

# week3

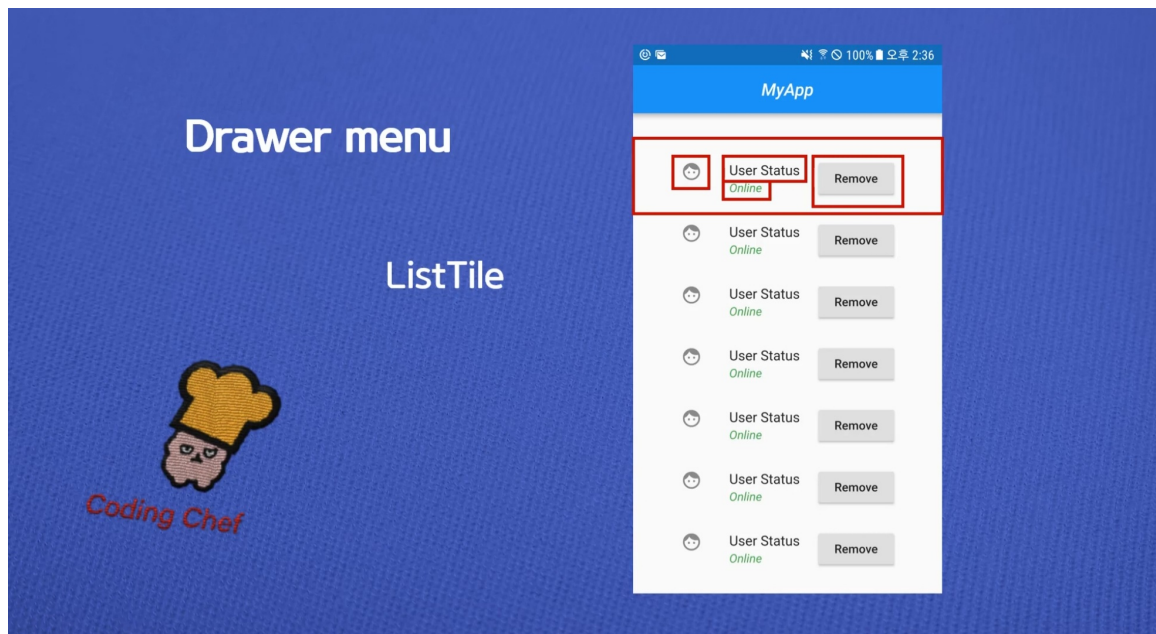
## 15강.

지난 시간 코드에서 Drawer 메뉴를 추가하면 자동으로 앱바 왼쪽 시작부분에 햄버거 메뉴가 추가되어, menus is clicked~ 이 부분의 ledaing~iconbutton 삭제해주고 시작함.

먼저 ctrl 키 누르고 scaffold 위젯 클릭하기 → drawer argument 보임 → 생성자에서 만들어진 named.argument라는걸 이해할 수 있을거임. drawer argument가 scaffold 위젯에 속해있는 속성임. → 앱바 위젯 밖으로 나와서 named argument 형식으로 drawer 속성 불러옴

### Drawer menu

- **ListTile**: list라 하면, 한줄만 보여주는 것이 아닌 세로 방향으로 여러줄로 된 정보를 보여줌. 복수의 위젯을 나열하는 위젯 리스트가 children 속성을 통해 호출될 것. 또한, 이 리스트의 한줄한줄을 ListTile이라고 함. 각각의 리스트 타일 내에서는 각 요소들의 배치가 필요함.



- 이번에도 assets란 폴더 만들어서 이미지 저장해줌. 두개의 이미지 정도. → pubspec.yaml로 가서 ctrl + /로 이미지 저장해주기

```

import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Appbar',
      theme: ThemeData(primarySwatch: Colors.red),
      home: MyPage(),
    );
  }
}

class MyPage extends StatelessWidget {
  const MyPage({Key? key}) : super(key: key);

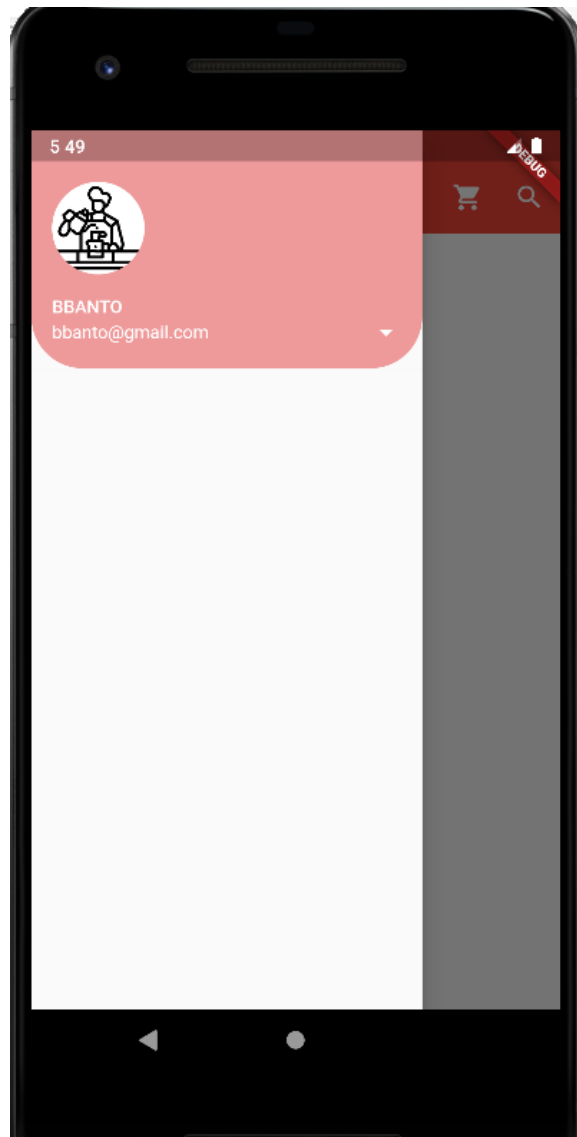
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Appbar icon menu'),
        centerTitle: true,
        elevation: 0.0,
        actions: <Widget>[
          IconButton(
            icon: Icon(Icons.shopping_cart),
            onPressed: () {
              print('Shopping cart button is clicked');
            }, // 기본적인 icon 메뉴만 만들어서
          ),
          IconButton(
            icon: Icon(Icons.search),
            onPressed: () {
              print('search button is clicked');
            },
          ),
        ],
      ),
      drawer: Drawer(
        // drawer 위젯 불러옴,
        child: ListView(
          // 여러 줄의 정보를 보여주는 목록이기에 child가 아닌 children 속성
          padding: EdgeInsets.zero,
          children: <Widget>[
            UserAccountsDrawerHeader(
              currentAccountPicture: CircleAvatar(
                backgroundImage: AssetImage('assets/barista.png'),
                backgroundColor: Colors.white,

```

```

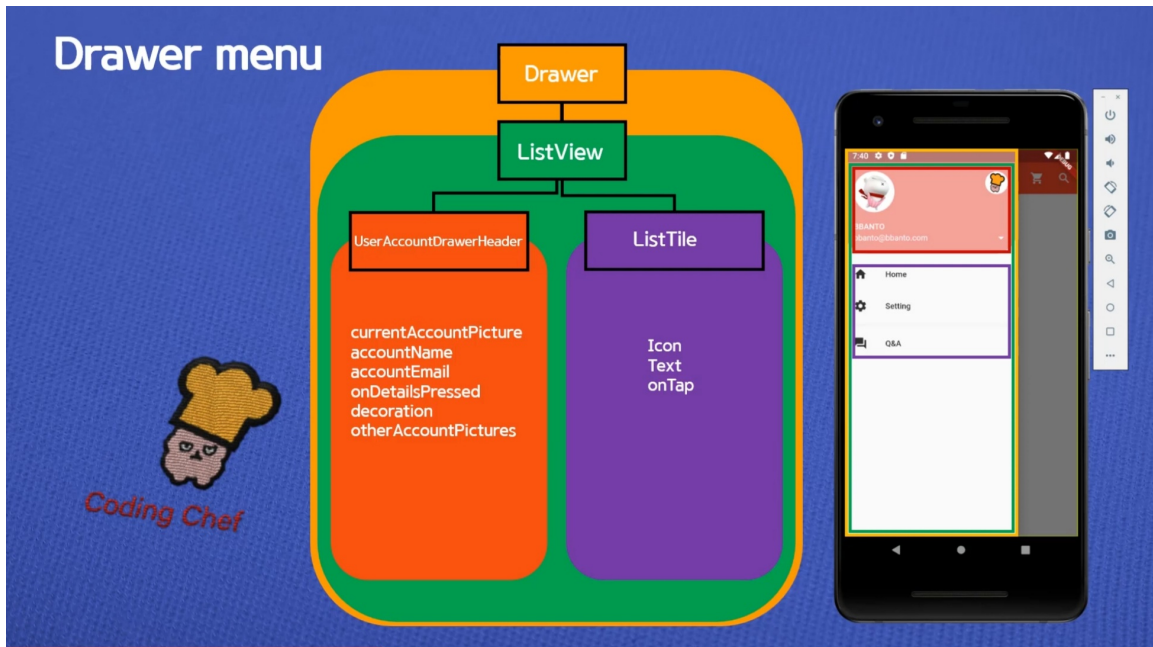
    ),
    accountName: Text('BBANTO'),
    accountEmail: Text('bbanto@gmail.com'),
    onDetailsPressed: () {
      print('arrow is clicked');
    },
    decoration: BoxDecoration(
      color: Colors.red[200],
      borderRadius: BorderRadius.only(
        bottomLeft: Radius.circular(40.0),
        bottomRight: Radius.circular(40.0))),
  ),
],
),
),
);
}
}

```



## 16강.

아까는 좌측 상단에 사용자 이미지를 불러왔다면, 이번에는 우측 상단에 다른 계정을 추가해보려고 함.



on Tap vs. onpressed? 기능적으로는 거의 동일함. onpressed 함수는 주로 버튼에, on Tap은 GestureDetector나 InkWell에 사용됨. 가령 길게 누르기, 두 번 탭하기 등 어떤 동작에 반응하는 이벤트를 위해 사용된다고 보면 됨.

ListTile에서 onTap 함수가 사용되는 이유는 일반 버튼과는 달리, 길게 탭하거나 액션을 감지하는 기능을 갖고 있기 때문임.

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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Appbar',
      theme: ThemeData(primarySwatch: Colors.red),
      home: MyPage(),
    );
  }
}

class MyPage extends StatelessWidget {
  const MyPage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
```

```

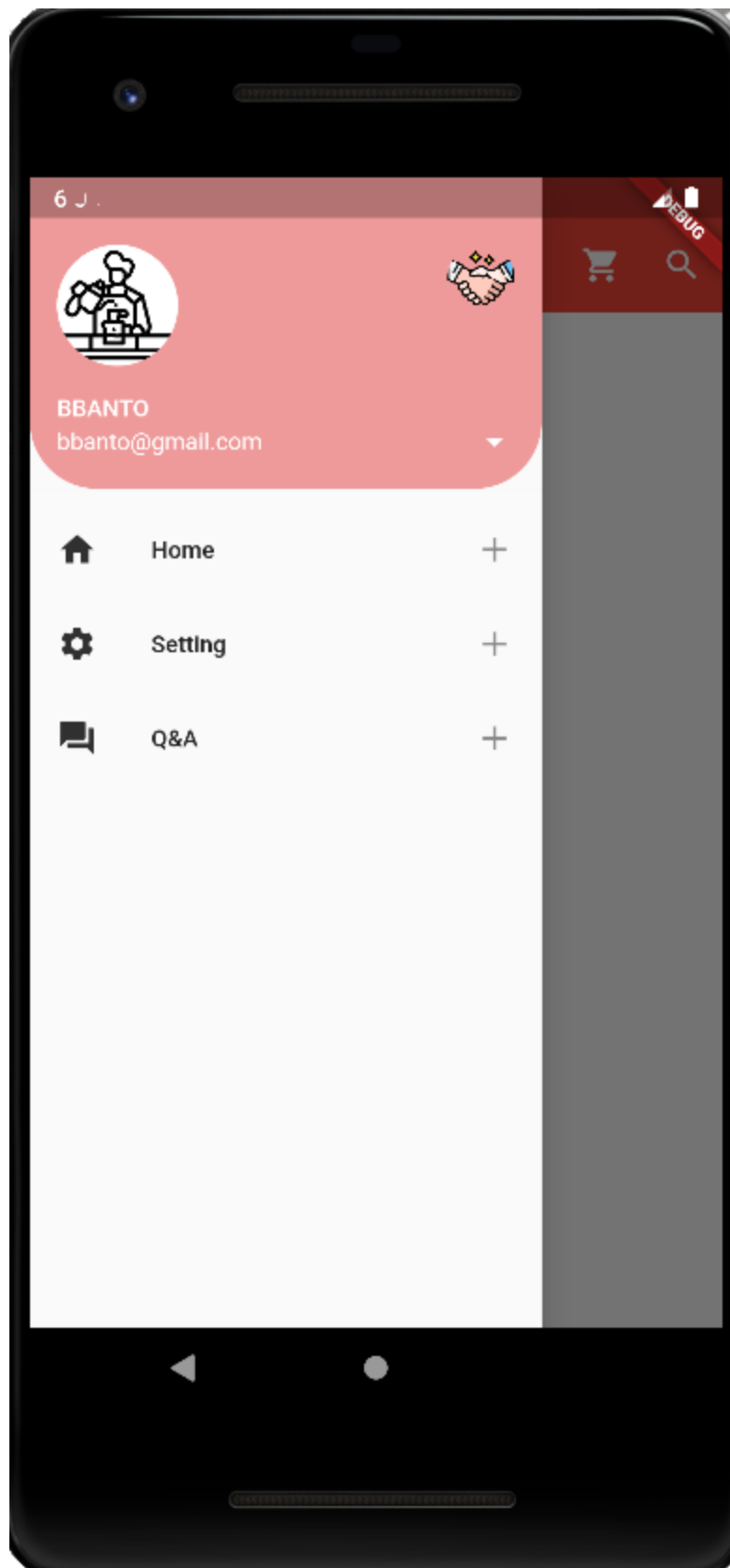
title: Text('AppBar icon menu'),
centerTitle: true,
elevation: 0.0,
actions: <Widget>[
  IconButton(
    icon: Icon(Icons.shopping_cart),
    onPressed: () {
      print('Shopping cart button is clicked');
    }, // 기본적인 icon 메뉴만 만들어서
  ),
  IconButton(
    icon: Icon(Icons.search),
    onPressed: () {
      print('search button is clicked');
    },
  ),
],
),
drawer: Drawer(
  // drawer 위젯 불러옴,
  child: ListView(
    // 여러 줄의 정보를 보여주는 목록이기에 child가 아닌 children 속성
    padding: EdgeInsets.zero,
    children: <Widget>[
      UserAccountsDrawerHeader(
        currentAccountPicture: CircleAvatar(
          backgroundImage: AssetImage('assets/barista.png'),
          backgroundColor: Colors.white,
        ),
        otherAccountsPictures: <Widget>[
          CircleAvatar(
            backgroundImage: AssetImage('assets/cooperate.png'),
            backgroundColor: Colors.red[200],
          ),
          // CircleAvatar(
          //   backgroundImage: AssetImage('assets/cooperate.png'),
          //   backgroundColor: Colors.red[200],
          // ),
        ],
        accountName: Text('BBANTO'),
        accountEmail: Text('bbanto@gmail.com'),
        onDetailsPressed: () {
          print('arrow is clicked');
        },
        decoration: BoxDecoration(
          color: Colors.red[200],
          borderRadius: BorderRadius.only(
            bottomLeft: Radius.circular(40.0),
            bottomRight: Radius.circular(40.0)),
        ),
      ),
      ListTile(
        leading: Icon(
          Icons.home,
          color: Colors.grey[850],

```

```

    ),
    title: Text('Home'),
    onTap: () {
      print('Home is clicked');
    },
    trailing: Icon(Icons.add),
  ),
  ListTile(
    leading: Icon(
      Icons.settings,
      color: Colors.grey[850],
    ),
    title: Text('Setting'),
    onTap: () {
      print('Setting is clicked');
    },
    trailing: Icon(Icons.add),
  ),
  ListTile(
    leading: Icon(
      Icons.question_answer,
      color: Colors.grey[850],
    ),
    title: Text('Q&A'),
    onTap: () {
      print('Q&A is clicked');
    },
    trailing: Icon(Icons.add),
  ),
],
),
),
);
}
}

```





## BuildContext

"Scaffold.of() called with a context that does not contain a Scaffold" → 에러 메시지 ...

1. widget tree에서 현재 widget의 위치를 알 수 있는 정보
  - flutter에서 모든 위젯은 build method(함수)를 가지고 있고, 이를 통해서 계층 구조를 만들어냄. class와 위젯의 정체라는 강의에서 함수에 대해 공부한 적이 있음.
    - dart의 모든 것은 type을 가지고 있음.
    - build라는 함수 명 앞에 Widget은 타입임, 즉 build 메서드는 Widget 타입. 그리고 build 메서드의 인자값은 BuildContext라는 타입의 context라는 인자임. → build 함수는 context라는 인자값을 대입한 scaffold라는 위젯을 return한다는 의미임.

⇒ 즉, build 함수는 scaffold라는 위젯을 return하는데. 이때 widget tree 상에서 어디에 위치하는지에 대한 정보를 가지고 있는 context라는 걸 넣어서 return을 한다는 의미임.

2. 모든 위젯은 자신만의 build context를 가지고 있음. 이 BuildContext는 stateless 위젯이나 state 빌드 메서드에 의해서 리턴 된 위젯의 부모가 된다.

### 1. ScaffoldMessenger class

→ Snackbar 구현이 이전보다 쉬워짐. Why ScaffoldMessenger?

- snackbar 더이상 scaffold에 의해서 관리되는 것이 아닌, 위젯 트리상에서 최상위에 있는 ScaffoldMessenger에 의해 만들어지는거임.

### 2. Buttons (TextButton, ElevatedButton, OutlinedButton)

### 3. Global key

#### 1. 빌더 위젯 없이 Snack bar 만들기

#### 2. Toast Message 구현하기.

: 사용자에게 짧은 메시지 형식으로 정보 전달하는 팝업임, 앱에서 정말 많이 사용하는 기능.

```

import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(primarySwatch: Colors.blue),
      home: MyPage(),
    );
  }
}

class MyPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Scaffold Messenger'),
      ),
      body: Homebody(),
      floatingActionButton: FloatingActionButton(
        child: Icon(Icons.thumb_up),
        onPressed: () {
          ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(
              content: Text('Like a new Snack bar'),
              duration: Duration(seconds: 5),
              action: SnackBarAction(
                label: 'Undo',
                onPressed: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(builder: (context) => ThirdPage()),
                  );
                },
              ),
            ),
          );
        },
      ),
    );
  }
}

class Homebody extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Center(
      child: ElevatedButton(
        child: Text('Go to the second page'),
        onPressed: () {

```

```

        Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => SecondPage()),
        );
    },
),
);
}
}

class SecondPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Second Page'),
      ),
      body: Center(
        child: Text(
          '"좋아요가 추가되었습니다"',
          style: TextStyle(fontSize: 20.0, color: Colors.redAccent),
        ),
      ),
    );
  }
}

class ThirdPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Third Page'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              '"좋아요"를 취소 하시겠습니까?',
              style: TextStyle(fontSize: 20.0, color: Colors.redAccent),
            ),
            ElevatedButton(
              onPressed: () {
                ScaffoldMessenger.of(context).showSnackBar(
                  SnackBar(
                    content: Text('"좋아요"가 취소되었습니다.'),
                    duration: Duration(seconds: 3),
                  ),
                );
              },
              child: Text('취소하기'),
            ),
          ],
        ),
      ),
    );
  }
}

```

```

    ),
  ),
);
}
}

```

```

import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(primarySwatch: Colors.blue),
      home: MyPage(),
    );
  }
}

class MyPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return ScaffoldMessenger(
      child: Scaffold(
        appBar: AppBar(
          title: Text('Scaffold Messenger'),
        ),
        body: Homebody(),
        floatingActionButton: FloatingActionButton(
          child: Icon(Icons.thumb_up),
          onPressed: () {
            ScaffoldMessenger.of(context).showSnackBar(
              SnackBar(
                content: Text('Like a new Snack bar'),
                duration: Duration(seconds: 5),
                action: SnackBarAction(
                  label: 'Undo',
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(builder: (context) => ThirdPage()),
                    );
                  },
                ),
              ),
            );
          },
        ),
      ),
    );
  }
}

```

```

}

class Homebody extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Center(
      child: ElevatedButton(
        child: Text('Go to the second page'),
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => SecondPage()),
          );
        },
      ),
    );
  }
}

class SecondPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Second Page'),
      ),
      body: Builder(builder: (context) {
        return Center(
          child: Text(
            '"좋아요가 추가되었습니다"',
            style: TextStyle(fontSize: 20.0, color: Colors.redAccent),
          ),
        );
      })
    );
  }
}

class ThirdPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Third Page'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              '"좋아요"를 취소 하시겠습니까?',
              style: TextStyle(fontSize: 20.0, color: Colors.redAccent),
            ),
            ElevatedButton(

```

```
onPressed: () {  
  ScaffoldMessenger.of(context).showSnackBar(  
    SnackBar(  
      content: Text('"좋아요"가 취소되었습니다.'),  
      duration: Duration(seconds: 3),  
    ),  
  );  
},  
child: Text('취소하기'),  
),  
],  
),  
),  
);  
}  
}
```