



다트 입문

by 이상용

=

01 다트 입문 요약

- 다트는 구글이 만든 프로그래밍 언어로, Flutter 개발에 필수
- UI 구성에 강하고 비동기 처리와 동시성, AOT 컴파일, 멀티플랫폼 지원 등 강력한 기능 보유

1. 닥트 특징 요약

- UI 제작 최적화
- null safety, collection if, spread operator 지원
- hot reload, 디버깅, 멀티플랫폼 실행 가능
- AOT 컴파일, JavaScript 완전 컴파일, 백엔드 지원

2. 닥트 컴파일 플랫폼 요약

- 네이티브(x64/ARM)
- 개발 시: JIT + VM (hot reload, 디버깅)
- 배포 시: AOT + 런타임
- 웹
- 개발 시: dartdevc
- 배포 시: dart2js

3. 문법 학습 환경

- DartPad: <https://dartpad.dev>
- Android Studio main.dart에 코딩 후 `dart lib/main.dart` 명령 실행

4. 기초 문법

- `main()` 함수: 프로그램 시작점
- 주석: `//`, `/* */`, `///` 사용
- `print()`: 출력 함수

5. 변수 선언

- var: 타입 자동 추론
- dynamic: 어떤 타입이든 저장 가능
- final/const: 수정 불가능, 런타임/빌드타임 상수

6. 기본 타입

- String, int, double, bool 등 기본 제공
- 명시적 타입 선언으로 유지보수 편리함

7. 컬렉션

- List: 순서 있는 값 저장
- Map: 키-값 저장
- Set: 중복 제거 집합

8. List 주요 함수

- add(): 값 추가
- where(): 조건 필터링
- map(): 값 변경
- reduce(), fold(): 값 축약

9. Map과 Set 주요 기능

- Map: .keys, .values 접근 가능
- Set: 중복 제거, contains(), toList() 등 사용 가능

10. enum 사용

- 제한된 값 정의
- 예: `enum Status { approved, pending, rejected }`

11. 연산자

- 기본 연산자: + * / %
- 비교 연산자: > < >= <= == !=
- 논리 연산자: && || !
- 타입 비교: is, is!

12. 제어문

- if/else: 조건 분기
- switch: enum과 함께 사용 유용
- for: 일반, for-in 루프 제공
- while / do-while: 조건 기반 반복

13. 함수

- 매개변수: 포지셔널, 네임드, 기본값 지정 가능
- 익명 함수 / 람다 함수: 일회성 함수로 콜백에 유용
- typedef: 함수 시그니처 정의

14. 예외 처리

- try-catch: 오류 포착
- throw: 에러 직접 발생
- 에러 발생 시 catch 블록 실행

15. 핵심 요약

- var로 변수 선언, dynamic은 유연한 타입
- 컬렉션: List, Map, Set
- 제어문과 반복문 풍부
- 함수: 재사용, 람다, typedef 지원
- JIT (개발), AOT (배포) 컴파일 방식 지원