

Mobile Application Development

Week 1. Introduction

Joon-Woo Lee

Software School
College of Software
Chung-Ang University



Instructor

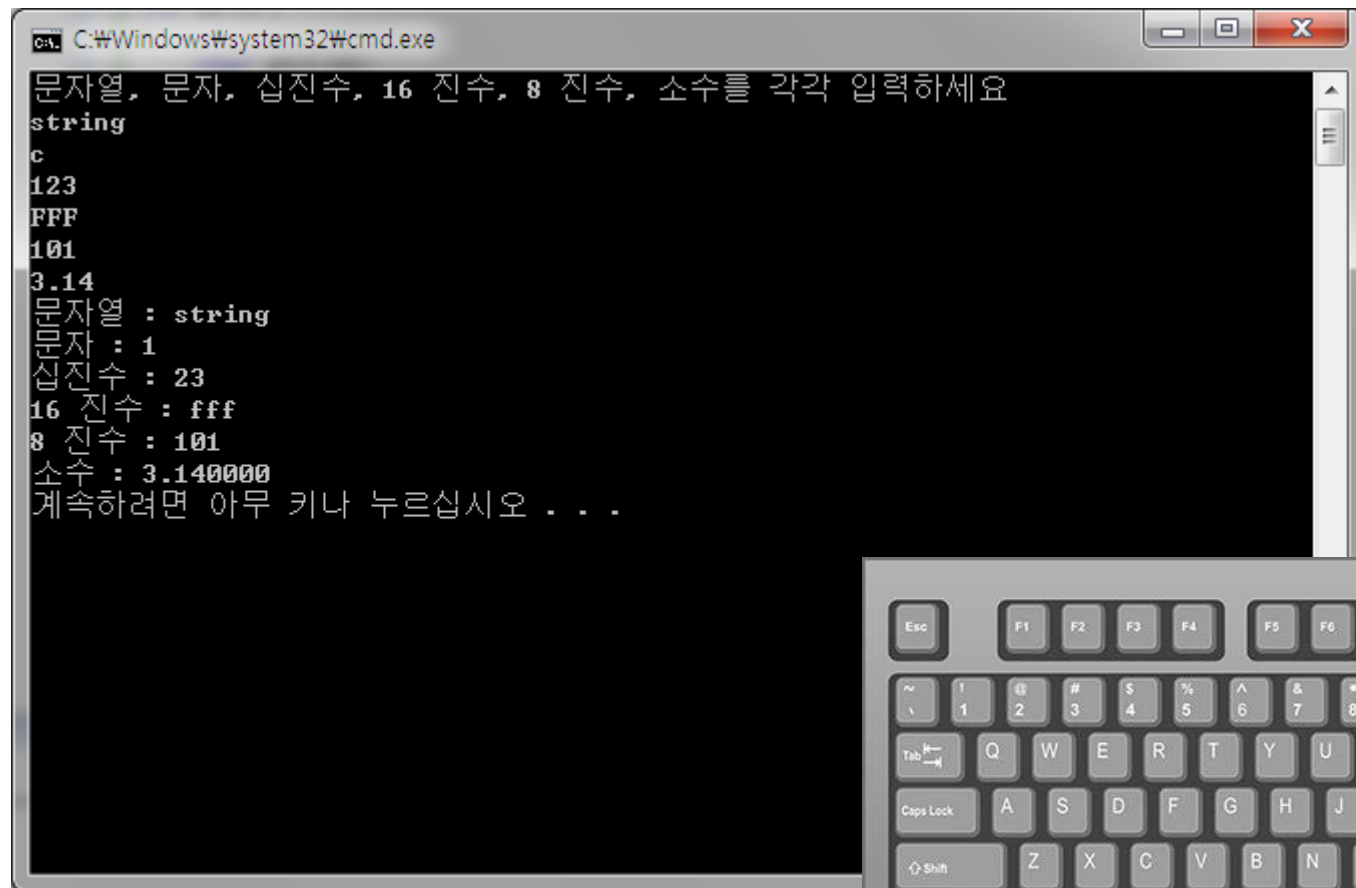
Joon-Woo Lee

- Full-time professor at Software School in Chung-Ang university. (2022.9 ~)
- Research interest
 - Privacy-preserving machine learning
 - Fully homomorphic encryption
 - Post-quantum cryptography
 - Lattice-based cryptography
- Homepage: <https://sites.google.com/view/joonwoolee>
- E-mail: jwlee2815@cau.ac.kr

Expected students for class

- **Students who have not made any mobile application!**
- Students who have knowledge of **objected-oriented programming**.
- Some experience with **functional programming** or **asynchronous programming** will be helpful.

Programs you made so far...

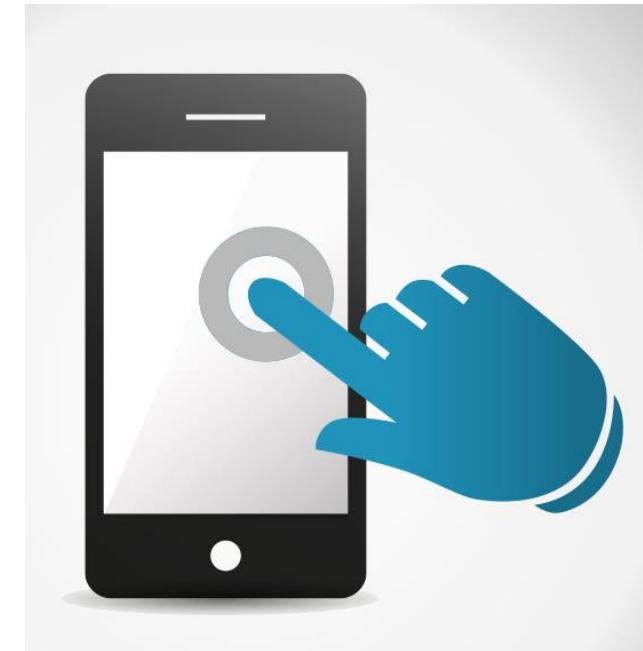
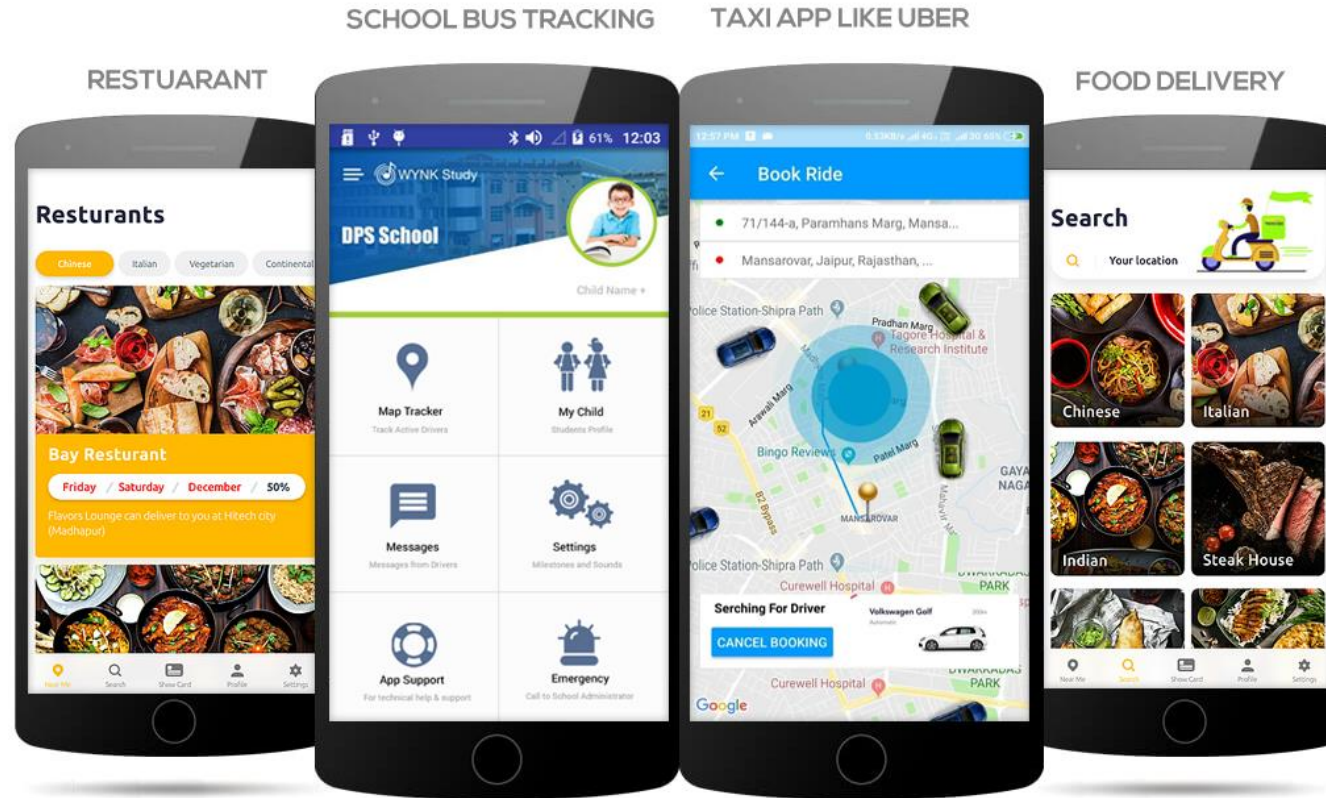


A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window displays the output of a C program. The text is as follows:

```
문자열, 문자, 십진수, 16 진수, 8 진수, 소수를 각각 입력하세요  
string  
c  
123  
FFF  
101  
3.14  
문자열 : string  
문자 : 1  
십진수 : 23  
16 진수 : fff  
8 진수 : 101  
소수 : 3.140000  
계속하려면 아무 키나 누르십시오 . . .
```



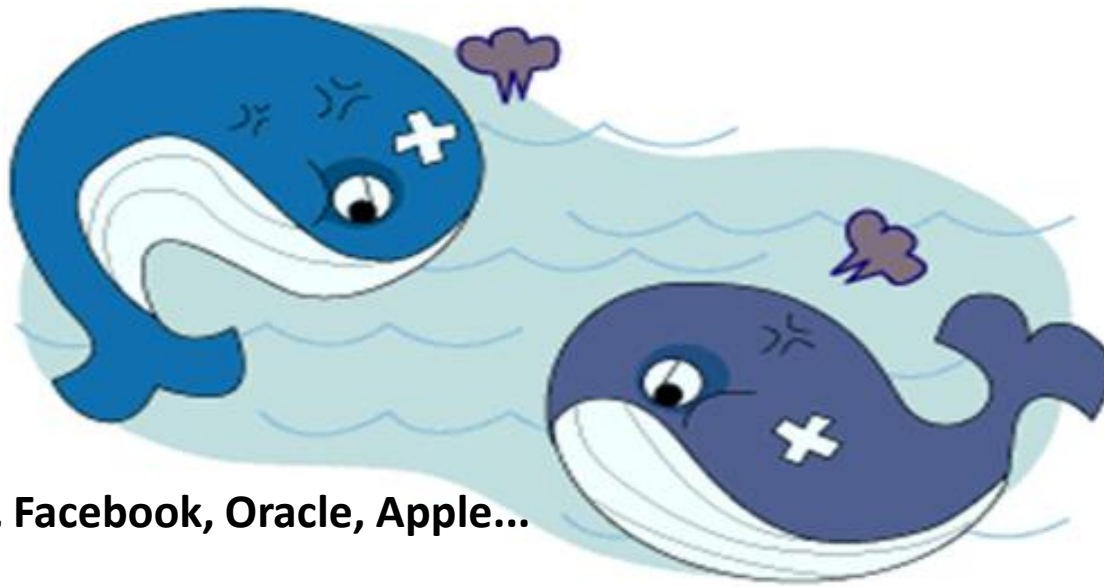
But mobile development seems different...



It seems to be too far from me..

Big companies struggles to make frameworks..

I will be the No.1 company for mobile application by making good framework!



Google, Facebook, Oracle, Apple...



Mobile development is getting more and more easy!!



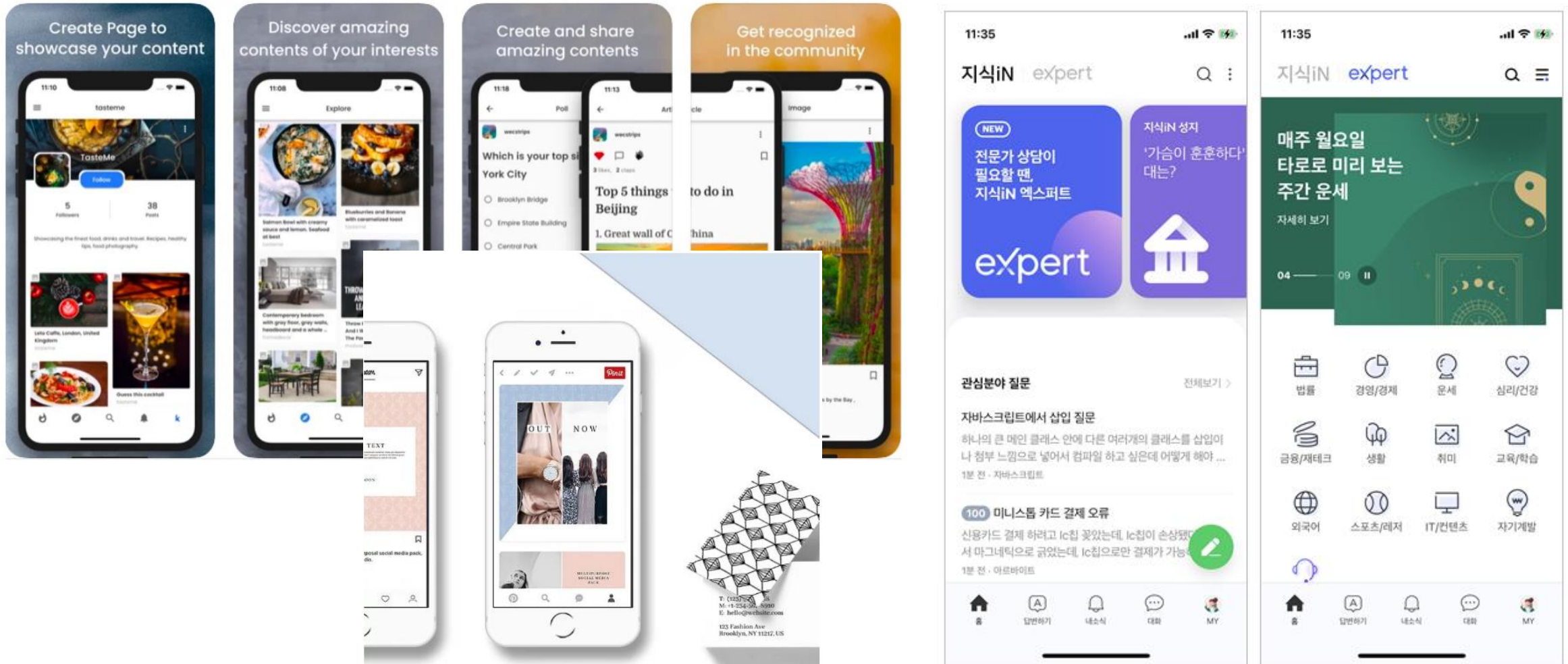
Mobile developers like us!

Flutter

- Open-sourced mobile software development kit (SDK) built and maintained by **Google**.
- Aim: **making everyone build beautiful mobile apps!**
- **“write once, and deploy everywhere.”**
 - current: Android, iOS, ChromeOS
- Flutter for **web apps** and **desktop apps** is actively being developed!
 - Windows, macOS, Linux...




Example applications by Flutter



Comparison of frameworks

Three types for mobile development

- Native development (Android, iOS)
 - React Native
 - Flutter
- 
- Cross-Platform

Native development

- Development with specialized framework for each operating system
- Pros:
 - Fastest performance and natural movement for each operating system
- Cons:
 - Workload is doubled because of large differences of framework.
 - It is hard to communicate with developers for different operating system.



➡ We need cross-platform for large productivity and efficient development!

React Native

- Cross-platform mobile development framework developed by **Meta**, which was **Facebook** before.
- Pros:
 - Android app and iOS app can be developed from **only one codebase**.
 - **Web developers** using React or Javascript can adjust easily to this framework.
- Cons:
 - **Performance** of apps is degraded by **Javascript bridge**.



➡ We need cross-platform with high performance!

Flutter

- Cross-platform mobile development framework developed by **Google**.
- Pros:
 - Android app and iOS app can be developed from **only one codebase**.
 - **Performance** of apps are **almost the same** as native-developed apps.
- Cons:
 - Since it is recently developed, community size is somewhat small. (yet it is **getting large fast!**)



➡ **Flutter is the cross-platform with high performance!**

Will learning Flutter be helpful to us?

1. Flutter is **easy to learn**! Flutter makes you familiar with mobile development!
2. Many enterprises have lots of interests in Flutter. Flutter will be helpful to **getting jobs** soon.
3. Flutter is similar to other SDK. You can **learn other frameworks fast** if you master Flutter!

 **Learning Flutter is helpful to you!**

React Native vs Flutter

- Flutter is rather recent framework.
- Many mobile applications are already developed with React Native.
- However, Google is putting so many resources into Flutter!

Which companies will mainly use each frameworks?

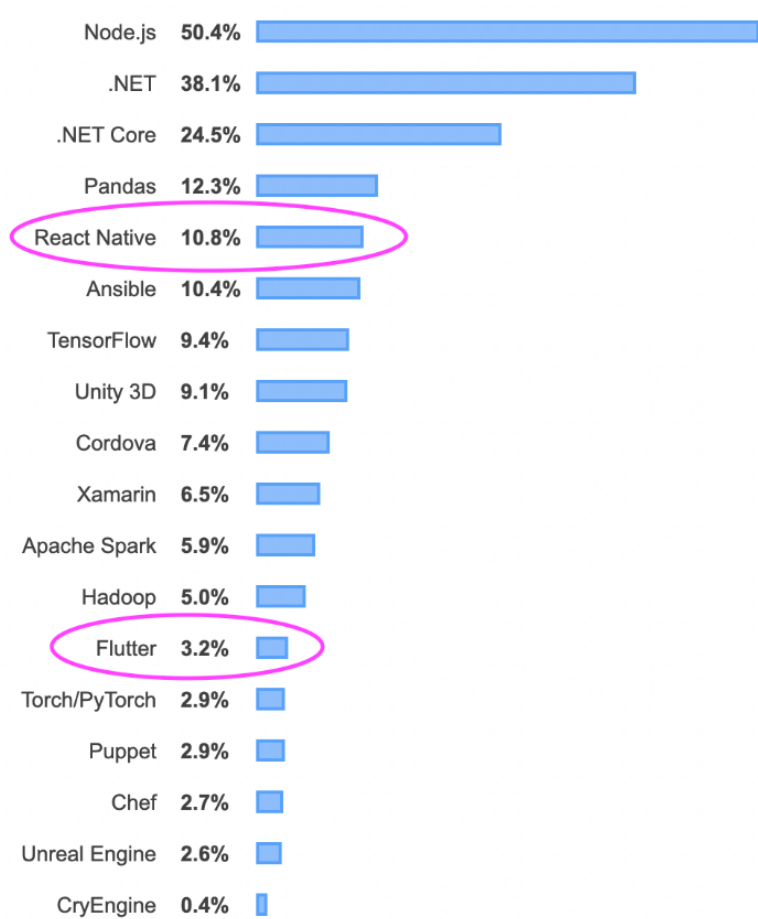
Flutter

- Start-up companies
- Companies which needs to develop new apps.

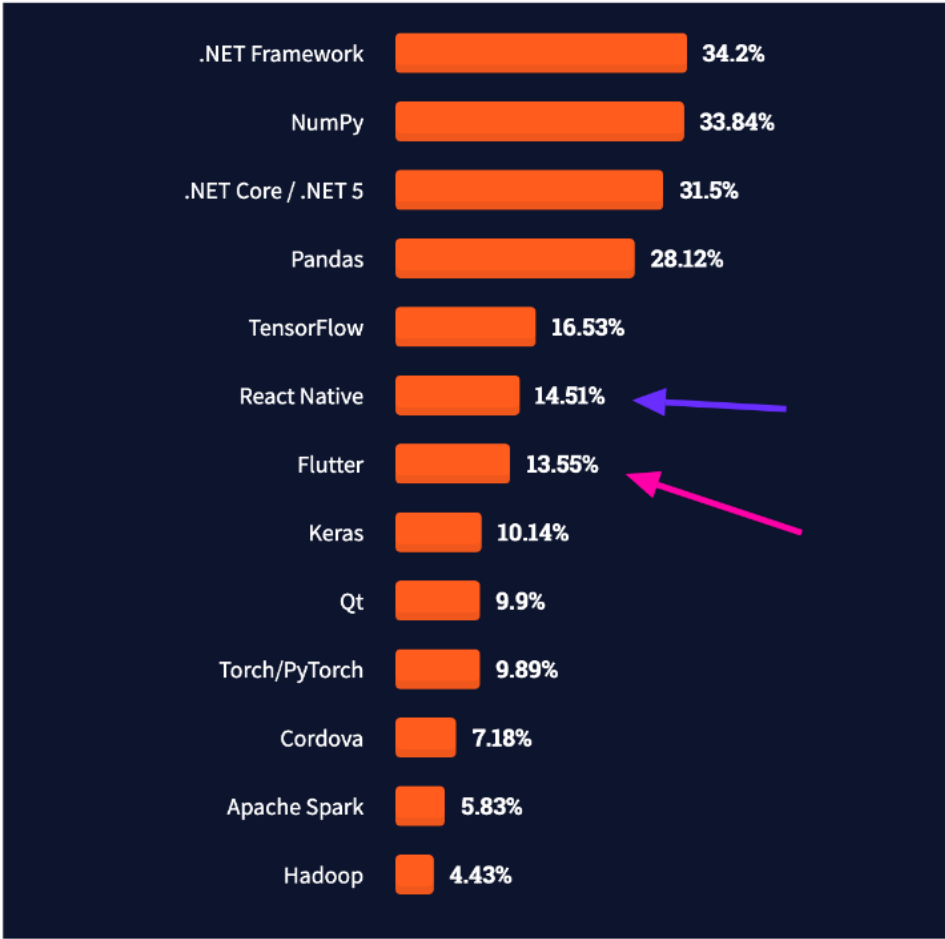
React Native

- Companies which need to maintain pre-developed apps.

Survey result for popularity (Stack Overflow)

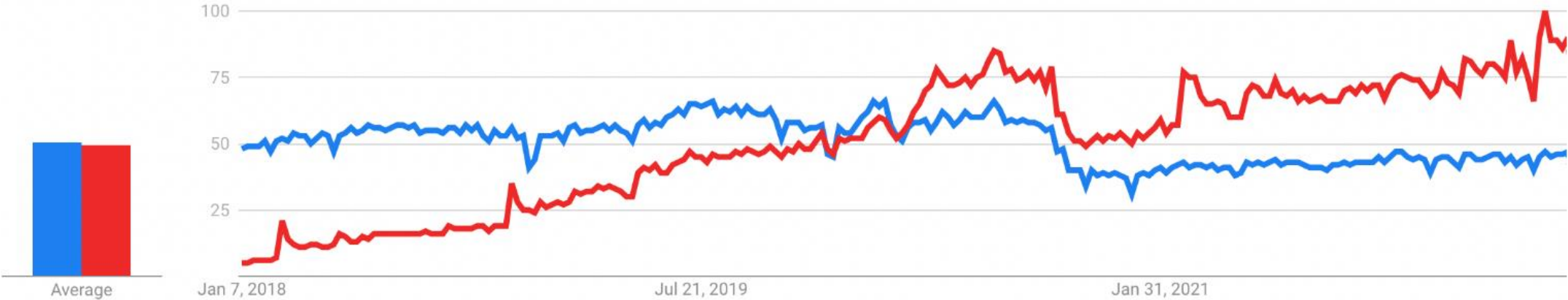


2019

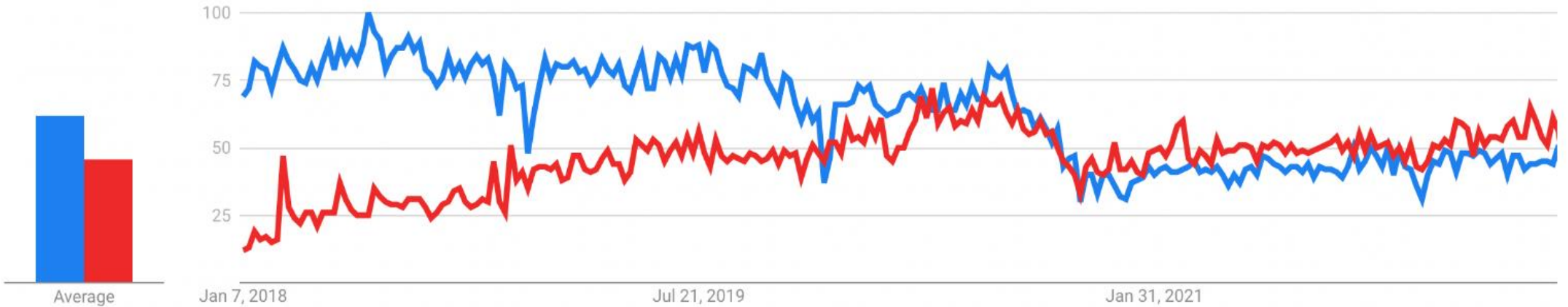


2021

Google Trends: React Native vs Flutter

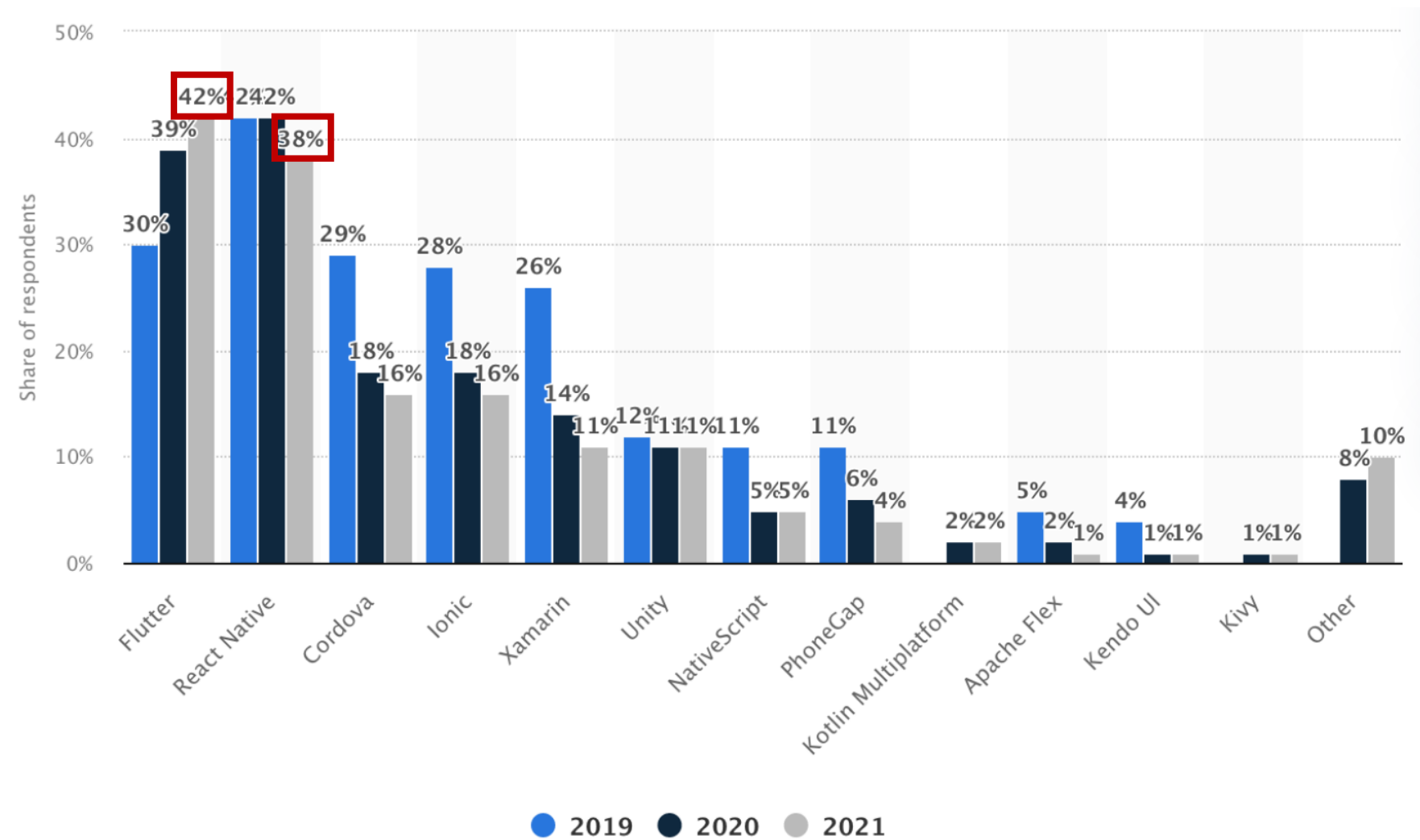


Worldwide (2018-2022) (Blue: React Native, Red: Flutter)



United States (2018-2022) (Blue: React Native, Red: Flutter)

Survey result for popularity (Statista)



Google's Fuchsia project and Flutter

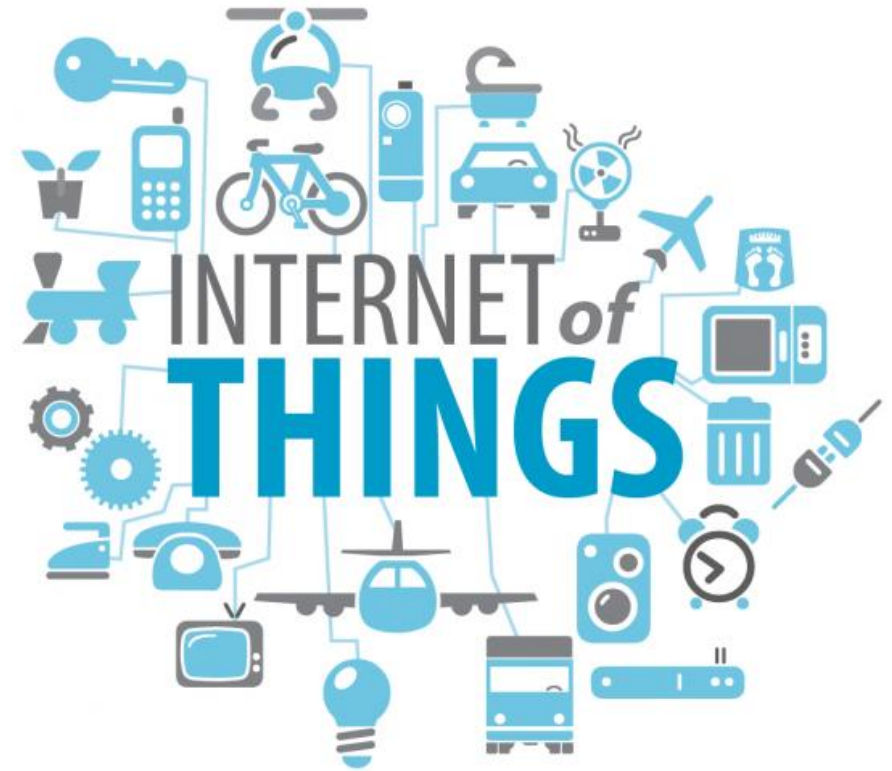
Fuchsia operating system

- Since 2016, Google is actively developing **entirely new** operating system, **Fuchsia OS**.
- While other operating systems developed by Google uses Linux kernels, it uses **Zircon** kernel.
- It will be working on all types of hardware, including:
 - smartphones, tablets, computers and an increasing number of smart home devices.
- Main developing tool: **Flutter**!



Internet of Things (IoT)

- **Internet of Things** is the concept of connecting **any device** to the Internet and to other connected devices.
 - Smart microwave
 - Self-driving cars
 - Wearable fitness devices
 - Connected footballs
- Many experts analyze that Google is developing **Fuchsia OS** for upcoming **IoT era**.



Fuchsia OS Project and Flutter

- Replacing operating system is somewhat difficult if the users are not adjusted to this new OS.
- Many experts analyze that Google is trying to make developers adjust to **Flutter** framework, which will be the main framework in **Fuchsia OS**.
- Google is strongly expected to support Flutter actively for a long time because of the big project of Fuchsia!



Dart Programming Language

Dart language

- The only programming language Flutter uses is **Dart** programming language.
- Dart is a programming language developed by **Google** for web and mobile programming.
- Dart is very similar to other OOP languages!
 - Java, C++, Python, ...

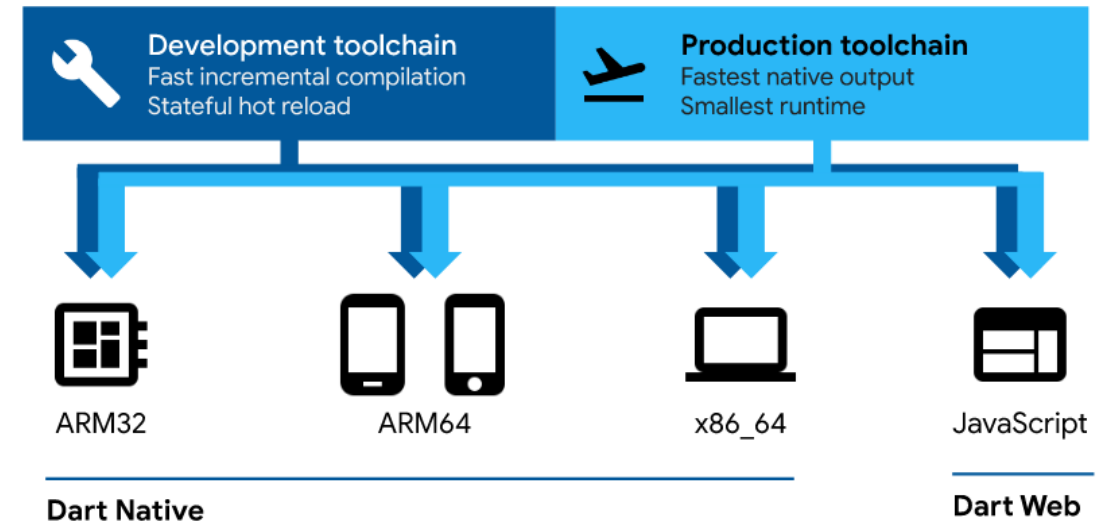


Dart



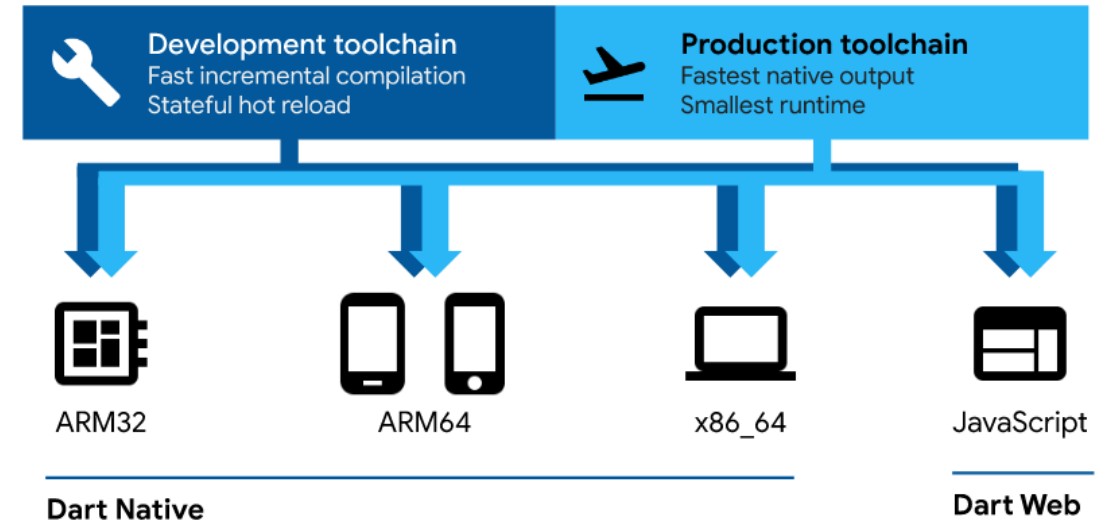
Why Flutter use Dart? (Compilation method)

- Dart supports both two compilation methods.
 - **Just-In-Time (JIT)** compiler: it converts source code into native machine code just before program execution.
 - ✓ Fast development cycle! (Hot reload)
 - **Ahead-Of-Time (AOT)** compiler: it compiles source code before it is “delivered” to whatever runtime environment runs the code.
 - ✓ Great performance of applications!



Why Flutter use Dart? (Compilation destination)

- Dart can be compiled into **native machine code**, which makes the applications fast.
 - JavaScript cannot be compiled into native machine code, so the resultant applications is not so fast.
- For web applications, Dart also supports compilation into **JavaScript**.



Why Flutter use Dart? (Google has it!)

- **Google** has been involved a big lawsuit by **Oracle** with the use of **Java API** in Android mobile development.
 - Google won this lawsuit from this 10-year patent dispute.
- Many expert analyzes that Google want to be free from this **patent dispute** for programming language.
- Google can also feel free to develop Dart programming language in the way that Flutter needs.



Advantage of Dart language



- Easy to learn for OOP programmers
- Great performance
- Fast development
- Support for various platforms and devices
- Active support for Flutter

Class information

What can be learned from this class?

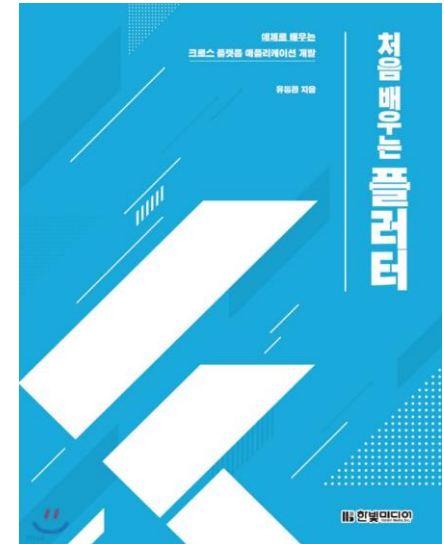
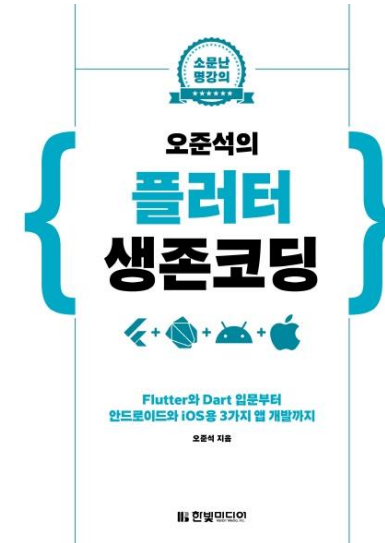
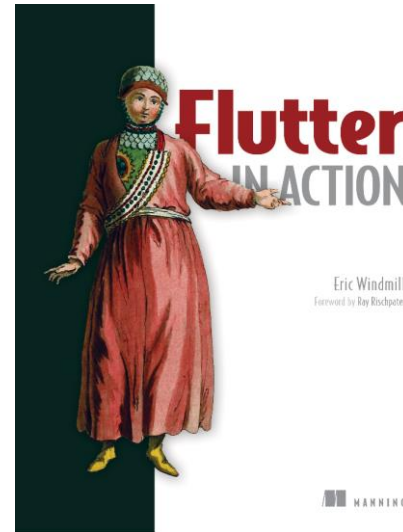
Short answer:

Learning Flutter!

- You can make any applications which you want!
- Good starting point of mobile development even if you are targeting to other frameworks!

Textbook for the class

- Three books are referred to in this class.
 - Flutter in action
 - 오준석의 플러터 생존코딩
 - 처음 배우는 플러터
- **You do not need to buy these books for this class**, but these will be good reference books for Flutter.
- Difficulty
 - Flutter in action > 오준석의 플러터 생존코딩 = 처음 배우는 플러터



Assessment

- Attendance (10%)
- Mid-term exam (30%) : 10/20 (Thu)
- Final exam (30%) : 12/15 (Thu)
- Assignment (30%)
 - Exercise projects
 - Term projects

Grade guide

- Grade Distribution
 - Grade distribution guideline (Da Vinci Learning)
 - A+/A0 : < 50%
 - A+/A0/B+/B0 : < 90%
- **'F' grade Policy** : You may get a 'F' grade for....
 - Any kind of cheating or plagiarism (homework, exam, quiz, etc.)
 - Absent for 4 times or more of the total classes.
 - Not taking an exam, or an exam score below 3 for any of the exams.
 - Inappropriate attitude during lectures.
 - Overall total score below 20 (out of 100)

Three steps for taking class

- Following example code (pre-recorded video uploaded on Saturday)
- Theory (face-to-face class on Tuesday)
- Practice (Personal exercise and individual teaching on Thursday)
 - You may end the class early on Thursday if you solve the exercise problem faster.
 - You have to submit the source code for each practice problem until the end of the class.

Course plan

| Week | Pre-recorded video | Theory (Tuesday) | Practice (Thursday) |
|------|--------------------|----------------------------|--|
| 1 | | | 9/1 |
| | | | Introduction |
| 2 | | 9/6 | 9/8 |
| | | Dart introduction | No class (Pre-recorded video: Flutter installation) |
| 3 | 9/13 uploaded | 9/13 | 9/15 |
| | Basic layout | Characteristics of Flutter | Making term project team / Basic layout practice |
| 4 | 9/17 uploaded | 9/20 | 9/22 |
| | Advanced layout | Functional programming | Advanced layout practice |

Course plan

| Week | Pre-recorded video | Theory (Tuesday) | Practice (Thursday) |
|------|---------------------------|--|---------------------------|
| 5 | 9/24 uploaded | 9/27 | 9/29 |
| | Form and gesture | Tree structure in Flutter | Form and gesture practice |
| 6 | 10/1 uploaded | 10/4 | 10/6 |
| | Animation | Principles in animation | Animation practice |
| 7 | 10/8 uploaded | 10/11 | 10/13 |
| | Routing | Principles in routing | Routing practice |
| 8 | | 10/18 | 10/20 |
| | No video (preparing exam) | Special lecture (Flutter developer) | Mid-term exam |

Course plan

| Week | Pre-recorded video | Theory (Tuesday) | Practice (Thursday) |
|------|--------------------------|---------------------------------|---|
| 9 | | 10/25 | 10/27 |
| | No video (during exam) | No class (during exam) | Project plan presentation |
| 10 | 10/29 uploaded | 11/1 | 11/3 |
| | State management | State concept in Flutter | State management practice |
| 11 | 11/5 uploaded | 11/8 | 11/10 |
| | Asynchronous programming | Asynchronous programming theory | Asynchronous programming practice |
| 12 | 11/12 uploaded | 11/15 | 11/17 |
| | Working with data | Data processing in Flutter | Required specification in term project / Data processing practice |

Course plan

| Week | Pre-recorded video | Theory (Tuesday) | Practice (Thursday) |
|------|----------------------------------|-----------------------------------|-----------------------------|
| 13 | 11/19 uploaded | 11/22 | 11/24 |
| | Testing | Principles in testing | Testing practice |
| 14 | 11/26 uploaded | 11/29 | 12/1 |
| | Web development with Flutter | Principles in web development | Term project |
| 15 | 12/3 uploaded | 12/6 | 12/8 |
| | Desktop development with Flutter | Principles in desktop development | 7-minute final presentation |
| 16 | 12/10 uploaded | 12/13 | 12/15 |
| | How to learn on your own | 7-minute final presentation | Final exam |

Term project

- Make a mobile application which your team wants!
- Timeline
 - 9/15 Deadline for making term project team
 - 10/6 Deadline for proposal documentation about the functions of the mobile application
 - 10/27 5-minute presentation for proposal
 - 11/17 Required specification for the mobile application
 - 12/6 Deadline for source code
 - 12/8, 12/13 Final 7-minute presentation for the proposed mobile application
- Assessment
 - (40%) Did your application satisfy the required specification?
 - (60%) Peer review (All of you will assess other team's application)

Let's enjoy learning Flutter!