

GDSC KU Flutter Junior Session



Week4- 내부 저장소 이용하기

GDSC Mobile Core 이유림

tel. 010-8248-6757

목차

1. 내부저장소 개념
2. AddToDoDialog 를 stateful로 전환
3. 비동기 함수
4. sharedPreferences 사용법
5. month.dart 완성
6. 과제 및 공지



내부저장소 개념

내부 저장소

개별 어플리케이션에서 사용하는 전용 파일이다.

내부 앱 데이터를 저장하는데 사용된다.

사용자가 앱을 삭제하면 저장된 파일이 삭제된다.

그렇기에 데이터가 앱을 지워도 남게하려면 MediaStore API를 사용해서 적절한 저장소에 저장해야한다.

외부 저장소

모든 안드로이드 기기는 파일을 저장하는데 사용할 수 있는 외부 저장소를 가지고 있다.

파일에 접근하기 전에 외부 저장소 디렉토리를 사용할 수 있는지 확인해야한다.

다른 앱에서도 접근해야하고 앱 제거 뒤에도 사용해야하는 데이터라면 외부 저장소를 이용하는 것이 좋다.

공유 기본 설정 (SharedPreferences)

데이터를 많이 저장하지 않고, 데이터 구조가 필요하지 않는 경우 사용한다.

boolean, float, int, long, string 유형의 데이터를 키-값으로 쓰고 읽을 수 있다.

이것은 XML파일에 작성되고, 앱이 종료되더라도 유지된다.

파일 이름을 직접 지정하거나 액티비티별로 파일을 사용해 데이터를 저장할 수 있다.

출처 : <https://dev-gold.tistory.com/83>



AddToDoDialog 를 stateful로 전환

- widget 자료형으로 선언되어 있던 것을 stateful class로 바꿔줍니다.

```
class AddToDoDialog extends StatefulWidget {  
  const AddToDoDialog({Key? key, required this.add}) : super(key: key);  
  final Function add;
```

```
  @override  
  AddToDoDialogState createState() => AddToDoDialogState();  
}
```

```
class AddToDoDialogState extends State<AddToDoDialog> {  
  ... (생략) ...
```

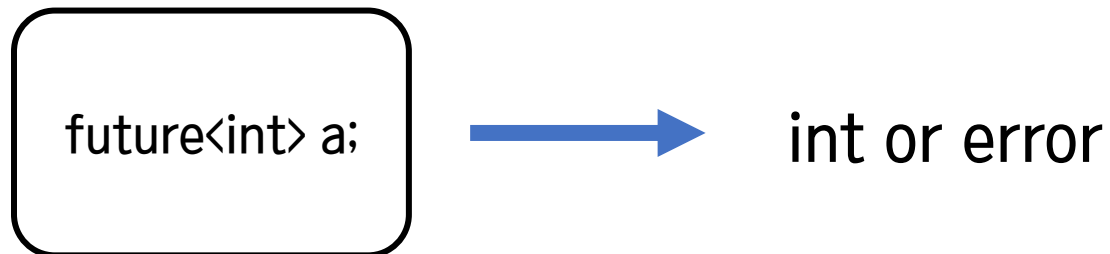


비동기 함수

future 자료형 -> 비동기를 나타내기 위함

Dart에서 Future는 자바스크립트의 Promise에 대응되며, Future와 Promise는 모두 싱글스레드 환경에서 비동기 처리를 위해 존재한다.

- JS의 Promise는 특정 동작의 수행이 완료되면, 다음 동작을 수행하겠다는 약속
- Dart의 Future는 지금은 없지만 미래에 요청한 데이터 혹은 에러가 담길 그릇



비동기 함수

```
import 'dart:async';

Future<int> futureNumber() {
  // 3초 후 100이 상자에서 나옵니다
  return Future<int>.delayed(Duration(seconds: 3), () {
    return 100;
  });
}

void main() {
  // future 라는 변수에서 미래에(3초 후에) int가 나올 것입니다
  Future<int> future = futureNumber();

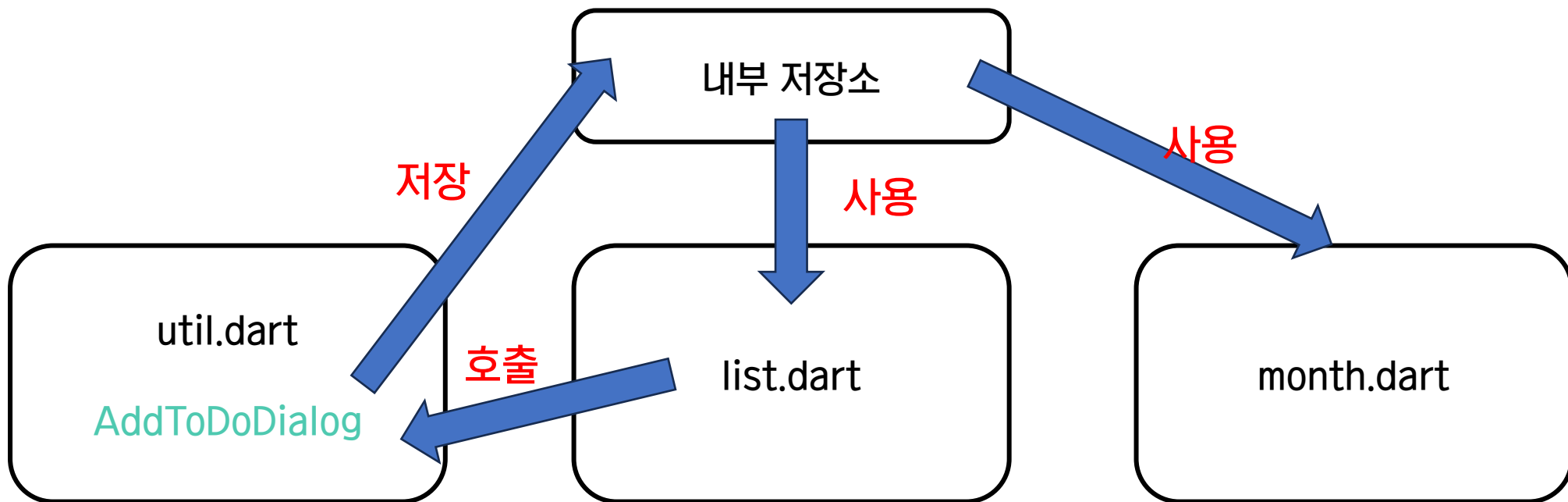
  future.then((val) {
    // int가 나오면 해당 값을 출력
    print('val: $val');
  }).catchError((error) {
    // error가 해당 에러를 출력
    print('error: $error');
  });

  print('기다리는 중');
}
```

기다리는 중
val: 100

sharedPreferences 사용법

- https://pub.dev/packages/shared_preferences



sharedPreferences 사용법 : list.dart

```
16 final Future<SharedPreferences> _prefs = SharedPreferences.getInstance();
17 List<String> listComponent = [];
18 Future<void> _loadData() async {
19     final SharedPreferences prefs = await _prefs;
20     setState(() {
21         listComponent = prefs.getStringList("listComponent") ?? [];
22     });
23 }
24
25 @override
26 void initState() {
27     super.initState();
28     _loadData();
29 }
30
```



sharedPreferences 사용법 : list.dart

```
return AddToDoDialog(add: (String title, String content,
    DateTime startDate, DateTime endDate) {
    setState(() {
        listComponent
            .add([title, content, startDate, endDate]);
    });
}); // AddToDoDialog
});
```

```
return AddToDoDialog(add: (String title, String content,
    String startDate, String endDate) async {
    final SharedPreferences prefs = await _prefs;
    setState(() {
        listComponent.add(
            jsonEncode([title, content, startDate, endDate]));
        prefs.setStringList("listComponent", listComponent);
    });
}); // AddToDoDialog
```



sharedPreferences 사용법 : list.dart

```
deleteButton(index, () async {  
  final SharedPreferences prefs = await _prefs;  
  setState(() {  
    listComponent.removeAt(index);  
    prefs.setStringList("listComponent", listComponent);  
  });  
}));
```

[https://github.com/yurimn/flutter-study/blob/main/to do list/lib/theme/basic/list 0.dart](https://github.com/yurimn/flutter-study/blob/main/to%20do%20list/lib/theme/basic/list_0.dart)



sharedPreferences 사용법 : month.dart

- list.dart 와 동일, 저장소를 가져오는 기능만 있으면 됨.

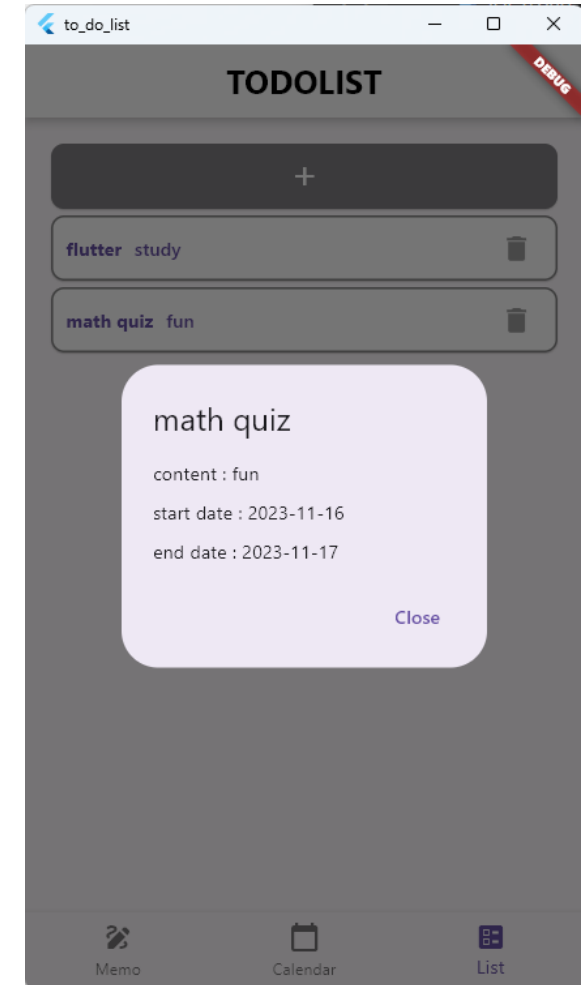
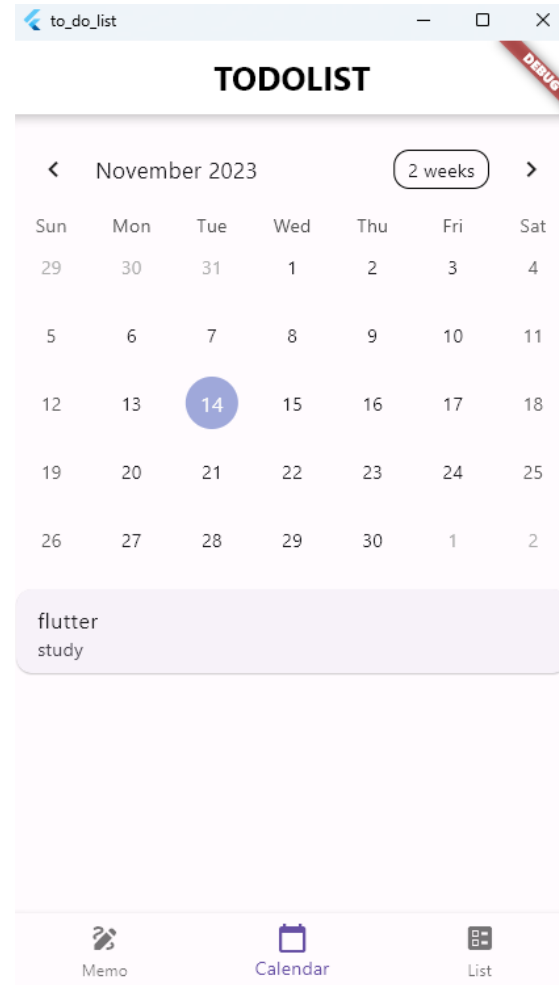
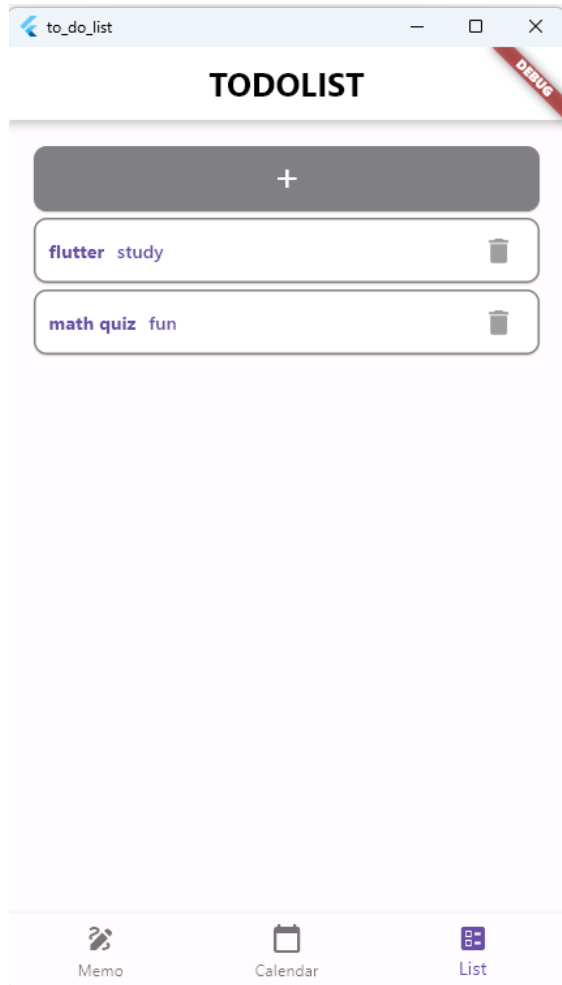


month.dart 완성

- https://pub.dev/packages/table_calendar
- https://github.com/yurimn/flutter-study/blob/main/to_do_list/lib/theme/basic/month_0.dart



예시 완성 화면 (basic theme)



과제

- 기능 & 디자인(글꼴, theme 색 포함) 완성시키기
- 이번주 토요일 (11/18) 23:59 까지 pull request를 받습니다.
- 해당 시간까지의 commit 내용으로 테마를 합쳐 앱을 배포할 예정입니다.
- 오류는 받지 않습니다... 모두 고쳐서 풀 해주세요...



공지

- 다음주는 배포한 앱을 다같이 다운받아보고, 사용한 기능을 발표, 피드백 하는 시간입니다.
- 플레이스토어 배포 방법/테스팅 에 관한 간단한 소개가 있을 예정입니다.





Developer Student Clubs