().
- Enemy: A game object representing an enemy. It implements the Drawable interface and has an additional method, attack().

Your task is to define these classes and their relationships, utilizing class inheritance, mixins, and interface implementation.

Tutorial 2 (Solution)