

Do it! 플러터 앱 프로그래밍

8장 내부 저장소 사용하기

목차

- 09-1 공유 환경설정에 데이터 저장하기
- 09-2 파일에 데이터 저장하기

09-1 공유 환경설정에 데이터 저장하기

실습하기

Pub.dev 에서 shared_preferences 검색하여 패키지 설치하기

Pubspec.yaml 파일에 추가하기

• pubspec.yaml

(...생략...)

dependencies:

flutter:

sdk: flutter

cupertino_icons: ^1.0.2

shared_preferences: 2.0.6

(...생략...)

09-1 공유 환경설정에 데이터 저장하기

실습하기

데이터를 저장하고, 불러오는 함수를 생성

SharedPreferences는 간단한 데이터를 저장할때 요긴하게 사용할 수 있음

```
// 데이터를 저장하는 함수
void _setData(int value) async {
    var key = 'count';
    SharedPreferences pref = await SharedPreferences.getInstance();
    pref.setInt(key, value);
}

// 데이터를 가져오는 함수
void _loadData() async {
    var key = 'count';
    SharedPreferences pref = await SharedPreferences.getInstance();
    setState(() {
        var value = pref.getInt(key);
        if (value == null) {
            _counter = 0;
        } else {
            _counter = value;
        }
    });
}
```

09-1 공유 환경설정에 데이터 저장하기

실습하기

앱을 실행할때 불러오기를 호출하여 데이터를 불러오고,

incrementCounter() 함수를 호출할 때마다 데이터가 저장되도록 수정하기

```
@override
void initState() {
    super.initState();
    _loadData();
}

void _incrementCounter() {
    setState(() {
        _counter++;
        _setData(_counter);
    });
}
```

09-1 공유 환경설정에 데이터 저장하기

실습하기

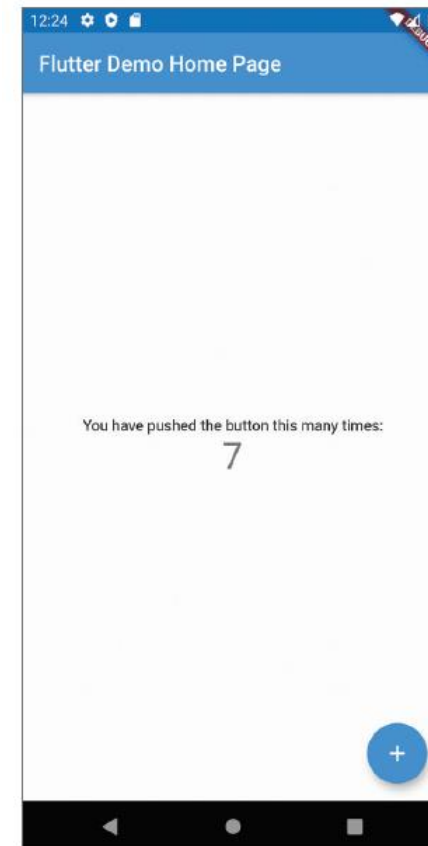
앱을 실행할때 불러오기를 호출하여 데이터를 불러오고,

incrementCounter() 함수를 호출할 때마다 데이터가 저장되도록 수정하기

```
@override
void initState() {
  super.initState();
  _loadData();
}

void _incrementCounter() {
  setState(() {
    _counter++;
    _setData(_counter);
  });
}
```

▶ 실행 결과



09-2 파일에 데이터 저장하기

실습하기

Path_provider 를 패키지를 찾아서 설치하기

Path_provider는 플러터에서 파일의 위치를 쉽게 알수있게 만드는 패키지입니다.

• pubspec.yaml

(...생략...)

dependencies:

flutter:

 sdk: flutter

cupertino_icons: ^1.0.2

path_provider: ^2.0.2

(...생략...)

09-2 파일에 데이터 저장하기

실습하기

Import를 통해서 패키지를 추가한 후

writeCountFile 파일을 쓸수있는 함수와

readCountFile 파일을 읽을수있는 함수를
만든 뒤에 코드를 작성합니다.



아하!

임시 디렉터리를 이용하는 방법

내부 저장소의 경로를 가져올 때 `getApplicationDocumentsDirectory()` 함수 외에 `getTemporaryDirectory()` 함수를 이용할 수도 있습니다. 이 함수는 임시 디렉터리 경로를 가져옵니다. 다만, 임시 디렉터리는 캐시(cache)를 이용하므로 앱이 종료되고 일정 시간이 지나면 사라질 수 있습니다.

```
import 'dart:io';  
  
import 'package:path_provider/path_provider.dart';  
  
void writeCountFile(int count) async {  
  var dir = await getApplicationDocumentsDirectory();  
  File(dir.path + '/count.txt').writeAsStringSync(count.toString());  
}  
  
void readCountFile() async {  
  try {  
    var dir = await getApplicationDocumentsDirectory();  
    var file = await File(dir.path + '/count.txt').readAsString();  
    print(file);  
    setState(() {  
      _count = int.parse(file);  
    });  
  } catch (e) {  
    print(e.toString());  
  }  
}
```


09-2 파일에 데이터 저장하기

실습하기

파일을 입력하고 출력할 수 있도록

화면을 구성한 후 빌드

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('File Example'),
    ),
    body: Container(
      child: Center(
        child: Text(
          '$_count',
          style: TextStyle(fontSize: 40),
        ),
      ),
    ),
    floatingActionButton: FloatingActionButton(
      onPressed: () {
        setState(() {
          _count++;
        });
        writeCountFile(_count);
      },
      child: Icon(Icons.add),
    ),
  );
}
```

▶ 실행 결과

