

# 앱 개발 입문

성균관대학교 컬처애편테크놀로지융합전공 하계 부트캠프

3일차 - 2022. 07. 29 (금)

## 강의 슬라이드 링크

[https://github.com/kunny/skku-bootcamp-2022-summer/blob/main/\\_slides/day3.pdf](https://github.com/kunny/skku-bootcamp-2022-summer/blob/main/_slides/day3.pdf)

# 오늘 강의에서 다룰 내용

- 노트 앱 프로젝트를 생성하고, 앱 테마 색상을 변경합니다.
- 노트 데이터를 저장할 수 있는 클래스를 만듭니다.
- 노트 목록을 그리드 형태로 표시하고, 샘플 노트를 넣어 결과를 확인합니다.
- 노트에 색상 정보를 추가하고, 이를 목록에 표시합니다.
- 노트 보기 옵션을 전환하는 기능을 추가합니다. (그리드 / 리스트)

# 노트 앱 프로젝트 만들기

- Project name: [sticky\\_notes](#)
- Organization: [edu.skku.sco](#). [본인 닉네임 혹은 ID]
  - 예) 닉네임이 androidhuman인 경우: [edu.skku.sco.androidhuman](#)

- pubspec.yaml 내 주석 삭제
- test 폴더 제거

## 노트 목록 화면 뼈대 작성

1. `lib` 폴더 아래에 `page` 폴더를 생성합니다.
2. `page` 폴더에 `note_list_page.dart` 파일을 생성합니다.
3. 노트 목록을 표시할 위젯인 `NoteListPage` 코드를 작성합니다.

# 노트 목록 화면 뼈대 작성

```
import 'package:flutter/material.dart';

class NoteListPage extends StatefulWidget {
  const NoteListPage({Key? key}) : super(key: key);

  @override
  State createState() => _NoteListPageState();
}

class _NoteListPageState extends State<NoteListPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Sticky Notes'),
      ),
    );
  }
}
```

note\_list\_page.dart



## 앱 첫 화면 변경 및 앱 테마 색상 변경

1. **MyApp** 클래스의 **home** 속성에 `NoteListPage`를 지정하여 첫 화면으로 노트 목록 페이지를 표시합니다.
2. **MyApp** 클래스의 **theme** 속성에 할당되어있는 **`ThemeData.primarySwatch`** 속성 변경하여 앱 테마 색상을 원하는 색상으로 변경합니다.

# 앱 첫 화면 변경 및 앱 테마 색상 변경

```
import 'package:flutter/material.dart';
import 'package:sticky_notes/page/note_list_page.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Sticky Notes',
      theme: ThemeData(
        primarySwatch: Colors.teal,
      ),
      home: const NoteListPage(),
    );
  }
}
```

## 완성된 코드

[https://github.com/kunny/skku-bootcamp-2022-summer/tree/main/sticky\\_notes/step1](https://github.com/kunny/skku-bootcamp-2022-summer/tree/main/sticky_notes/step1)

노트 데이터를 저장할 수 있는  
클래스 만들기

## 노트 데이터를 저장할 수 있는 클래스 만들기

1. `lib` 폴더 아래에 `data` 폴더를 생성합니다.
2. `data` 폴더에 `note.dart` 파일을 생성합니다.
3. 노트에 저장할 데이터를 정의한 후, `Note` 클래스를 작성합니다.

# 노트 데이터를 저장할 수 있는 클래스 만들기

```
import 'package:flutter/material.dart';

class Note {
  final String title;

  final String body;

  Note(
    this.body, {
    this.title = '',
  });
}
```

# 노트 데이터를 저장할 수 있는 클래스 만들기

```
Note(  
    this.body, {  
        this.title = '',  
    });
```

생성자: 클래스의 새로운 인스턴스를 만들때  
사용합니다.

```
// body = '샘플 노트입니다', title = '샘플 노트'  
Note('샘플 노트입니다', title: '샘플 노트');
```

```
// body = '제목 없는 노트', title = ''  
Note('제목 없는 노트');
```

노트 목록을  
그리드 형태로 표시하기



## 노트 목록을 그리드 형태로 표시하기

1. `_NoteListPageState` 클래스에 샘플 노트 데이터를 추가합니다.
2. 각 노트를 화면에 그려주는 함수인 `_buildCard()` 함수를 구현합니다.
3. 노트 목록을 그리드 형태로 그려주는 함수인 `_buildCards()` 함수를 구현합니다.

# 노트 목록을 그리드 형태로 표시하기

## 샘플 노트 코드

```
class _NoteListPageState extends State<NoteListPage> {  
  final _notes = [  
    ... // 샘플 노트를 붙여넣습니다.  
  ];  
  
  @override  
  Widget build(BuildContext context) {  
    ...  
  }  
}
```

note\_list\_page.dart

# 노트 목록을 그리드 형태로 표시하기

```
Widget _buildCard(int index, Note note) {  
  return Card(  
    child: Padding(  
      padding: const EdgeInsets.all(12.0),  
      child: Column(  
        mainAxisAlignment: MainAxisAlignment.min,  
        crossAxisAlignment: CrossAxisAlignment.start,  
        children: [  
          Text(  
            note.title.isEmpty ? '(제목 없음)' : note.title,  
            style: const TextStyle(  
              fontSize: 18.0,  
              fontWeight: FontWeight.bold,  
            ),  
          ),  
          const SizedBox(height: 16.0),  
          Expanded(  
            child: Text(  
              note.body,  
              overflow: TextOverflow.fade,  
            ),  
          ),  
        ],  
      ),  
    ),  
  );  
}
```

## 노트 목록을 그리드 형태로 표시하기

```
Widget _buildCards(List<Note> notes) {  
  return GridView.builder(  
    padding:  
    const EdgeInsets.symmetric(horizontal: 12.0, vertical: 16.0),  
    itemCount: notes.length,  
    itemBuilder: (context, index) => _buildCard(index, notes[index]),  
    gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(  
      crossAxisCount: 2,  
      childAspectRatio: 1,  
    ),  
  );  
}
```

# 노트 목록을 그리드 형태로 표시하기

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Sticky Notes'),
    ),
    body: _buildCards(_notes),
  );
}
```

## Sticky Notes

**About Flutter**

Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows, Google

**(제목 없음)**

부트캠프 앱 개발 강의 신청하기

**(제목 없음)**

목요일까지 관리비 납부하기

**장보기 목록**

- 🍎 양파 1망
- 🥬 양배추 1통
- 🍊 귤 1박스
- 🍖 손질 닭고기 1팩
- 🍖 삼겹살 1팩
- 🍌 오트밀 4팩 (4인)

**만들어야 할 것들**

- 파이어베이스 프로젝트
- 애드몹 계정
- 구글 플레이 개발자 계정
- 애플 개발자 계정

**(제목 없음)**

목요일 오후 7시 강남역 11번출구

**(제목 없음)**

플러터 위젯 공부하기 - CircleAvatar

노트에 색상 추가하기

## 노트에 색상 추가하기

1. **Note** 클래스에 색상을 저장할 수 있는 필드를 추가합니다.
2. 샘플 노트에 색상을 추가합니다.
3. 노트 목록에서 색상 정보를 화면에 그려주도록 코드를 수정합니다.



# 노트에 색상 추가하기

```
class Note {  
  static const colorDefault = Colors.white;  
  
  static const colorRed = Color(0xFFFFCDD2);  
  
  static const colorOrange = Color(0xFFFFE0B2);  
  
  static const colorYellow = Color(0xFFFFF9C4);  
  
  static const colorLime = Color(0xFFFF0F4C3);  
  
  static const colorBlue = Color(0xFFBBDEFB);  
  
  final String title;  
  
  final String body;  
  
  final Color color;  
  
  Note(  
    this.body, {  
    this.title = '',  
    this.color = colorDefault,  
  });  
}
```

note.dart

# 노트 데이터를 저장할 수 있는 클래스 만들기

```
Color(0xFFFCDD2);
```

0x	FF	FF	CD	D2
Hexadecimal	Alpha (255)	Red (255)	Green (205)	Blue (210)

# 노트에 색상 추가하기

```
final _notes = [
  ...
  Note(
    '부트캠프 앱 개발 강의 신청하기',
    color: Note.colorRed,
  ),
  Note(
    '목요일까지 관리비 납부하기',
    color: Note.colorOrange,
  ),
  Note(
    '''
- 🍄 양파 1망
- 🥬 양배추 1통
- 🍊 귤 1박스
- 🐔 손질 닭고기 1팩
- 🥓 삼겹살 1팩
- 🍜 우동면 1팩 (4입)
- 🥛 우유 2팩
- 🍞 식빵 1개
'''
    ,
    title: '장보기 목록',
    color: Note.colorYellow,
  ),
  ...
]
```

## 노트에 색상 추가하기

```
Widget _buildCard(int index, Note note) {  
  return Card(  
    color: note.color,  
    child: Padding(  
      ...  
    ),  
  );  
}
```

## Sticky Notes

## About Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows, Google

## (제목 없음)

부트캠프 앱 개발 강의 신청하기

## (제목 없음)

목요일까지 관리비 납부하기

## 장보기 목록

- 🍎 양파 1망
- 🥬 양배추 1통
- 🍊 귤 1박스
- 🐔 손질 닭고기 1팩
- 🥩 삼겹살 1팩
- 🥬 오뚜기 4팩 2400

## 만들어야 할 것들

- 파이어베이스 프로젝트
- 애드몹 계정
- 구글 플레이 개발자 계정
- 애플 개발자 계정

## (제목 없음)

목요일 오후 7시 강남역 11번출구

## (제목 없음)

플러터 위젯 공부하기 - CircleAvatar

노트 보기 옵션 전환 기능 추가하기

## 노트 보기 옵션 전환 기능 추가하기

1. `_NoteListPageState` 클래스에 보기 옵션을 결정하는 필드를 추가합니다.  
(그리드/리스트)
2. `_buildCards()` 함수를 수정하여 보기 옵션에 따라 그리드 혹은 리스트 형태로 노트를 표시합니다.
3. `AppBar`의 액션 버튼에 보기 옵션을 전환하는 아이콘 버튼을 추가합니다.

## 노트 보기 옵션 전환 기능 추가하기

```
class _NoteListPageState extends State<NoteListPage> {  
  final _notes = [ ... ];  
  
  bool _showAsGrid = true;  
  
  @override  
  Widget build(BuildContext context) {  
    ...  
  }  
  ...  
}
```

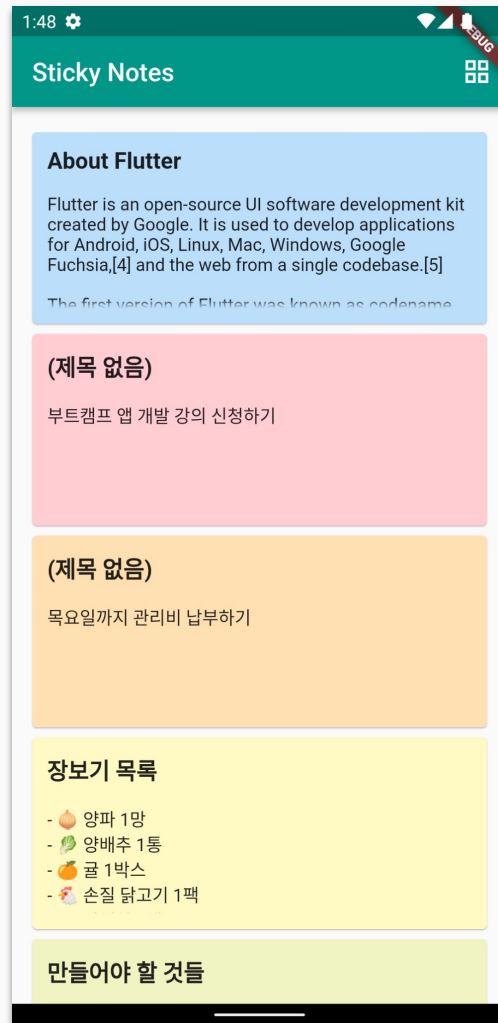


# 노트 보기 옵션 전환 기능 추가하기

```
Widget _buildCards(List<Note> notes) {  
  return _showAsGrid  
    ? GridView.builder(  
      padding:  
        const EdgeInsets.symmetric(horizontal: 12.0, vertical: 16.0),  
      itemCount: notes.length,  
      itemBuilder: (context, index) => _buildCard(index, notes[index]),  
      gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(  
        crossAxisCount: 2,  
        childAspectRatio: 1,  
      ),  
    )  
    : ListView.builder(  
      padding:  
        const EdgeInsets.symmetric(horizontal: 12.0, vertical: 16.0),  
      itemCount: notes.length,  
      itemBuilder: (context, index) => SizedBox(  
        height: 160,  
        child: _buildCard(index, notes[index]),  
      ),  
    );  
}
```

# 노트 보기 옵션 전환 기능 추가하기

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Sticky Notes'),
      actions: [
        IconButton(
          icon: Icon(_showAsGrid ? Icons.list : Icons.grid_view),
          tooltip: _showAsGrid ? '목록으로 보기' : '격자로 보기',
          onPressed: () {
            setState(() {
              _showAsGrid = !_showAsGrid;
            });
          },
        ),
      ],
    ),
    body: _buildCards(_notes),
  );
}
```



완성된 코드

[https://github.com/kunny/skku-bootcamp-2022-summer/tree/main/sticky\\_notes/step2](https://github.com/kunny/skku-bootcamp-2022-summer/tree/main/sticky_notes/step2)