다트 기초 개념을 복습하고 실습할 수 있는 프로젝트를 제안해드릴게요! 지금까지 배우신 내용(변수 타입, null 안전성, 컬렉션, 함수 파라미터, typedef 등)을 모두 활용할 수 있는 미니 프로젝트들을 몇 가지 소개해드립니다.

**1. 📝 할 일 관리 앱 (Todo Manager)**

**개념 적용**: 다양한 파라미터 타입, enum, Map, List, Set, typedef

void main() {

final TodoManager todoManager = TodoManager();

// 다양한 파라미터 타입으로 할 일 추가해보기

todoManager.addTask("다트 공부하기", priority: Priority.high);

todoManager.addTask("Flutter 강의 듣기");

todoManager.addTask("쇼핑하기", category: "개인");

// typedef를 활용한 필터링

todoManager.filterAndPrint((task) => task.priority == Priority.high);

// Map과 List 활용

todoManager.printTasksByCategory();

}

이 프로젝트에서는 할 일을 추가, 완료 표시, 삭제하고 카테고리별/우선순위별로 필터링하는 기능을 구현할 수 있어요.

**2. 🧮 미니 계산기 (Calculator)**

**개념 적용**: typedef, 함수형 프로그래밍, 다양한 파라미터, nullable

void main() {

final calculator = Calculator();

// 기본 연산들

print("덧셈 결과: ${calculator.calculate(10, 5, calculator.add)}");

print("뺄셈 결과: ${calculator.calculate(10, 5, calculator.subtract)}");

// 옵셔널 파라미터 활용

final result = calculator.performOperation(

firstNumber: 10,

secondNumber: 5,

operation: "multiply"

);

print("연산 결과: $result");

}

이 프로젝트에서는 typedef를 활용한 연산 함수를 정의하고, 다양한 파라미터를 통해 계산기 기능을 구현할 수 있어요.

**3. 📚 미니 도서관 관리 시스템 (Library)**

**개념 적용**: 클래스, List, Map, Set, enum, nullable, typedef

void main() {

final library = Library();

// 다양한 형태로 책 추가하기

library.addBook("해리 포터", "J.K. 롤링", category: BookCategory.fiction);

library.addBook("다트 프로그래밍", "코드 팩토리");

// Set을 활용한 중복 제거

library.addBook("해리 포터", "J.K. 롤링", category: BookCategory.fiction);

// Map과 List 활용해 책 찾기

library.findBooksByAuthor("J.K. 롤링");

// typedef를 활용한 검색 기능

library.searchBooks((book) => book.title.contains("다트"));

}

이 프로젝트는 도서 추가, 검색, 대여 상태 관리 등의 기능을 구현할 수 있어요.

**4. 🎮 간단한 텍스트 RPG 게임**

**개념 적용**: 클래스, enum, 함수형 프로그래밍, 다양한 파라미터, Map

void main() {

final game = TextRPG();

// 플레이어 생성 (네임드 파라미터 활용)

game.createPlayer(name: "용사", characterClass: CharacterClass.warrior);

// 게임 시작

game.startAdventure();

// typedef를 활용한 이벤트 처리

game.onPlayerAction((action) {

if (action == PlayerAction.attack) {

print("플레이어가 공격합니다!");

}

});

}

간단한 텍스트 기반 RPG 게임을 만들어서 다양한 상황에 따라 다른 함수를 실행하는 연습을 할 수 있어요.

**5. 📊 데이터 분석기 (Data Analyzer)**

**개념 적용**: Map, List, typedef, 화살표 함수, 옵셔널/네임드 파라미터

void main() {

final analyzer = DataAnalyzer();

// 샘플 데이터 추가

analyzer.addData("판매량", [120, 200, 150, 80, 210]);

analyzer.addData("방문자", [1500, 1350, 1800, 1950, 2000]);

// 다양한 분석 함수 적용 (typedef 활용)

analyzer.analyze("판매량", analyzer.average);

analyzer.analyze("방문자", analyzer.max);

// 커스텀 분석 함수 전달 (화살표 함수)

analyzer.analyze("판매량", (List<num> data) => data.reduce((a, b) => a + b));

}

다양한 데이터를 분석하는 기능을 구현하며 함수형 프로그래밍과 컬렉션을 활용할 수 있어요.

**추천 접근 방법**

1. **가장 끌리는 프로젝트를 선택**하세요. 흥미가 있는 주제일수록 학습 효과가 높아요.
2. **단계적으로 구현**하세요:
   * 기본 클래스 구조 먼저 설계
   * 핵심 기능 하나씩 추가
   * 동작 테스트를 통해 즉시 피드백 받기
3. **배운 개념을 의도적으로 모두 사용**해보세요:
   * null 안전성 (nullable/non-nullable)
   * 다양한 파라미터 유형 (포지셔널, 옵셔널, 네임드)
   * typedef와 함수형 프로그래밍
   * 컬렉션 (List, Map, Set)
   * final과 const의 적절한 사용

이런 미니 프로젝트를 통해 다트의 기초 개념을 확실히 다질 수 있을 거예요. 특히 같은 기능도 다양한 방식으로 구현해보면서 어떤 방식이 더 Dart스러운지 느껴보는 것이 좋습니다. 어떤 프로젝트가 가장 매력적으로 느껴지나요?