

앱 개발 입문

성균관대학교 컬처애펀테크놀로지융합전공 하계 부트캠프

2일차 - 2022. 07. 28 (목)

강의 슬라이드 링크

https://github.com/kunny/skku-bootcamp-2022-summer/blob/main/_slides/day2.pdf

오늘 강의에서 다룰 내용

- 카운터 앱 동작을 바꿔봅니다.
- 머티리얼 디자인을 알아봅니다.
- 위젯 배치를 결정하는 컨테이너 위젯을 살펴봅니다.
- 머티리얼 디자인 컴포넌트로 화면을 구성해봅니다.
- 노트 앱 프로젝트를 생성합니다.

앱 동작 변경해보기

앱 동작 변경해보기

버튼을 눌렀을 때 숫자가 2씩 증가하게끔 `_incrementCounter()` 함수를 변경해봅니다.

```
void _incrementCounter() {  
  setState(() {  
    _counter+=2;  
  });  
}
```

코드 변경 후 저장 버튼을 누르면, 변경된 동작이 바로 앱에 반영되므로 (Hot Reload) 앱을 재시작하지 않아도 됩니다.

앱 동작 변경해보기 (실습)

- 버튼을 눌렀을 때 숫자가 5배씩 증가하게끔 `_incrementCounter()` 함수를 변경해봅니다.
- 주석을 사용하여 수정한 부분에 대한 추가 설명을 넣어줍니다.

힌트

- 곱하기 연산은 `*` (별표)로 할 수 있습니다.
- $0 * n = 0$

머티리얼 디자인 알아보기



머티리얼 디자인 알아보기

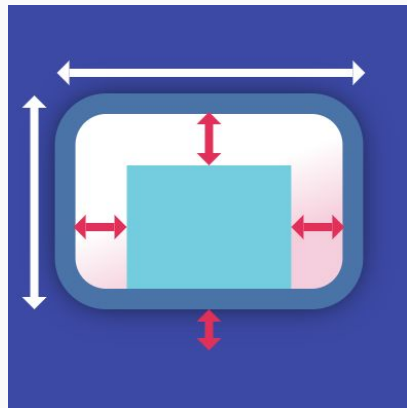
- 구글에서 제공하는 디자인 프레임워크로, 다양한 플랫폼 (휴대폰, 태블릿, PC 등)에서 최소한의 수고로도 미려한 디자인의 앱을 제작할 수 있습니다.
- 다양한 플랫폼에서 사용할 수 있는 머티리얼 디자인 컴포넌트를 제공합니다.
- 머티리얼 디자인에서 사용할 수 있는 [아이콘](#) 및 [색상](#)을 지원합니다.
- <https://material.io/>

컨테이너 위젯 알아보기



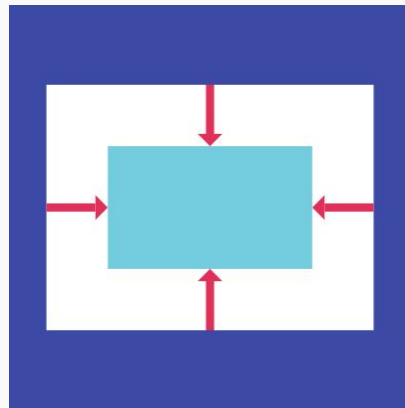
Container

- 위젯을 배치할 수 있는 다양한 방법을 지원합니다.
- 주요 속성
 - child: 자식 위젯을 지정합니다.
 - width: 컨테이너 너비를 지정합니다.
 - height: 컨테이너 높이를 지정합니다.
 - color: 컨테이너 배경 색상을 설정합니다.
 - padding: 컨테이너 내부 여백을 설정합니다.
 - margin: 컨테이너 외부 여백을 설정합니다.
 - alignment: 자식 위젯의 정렬 방법을 결정합니다.



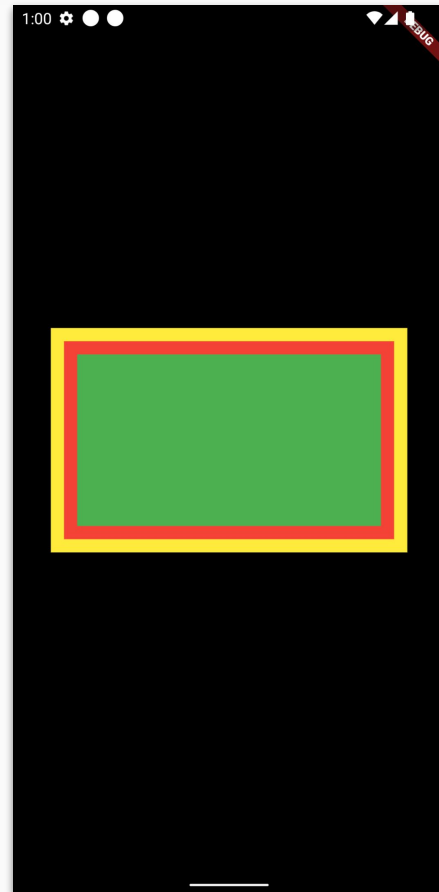
Center

- 자식 위젯을 컨테이너의 정가운데에 배치합니다.
- 주요 속성
 - child: 자식 위젯을 지정합니다.



Demo

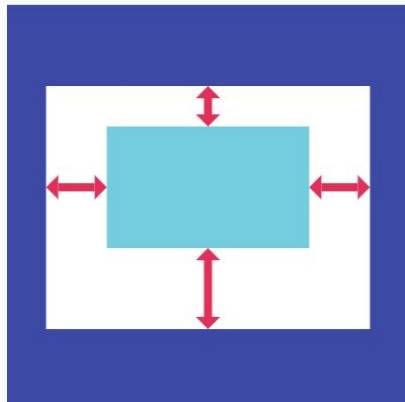
```
class ContainerDemo extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Center(  
      child: Container(  
        color: Colors.yellow,  
        child: Container(  
          color: Colors.red,  
          width: 300,  
          height: 180,  
          margin: const EdgeInsets.all(12.0),  
          padding: const EdgeInsets.all(12.0),  
          child: Container(color: Colors.green),  
        ),  
      ),  
    );  
  }  
}
```





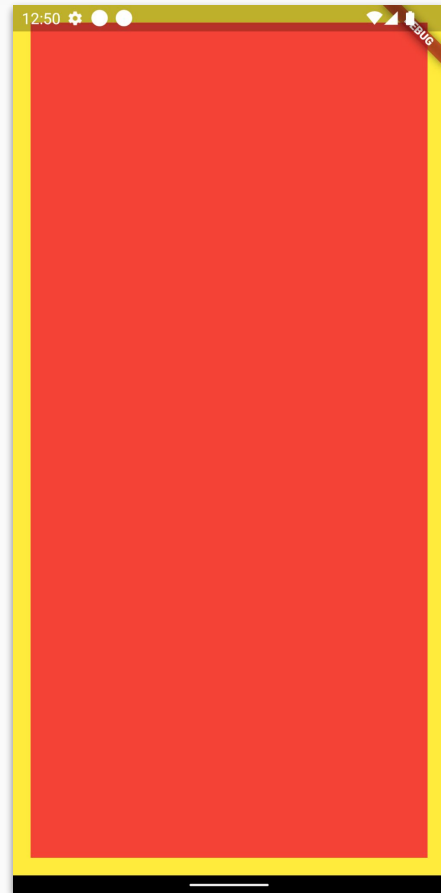
Padding

- 컨테이너 위젯 내부에 여백을 추가합니다.
- 주요 속성
 - child: 자식 위젯을 지정합니다.
 - padding: 여백을 넣어줄 위치 및 값을 지정합니다.



Demo

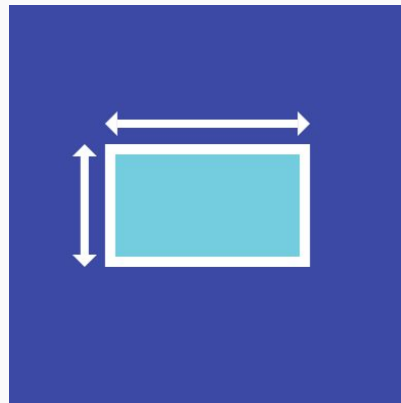
```
class PaddingDemo extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Container(  
      color: Colors.yellow,  
      child: Padding(  
        padding: const EdgeInsets.all(16.0),  
        child: Container(color: Colors.red),  
      ),  
    );  
  }  
}
```





SizedBox

- 너비와 높이를 지정할 수 있는 컨테이너입니다.
- 주요 속성
 - child: 자식 위젯을 지정합니다.
 - width: 컨테이너 너비를 지정합니다.
 - height: 컨테이너 높이를 지정합니다.

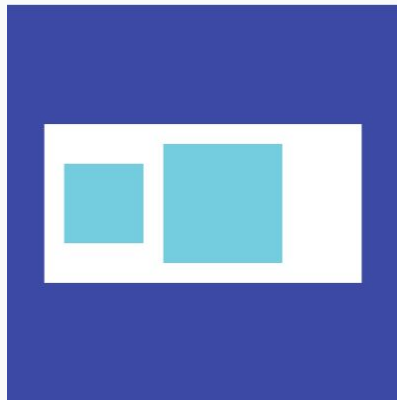


Column

- 자식 위젯들을 수직 방향으로 배치할 수 있는 컨테이너입니다.
- 주요 속성
 - children: 자식 위젯을 지정합니다.
 - mainAxisAlignment: 수직 방향 정렬 방법을 결정합니다.
 - mainAxisSize: 수직 방향으로 공간을 얼마나 차지할지 결정합니다.
 - crossAxisAlignment: 자식 위젯들을 수평 방향으로 어떻게 정렬할지 결정합니다.



- 자식 위젯들을 수평 방향으로 배치할 수 있는 컨테이너입니다.
- 주요 속성
 - children: 자식 위젯을 지정합니다.
 - mainAxisAlignment: 수평 방향 정렬 방법을 결정합니다.
 - mainAxisSize: 수평 방향으로 공간을 얼마나 차지할지 결정합니다.
 - crossAxisAlignment: 자식 위젯들을 수직 방향으로 어떻게 정렬할지 결정합니다.



Expanded

- Column/Row 위젯 내부의 남은 공간을 모두 차지해야 할 때 사용합니다.
- 주요 속성
 - child: 자식 위젯을 지정합니다.

SingleChildScrollView

- 위젯이 화면 크기보다 더 많은 공간을 차지하는 경우, 스크롤을 통해 나머지 부분을 볼 수 있게 해 줍니다.
- 주요 속성
 - child: 자식 위젯을 지정합니다.
 - scrollDirection: 스크롤 방향을 지정합니다.

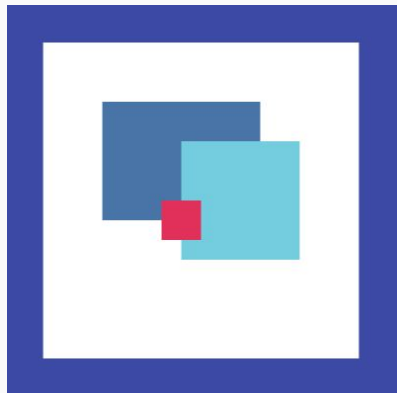
Demo

```
class ColumnDemo extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return SingleChildScrollView(  
      child: SizedBox(  
        height: 2000,  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.min,  
          children: [  
            Container(height: 300, color: Colors.yellow),  
            const SizedBox(height: 24),  
            Expanded(  
              child: Container(color: Colors.red),  
            ),  
          ],  
        ),  
      ),  
    );  
  }  
}
```



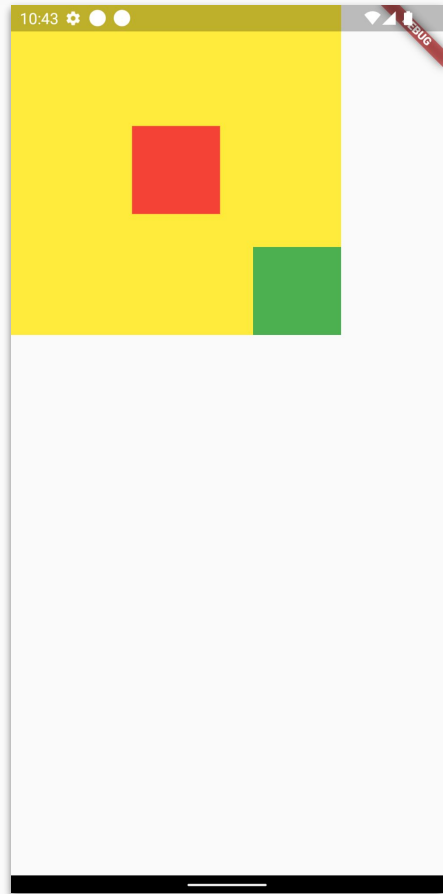


- 자식 위젯들을 포개어 배치할 수 있는 컨테이너입니다. Align 위젯을 함께 사용하면 자식 위젯을 다양한 방법으로 배치할 수 있습니다.
- 주요 속성
 - children: 자식 위젯을 지정합니다.
 - alignment: 별도로 정렬 방법이 정의되지 않은 자식 위젯을 어떻게 정렬할지 결정합니다.



Demo

```
class StackDemo extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: Container(  
        width: 300,  
        height: 300,  
        color: Colors.yellow,  
        child: Stack(  
          children: [  
            Align(  
              alignment: Alignment.center,  
              child: Container(  
                width: 80,  
                height: 80,  
                color: Colors.red,  
              ),  
            ),  
            Align(  
              alignment: Alignment.bottomRight,  
              child: Container(  
                width: 80,  
                height: 80,  
                color: Colors.green,  
              ),  
            ),  
          ],  
        ),  
      ),  
    );  
  }  
}
```



머티리얼 디자인 컴포넌트로 화면 구성하기

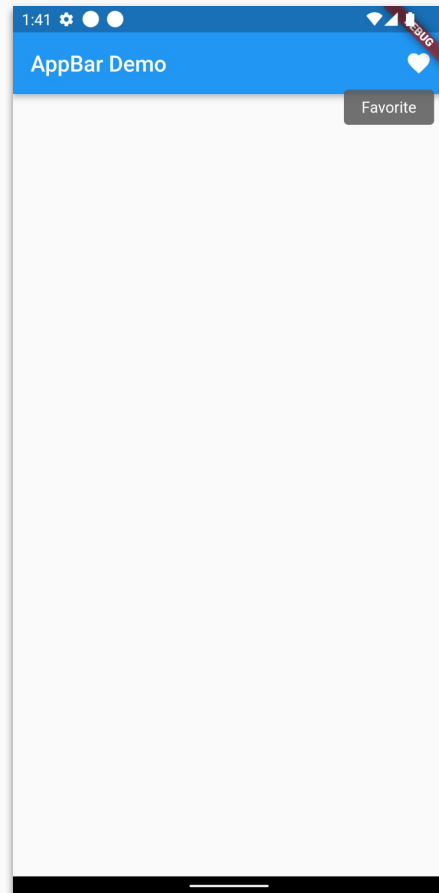
Flutter Gallery
(<https://gallery.flutter.dev>)

직접 실습해볼 위젯들

1. AppBar, IconButton, CircularProgressIndicator
2. Buttons (Text, Elevated, Outlined, FloatingActionButton)
3. ListTile, CircleAvatar
4. TextField

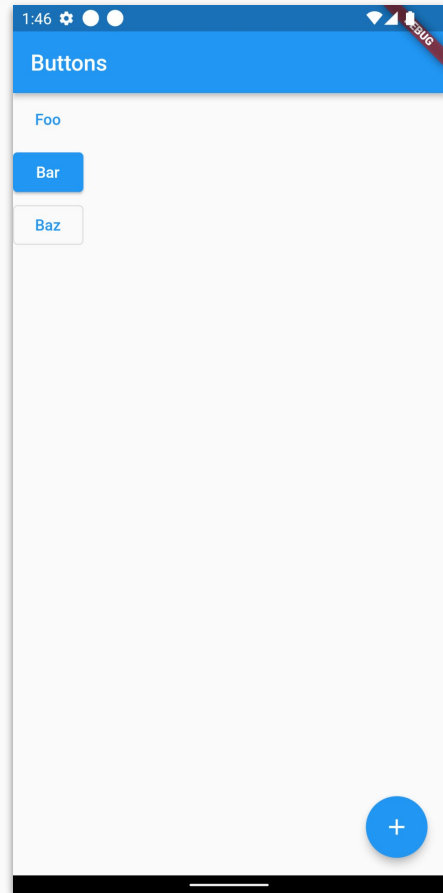
AppBar, IconButton, CircularProgressIndicator

```
class AppBarDemo extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('AppBar Demo'),  
        actions: [  
          IconButton(  
            icon: const Icon(Icons.favorite),  
            tooltip: 'Favorite',  
            onPressed: () {  
              print('Clicked Favorite');  
            },  
          ),  
        ],  
      ),  
    );  
  }  
}
```



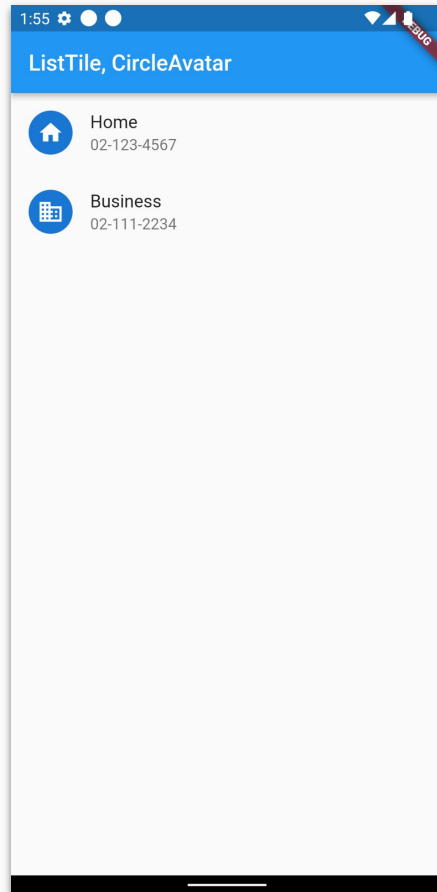
Buttons (Text, Elevated, Outlined, FloatingActionButton)

```
class ButtonDemo extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Buttons'),  
      ),  
      body: Column(  
        children: [  
          TextButton(  
            onPressed: () {},  
            child: const Text('Foo'),  
          ),  
          ElevatedButton(  
            onPressed: () {},  
            child: const Text('Bar'),  
          ),  
          OutlinedButton(  
            onPressed: () {},  
            child: const Text('Baz'),  
          ),  
        ],  
      ),  
      floatingActionButton: FloatingActionButton(  
        onPressed: () {},  
        tooltip: 'Add',  
        child: const Icon(Icons.add),  
      ),  
    );  
  }  
}
```



ListTile, CircleAvatar

```
class ListTileDemo extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('ListTile, CircleAvatar'),  
      ),  
      body: Column(  
        children: [  
          ListTile(  
            leading: const CircleAvatar(  
              child: Icon(Icons.home),  
            ),  
            title: const Text('Home'),  
            subtitle: const Text('02-123-4567'),  
            onTap: () {},  
          ),  
          ListTile(  
            leading: const CircleAvatar(  
              child: Icon(Icons.business),  
            ),  
            title: const Text('Business'),  
            subtitle: const Text('02-111-2234'),  
            onTap: () {},  
          ),  
        ],  
      ),  
    );  
  }  
}
```



TextField

```
class TextFieldDemo extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Text fields'),  
      ),  
      body: Padding(  
        padding: const EdgeInsets.all(16.0),  
        child: Column(  
          children: const [  
            TextField(  
              decoration: InputDecoration(  
                icon: Icon(Icons.person),  
                labelText: 'Name',  
                filled: true,  
              ),  
            ),  
            SizedBox(height: 24.0),  
            TextField(  
              decoration: InputDecoration(  
                icon: Icon(Icons.place),  
                labelText: 'Address',  
                filled: true,  
              ),  
            ),  
          ],  
        ),  
      ),  
    );  
  }  
}
```

