**Flutter**

- Google에서 개발한 Cross Platform development framework

- iOS, Android 및 Windows, MacOS, Linux 등의 Application 제작 가능

**Advantage**

- Easy: 쉬운 언어인 Dart 사용, UI & Logic 코드가 하나의 파일에서 생성

- High Productive: 여러 device에 동시 적용 가능, 객체 지향 프로그래밍, 함수형 프로그래밍도 지원

- Flexible: 가변적인 display에 특화

- High Performance: 별도의 UI 엔진을 활용, C / C++ 언어 기반의 엔진으로 호환성 및 성능 높음

- Open Source: Google이 개발하였지만 누구나 Flutter의 소스코드에 기여 가능

- Simple & Beautiful: 기본 UI Widget들의 디자인이 예쁨

**Dart**

- Google에서 개발한 고급 프로그래밍 언어

- 객체지향 프로그래밍 언어

- 두가지의 컴파일러를 보유 (JIT, AOT)

- JIT Compiler (Just In Time)  
 : 코드 실행 시점에서 Compile -> 코드를 수정하면 즉각적으로 반영  
 (Hot-Reload / Hot-Restart를 구현할 때 용이)

- AOT Compiler (Ahead Of Time)

: 높은 실행속도를 필요로 할 때 사용 -> 배포 직전 Release 용으로 Compile

- 강력한 타입 시스템을 가짐

**Type**

- 특정한 값을 담아두는 그릇

- 기본형, 확장형으로 구분  
 (bool, int, double, string, null -> 기본형 / Object, Enum, Future, Stream -> 확장형)

- Type 뒤에 ? or !을 붙여 nullable or non-nullable을 선언

가변형 Type

- var: 최초 한 번 부여된 type이 고정적으로 사용

- dynamic: Type이 코드 진행 중에라도 언제든지 변환

텍스트, 멀티미디어 소프트웨어, 소프트웨어, 그래픽 소프트웨어이(가) 표시된 사진

자동 생성된 설명

상수 선언

- const: Compile 시점에 상수 처리 될 경우에 활용

- final: 프로그램의 진행 중 상수 처리 될 경우에 활용

**사칙연산**

- Divide ( / )의 결과물은 double

- String의 sum은 “ + “ 보다는 ‘$variable ~~~’ 형태로 $의 사용을 권장

- 삼항 연산자 (논리연산 ? 참일 경우 : 거짓일 경우)

- Null 처리 (? -> null 허용, ! -> null 비허용)

- (변수 ?? 5) -> 변수가 null일 때, 변수의 값으로 5를 사용

텍스트, 폰트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명

- Nullable 변수에 !를 사용하는 경우, 변수에 값을 지정해주어야 한다. ( ! 는 null 비허용을 의미)

**Class**

- Object를 만들기 위한 Template

구성요소

- Field: Class 내부에 선언된 data (variable, const data)

- Method: Class 내부에 선언된 function

- Constructor: Class instance를 생성할 때 사용되는 Code, 생성 시 특정 작업을 지시하는 등의 활용 가능

Class 생성 Code example

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Switch – case 문

- Dart 3.0.0 부터는 case 조건에 비교문도 가능

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

For 문

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

Error Exception

<https://m.blog.naver.com/mingdyuo/221803704762>

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Error: IntegerDivisionByZeroExcepion

나머지는 stack

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

일반적으로 try – catch – finally 가 세트로 사용

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Error Type에 따른 분류 가능

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Throw는 Error 발생

Rethrow는 해당 프로그램 밖으로 Error를 전달

Flutter UI의 기본 단위: Widget

* 간단한 블록부터 텍스트 입력 창, 화면 내 특정 요소 간의 간격조차 Widget으로 관리

Widget Tree

* Flutter의 Widget은 마치 블록과 같음
* Widget들 간의 관계를 나타내는 형태 (수직적 또는 수평적 관계)

Stateless Widget / Stateful Widget

* 화면 갱신이 필요 없는 정적 화면 -> Stateless Widget
* 화면 갱신이 필요한 동적 화면 -> Stateful Widget

Android Studio Tips

* Ctrl + Alt + L -> 자동 정리
* Alt + Enter -> 감쌀 Object 추가 가능

Hot restart

* App을 재시작 하는 형태
* App의 모든 상태가 초기화 & 코드 변경사항 적용

Hot reload

* App을 재시작하지 않고 변경사항을 반영

Hot restart & Hot reload 주의사항

* App을 재컴파일 후 빌드하는 것이 아니므로 Native Code 수정, Library나 Asset의 추가등은 반영되지 않음
* 지나치게 많은 변경사항이 있을 경우 상위 재시동이 요청됨

텍스트, 폰트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명

번개: Hot reload

되돌이 번개: Hot restart

Design Guide

- Material Design: Google에 맞는 Design Guide -> Android에 적합

- Cupertino Design: Apple에 맞는 Design Guide -> iOS에 적합

Materials 3를 발표하면서 개인화된 App을 구현하는데 중점

Cupertino Widget보다 Material Widget이 더 많이 존재

Widget

1. MaterialApp

- 모든 하위 페이지나 구성요소를 최상위에서 담는 그릇

- title, theme, home은 기본 옵션 (home은 필수)

- 가장 먼저 나오는 페이지는 home

- 페이지는 Scaffold라는 형태를 가짐

2. Scaffold

- 필요한 기본적인 것들이 포함된 widget

- 화면 구성의 기본이 되는 Widget