

# 앱프로그래밍응용

---

크로스플랫폼 프로그래밍  
플러터를 내 것으로!

# 강사

---

정세일 겸임 교수 (dimohy)

Microsoft MVP 22/23/24

[dimohy@g.shingu.ac.kr](mailto:dimohy@g.shingu.ac.kr), [dimohy@slogs.dev](mailto:dimohy@slogs.dev)

<https://forum.dotnetdev.kr>, <https://forum.slogs.dev>

C / C++ / C# / Java / TypeScript / Python / Rust

front-end : Vue, Blazor

back-end: ASP.NET Core / Spring Boot



# 강의 소개

---

- 앱프로그래밍기초 강의에서 더 나아가
  - 앱 개발 환경의 공통 분모 확인
  - 크로스플랫폼 개발 환경 이해
  - 크로스플랫폼 개발 이점 이해
- 만들고자 하는 것을 다양한 플랫폼에 배포할 수 있는 능력 함양

# 강의교재



오준석의

# 플러터 생존코딩

개정판



Flutter와 Dart 입문부터  
안드로이드와 iOS용 3가지 앱 개발까지

오준석 지음

플러터 2  
대응

한빛미디어  
Hanbit Media, Inc.

# 플러터란?

---



<https://flutter.dev/>

- 구글에서 개발한 **개발 플랫폼**으로, Dart 언어를 사용하며, Skia 그래픽 엔진을 사용해서 **다양한 플랫폼**에서 구동 가능한 앱을 만들 수 있음
- 크로스플랫폼 지원
- 고성능
- 사용자 정의 UI 구성 용이
- 빠른 개발
- 강력한 커뮤니티 및 생태계

# 학생이 플러터를 배워야 하는 이유

---

- 크로스플랫폼 개발 역량
- 빠른 개발 피드백과 디버깅 : 핫리로드(Hot-Reload)
- 커리어 기회 : 성장하는 기술, 스타트업 회사
- 커뮤니티와 리소스
- 창의성 발휘 : 빠르고 쉽게 빌드-업

# 앱프로그래밍기초에 이어서...

---

- 전혀 다른 개발 환경 / 그러나 동일하거나 유사한 구성, 확인하면서 개발  
“필수!”
- 수평적 사고 계발
- 안드로이드, 아이폰, 윈도우, 리눅스, ...

“원하는 것을 플랫폼 상관없이 개발 하는 능력 함양”

# 강의 계획

---

1주차 - 강의 소개 및 개발 환경 구축

2주차 - 다트 문법 1

3주차 - 다트 문법 2

4주차 - 프로젝트 구조와 앱 구조

5주차 - 기본 위젯

6주차 - 내비게이션

7주차 - 복잡한 UI 설계 및 구현

8주차 - 객관식 중간평가

앱 개발 개인 프로젝트 진행

9주차 - 앱 개발 계획

10주차 - 프로젝트 진행

11주차 - 프로젝트 진행

12주차 - 앱 개발 중간 보고

13주차 - 기본 위젯

14주차 - 프로젝트 진행

15주차 - 프로젝트 발표 기말평가



# 1주차

---

목표 : 플러터와 다트 개발 환경을 이해하고 설정할 수 있다.

학습내용:

- 플러터와 다트 소개
- 개발 환경 설치 (Flutter SDK, Android Studio)
- 첫 번째 플러터 앱 실행

# 2주차

---

목표 : (1/2) 다트의 기본 문법을 이해하고 사용할 수 있다.

학습내용:

- 변수와 자료형
- 연산자와 제어문
- 함수 정의 및 호출

# 3주차

---

목표 : (2/2) 다트의 기본 문법을 이해하고 사용할 수 있다.

학습내용:

- 클래스와 객체지향 프로그래밍
- 예외 처리
- 컬렉션 (List, Set, Map)

# 4주차

---

목표 : 플러터 프로젝트의 구조를 이해하고 설명할 수 있다.

학습내용:

- 플러터 프로젝트 파일 구조
- 앱의 생명주기
- 상태 관리 개념

# 5주차



목표 : 플러터의 기본 위젯을 이해하고 사용할 수 있다.

학습내용:

- 텍스트, 버튼, 이미지 위젯

# 6주차

---

목표 : 플러터에서 화면 전환과 네비게이션을 구현할 수 있다.

학습내용:

- Navigator와 Route
- Named Route
- 화면 전환 애니메이션

# 7주차

---

목표 : 복잡한 UI를 설계하고 구현할 수 있다.

학습내용:

- 리스트와 그리드 뷰
- 커스텀 위젯 만들기
- 테마와 스타일링

# 8주차

---

목표 : (중간고사) 1 ~ 7주차 동안 배운 내용을 리와인드 한다.

평가내용:

- 개발 환경 구축
- 다탈 문법
- 프로젝트 구조와 앱 구조
- 기본 위젯
- 네비게이션
- 리스트, 그리드, 커스텀 위젯, 테마, 스타일링



# 9주차

---

목표 : 앱 개발 프로젝트를 계획하고 설계할 수 있다.

학습내용:

- 프로젝트 주제 선정
- 요구사항 분석 및 설계
- 개발 일정 계획

# 10주차

---

목표 : 플러터 앱 개발 프로젝트 진행

학습내용:

- 프로젝트 초기 설정
- 기본 기능 구현

# 11주차

---

목표 : 앱의 주요 기능을 구현하고 테스트할 수 있다.

학습내용:

- 주요 기능 개발
- 단위 테스트 작성

# 12주차

---

목표 : 프로젝트 진행 상황을 점검하고 피드백을 받는다.

학습내용:

- 중간 보고서 작성
- 피드백 세션 진행

# 13주차

---

목표 : 앱의 고급 기능을 구현하고 최적화 할 수 있다.

학습내용:

- 고급 기능 개발
- 성능 최적화

# 14주차

---

목표 : 앱 개발을 마무리하고 배포 준비를 할 수 있다.

학습내용:

- 버그 수정 및 최종 테스트
- 앱 배포 준비

# 15주차

---

목표 : 개발한 앱을 발표하고 평가 받는다.

평가내용:

- 발표 준비 및 발표

# 강의 목표

---

우선순위로

1. 취업에 강력히 도움이 되는 포트폴리오가 됨
2. 수평적 사고가 무엇인지 완전히 이해함
3. 컴퓨팅 사고를 이해하고 조금이라도 적용함
4. 크로스플랫폼 개발 환경의 이점을 이해하고 활용하게 됨
5. 플러터로 모델링 할 수 있게 됨 (플러터로 프로토타입을 만들 수 있게 됨)



# 과제

---

- 목표 : 꾸준한 활동으로 취업 시 강력한 플러스 점수가 된다.
- 과제 방안
  - 많은 양의 과제는 지양
  - 하지만 꾸준한 활동을 할 수 있도록 가이드 - 예) GitHub 활동

“20점 배점”

# 중간고사

---

- 목표 : 강의의 중요 용어와 개념을 이해하고 말할 수 있다.
- 평가 방법
  - 객관식 4지선다형 ?
  - 발표 ?
- 학생들의 의견 및 학습 효과를 고려하여 최종 결정할 예정

“30점 배점”

# 기말고사

---

- 목표 : 스스로 발표하는 힘을 함양한다. 취업 시 포트폴리오 자료가 된다.
- 평가 방법
  - 학교에서 요구한 자료가 아닌 것으로 작성해야 함
  - 이빠야 함 - 이후 포트폴리오 자료가 될 것이므로
  - A-Z 까지 스스로 설명할 수 있어야 함
  - 자료에서 어떤 질문을 하든지 답변할 수 있어야 함

“30점 배점”

# GitHub 활용

# GitHub를 해야 하는 이유

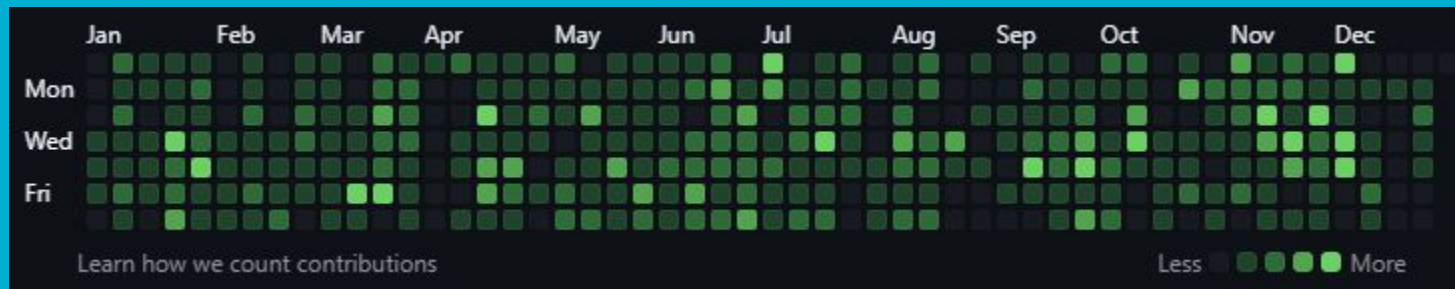
- 활동의 기록은 시간을 거슬러 올라갈 수 없기 때문
- 면접 시 - 꾸준하고 성실하며 성장하고 성장 가능성이 있음을 PR할 수 있으므로 (회사의 인재상 - 취업률 높아짐)



※ GitLab 등 다른 Git 서비스를 이용해도 상관 없음

# 이상적인 GitHub 활동이란?

- 일주일에 3~4회 이상, 주말에도 Commit
- 특정일에 다수 Commit



# 좀 티나는 활동은?

—



나쁘지 않지만 GitHub에 Commit하는 것이 목적인 것 처럼 보임

\_\_\_\_\_



# 강의 시 GitHub 활용 방안

---

- 매주 학습 한 내용을 GitHub에 올릴 것 - 커밋 스트릭
  - 문서 Commit
  - 코드 Commit

※ 본 강의 뿐만 아니라 대학교에서 배우는 모든 강의의 복습을 GitHub에 올릴 것

※ 그냥 Github가 저장소라고 생각하고 활용하는 것이 좋음

# 1년 뒤의 여러분의 GitHub은...

Best!



또는,

Good!



# 컴퓨팅 사고 활용

# 컴퓨팅 사고란?

---

컴퓨팅 사고는 문제를 해결하거나 시스템을 설계하고 자동화 하거나 인간 행동을 이해하는 등 문제를 해결하는데 필요로 하는 사고 과정 중의 하나

- |             |                              |
|-------------|------------------------------|
| - 분해        | Decomposition                |
| - 패턴 인식     | Pattern Recognition          |
| - 추상화 / 일반화 | Abstraction / Generalization |
| - 알고리즘 설계   | Algorithm Design             |

# 컴퓨팅 사고를 활용하려면?

기존 방식

: 언어학습 + 자료구조 + 알고리즘 → 프로그래밍 능력 함양

컴퓨팅 사고 방식

: 컴퓨팅 사고 + 알고리즘 → 논리 프로그래밍 능력 함양

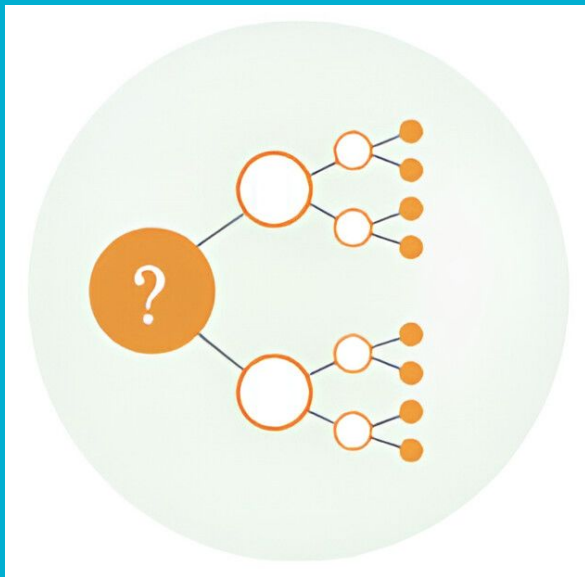
: 프로그래밍 언어 학습 → 물리 프로그래밍 능력 함양

: 프로그래밍 능력 = 논리 + 물리 프로그래밍

# 문제 분해

---

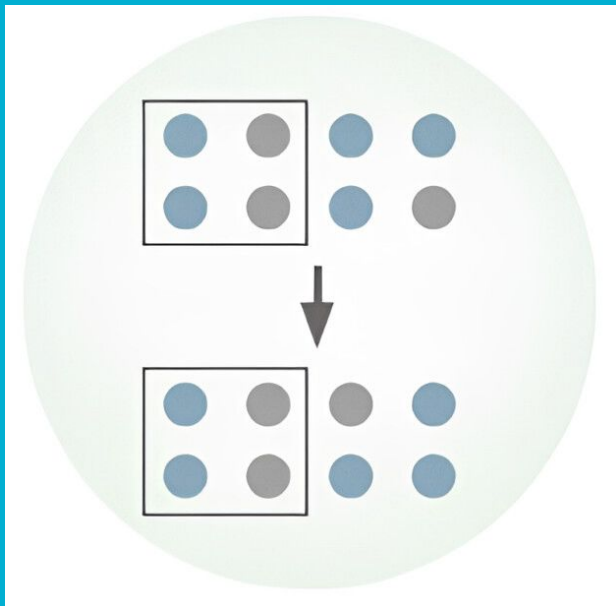
큰 문제 하나보다 작은 문제들이 해결하기 훨씬 쉽다



# 패턴 인식

---

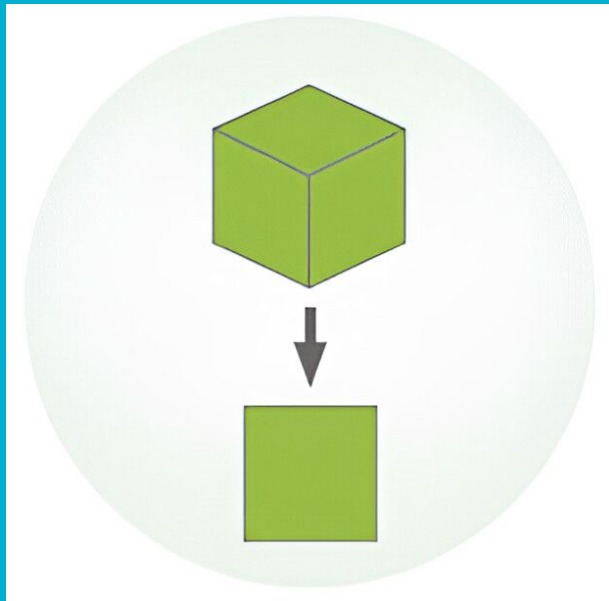
패턴을 알게 되면 복잡해 보이는 문제가 단순해 진다



# 추상화 / 일반화

---

대상을 단순하게 볼 수록 문제는 쉬워진다

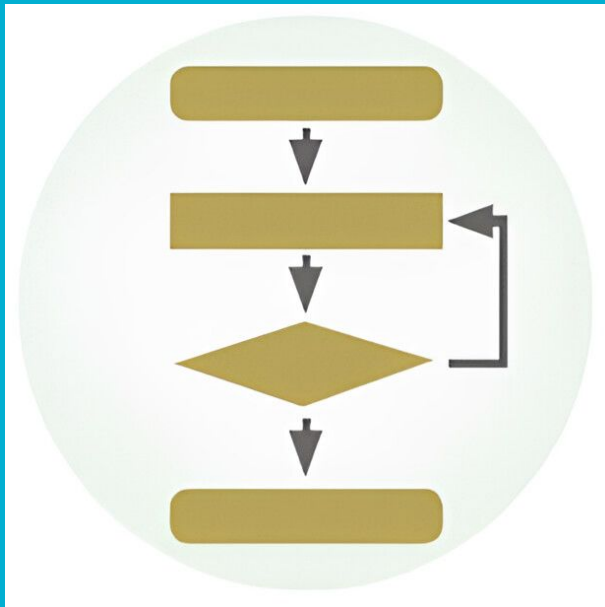




# 알고리즘 설계

---

이제 문제 해결을 순서대로 전개하자



# 인공지능 활용

# 인공지능 (AI)를 왜 학습에 활용해야 하는가?

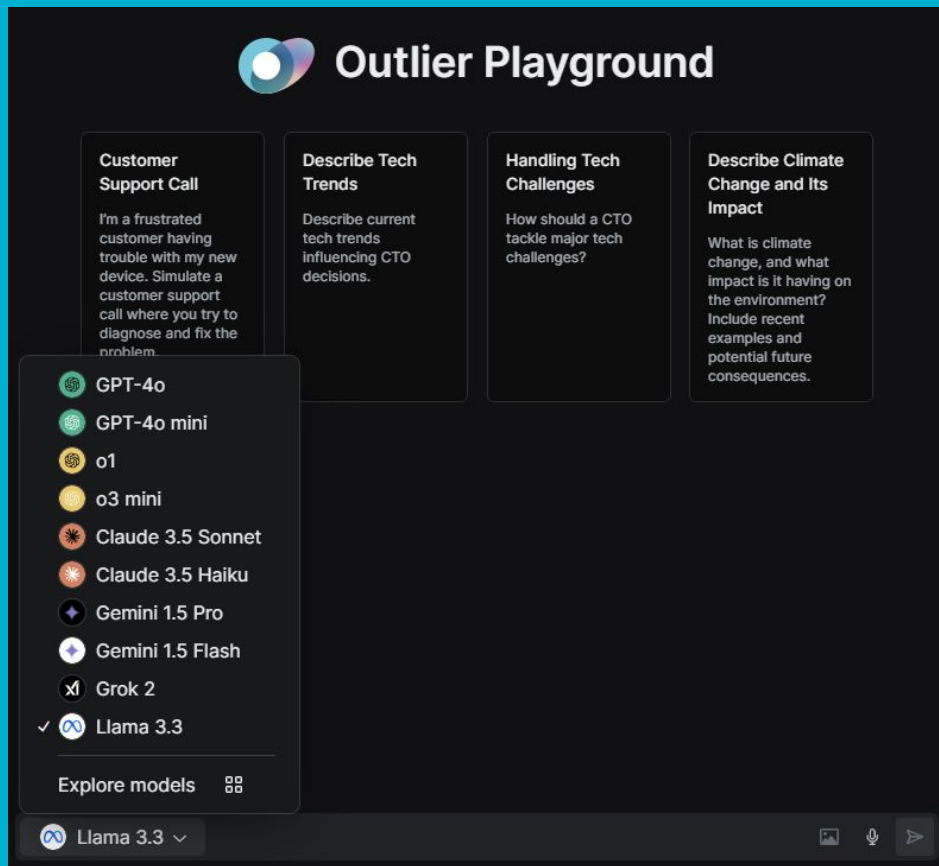
---

- 1 ~ 2년 사이 LLM(Large Language Model)이 급속도로 발전했음
- ChatGPT, Gemini 등 대부분의 LLM 서비스가 ‘준전문가’ 급의 답변을 하기 시작함
- 코딩도 만들어주기 시작함

프로그래밍 분야 뿐만 아니라 전문 지식이 필요한 영역에서 급속도로 AI가 활용되기 시작함

# LLM 서비스 이용 방법

- Windows Copilot 사용  
→ 가장 접근하기 쉬움
- ChatGPT 무료 서비스 활용
- Outlier.ai 인공지능 학습 일터를 통해  
무제한으로 LLM 서비스 사용 가능



# LLM 서비스 활용 방법

---

- 용어의 정확한 정의 확인
- 학습의 단계 확인
- 접근 방법에 대한 가이드
- 코딩을 한 후 좀 더 나은 방법을 확인

※ 단순히 답을 얻기 위한 목적으로 활용하지 말것. 머리를 안쓰게 됨

※ 올바르게 질문했는가, 답변이 정확한가 계속 확인하면서 사용해야 함

# LLM 서비스의 춘추전국시대

---

- 값싸고 훌륭한 LLM 서비스를 사용할 수 있는 시대가 열렸음
- 아이디어와 추진력이 있다면 누구나 돈을 벌 수 있는 기회의 시기

Open AI(Microsoft), Google, X, Facebook, ... 다양한 대기업에서 LLM 서비스 출시