Flutter 화면 데이터 구성하기

데이터 표시 위젯 학습

학습목표

- Flutter의 기본 데이터 표시 위젯 이해하기
- 리스트뷰와 그리드뷰의 구현 방법 습득하기
- 반응형 헤더와 알림 구현 방법 학습하기

Flutter의 데이터 표시

- ListView를 사용한 세로/가로 리스트 구현 가능
- SliverAppBar를 통한 반응형 헤더 구현 제공
- GridView로 그리드 형태의 데이터 표시 지원
- SnackBar로 간단한 알림 메시지 표시 가능

세로형 ListView 구현

- ListView.builder를 사용한 효율적인 리스트 구현
- 스크롤이 가능한 세로형 목록 생성
- 데이터 클래스와 빌더 패턴 활용

ListView_vertical.dart

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

// ListTile에 표시될 항목을 정의하는 클래스
class ListTileItem {
  final String monthItem;
  const ListTileItem({required this.monthItem});
}

// ListView에 사용될 데이터 소스를 정의하는 클래스
class ListDataItems {
  final List<String> monthItems = [
    'January', 'February', 'March', 'April',
    'May', 'June', 'July', 'August',
    'September', 'October', 'November', 'December',
  ];
  ListDataItems();
}

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

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

  @override
  Widget build(BuildContext context) {
    const title = 'MyAwesomeApp';
    return MaterialApp(
      title: title,
      home: Scaffold(
        appBar: AppBar(
          title: const Text(title),
        ),
        body: MyListView(),
      ),
    );
  }
}

class MyListView extends StatelessWidget {
  MyListView();
  final ListDataItems item = ListDataItems();

  @override
  Widget build(BuildContext context) {
    return ListView.builder(
      itemCount: item.monthItems.length, // ListView에 표시할 항목 수
      itemBuilder: (context, index) {
        return ListTile(
          title: Text(item.monthItems[index]) // 각 ListTile에 월 이름 표시
        );
      },
    );
  }
}
```

가로형 ListView 구현

- scrollDirection: Axis.horizontal을 사용한 가로 리스트 구현
- Row 위젯을 사용하여 가로로 항목 배치
- 필요에 따라 높이와 너비 조정

ListView_horizontal.dart

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

// ListTile에 표시될 항목을 정의하는 클래스
class ListTileItem {
  final String monthItem;
  const ListTileItem({required this.monthItem});
}

// ListView에 사용될 데이터 소스를 정의하는 클래스
class ListDataItems {
  final List<String> monthItems = [
    'January', 'February', 'March', 'April',
    'May', 'June', 'July', 'August',
    'September', 'October', 'November', 'December',
  ];
  ListDataItems();
}

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

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

  @override
  Widget build(BuildContext context) {
    const title = 'MyAwesomeApp';
    return MaterialApp(
      title: title,
      home: Scaffold(
        appBar: AppBar(
          title: const Text(title),
        ),
        body: MyListView(),
      ),
    );
  }
}

class MyListView extends StatelessWidget {
  MyListView({Key? key}) : super(key: key);
  final ListDataItems item = ListDataItems();

  @override
  Widget build(BuildContext context) {
    return ListView.builder(
      scrollDirection: Axis.horizontal, // 가로 방향으로 스크롤 설정
      itemCount: item.monthItems.length, // ListView에 표시할 항목 수
      itemBuilder: (context, index) {
        return Row(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(item.monthItems[index]), // 각 항목에 월 이름 표시
            const SizedBox(width: 10.0), // 항목 간 간격 추가
          ],
        );
      },
    );
  }
}
```

SliverAppBar를 사용한 반응형 헤더 구현

- 스크롤에 따라 AppBar의 높이 변경
- expandedHeight와 collapsedHeight 속성 사용
- floating 속성으로 AppBar의 동작 제어

SliverAppBar_example.dart

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'SliverAppBar Demo',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const MyHomePage(title: 'Flutter and Dart Cookbook'),
    );
  }
}

class MyHomePage extends StatelessWidget {
  final String title;

  const MyHomePage({
    Key? key,
    required this.title,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[300],
      body: const CustomScrollView(
        slivers: [
          SliverAppBar(
            leading: Icon(Icons.menu), // leading 아이콘 추가
            title: Text('Sliver App Bar'), // AppBar 제목
            expandedHeight: 300, // 확장된 높이
            collapsedHeight: 150, // 축소된 높이
            floating: false, // 스크롤 시 AppBar 고정 여부
          ),
        ],
      ),
    );
  }
}
```

SliverAppBar의 기능

- 스크롤에 반응하는 앱바 구현 가능
- expandedHeight로 확장 높이 지정
- collapsedHeight로 축소 높이 지정
- floating 속성으로 스크롤 시 동작 제어

SliverList를 사용한 스크롤 가능한 리스트 구현

- SliverAppBar와 함께 사용
- SliverChildBuilderDelegate를 사용하여 리스트 항목 생성
- CustomScrollView 내에서 SliverList 사용

SliverList_example.dart

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

// CarItem 클래스: 자동차 정보를 담는 데이터 모델
class CarItem {
  final String title;   // 자동차 제목
  final String subtitle;  // 자동차 부제목
  final String url;       // 자동차 이미지 URL

  CarItem({
    required this.title,
    required this.subtitle,
    required this.url,
  });
}

// ListDataItems 클래스: 자동차 아이템 리스트를 관리
class ListDataItems {
  final List<CarItem> carItems = [
    CarItem(
      title: '911 Cabriolet',
      subtitle: '911 Carrera Cabriolet Porsche',
      url: 'https://oreil.ly/m3OXC',
    ),
    CarItem(
      title: '718 Spyder',
      subtitle: '718 Spyder Porsche',
      url: 'https://oreil.ly/hca-6',
    ),
    CarItem(
      title: '718 Boxster T',
      subtitle: '718 Boxster T Porsche',
      url: 'https://oreil.ly/Ws4EX',
    ),
    CarItem(
      title: 'Cayenne',
      subtitle: 'Cayenne S Porsche',
      url: 'https://oreil.ly/gwvnL',
    ),
  ];

  ListDataItems();
}
```

SliverList의 활용

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

class MySliverList extends StatelessWidget {
  MySliverList({Key? key}) : super(key: key);
  final ListDataItems item = ListDataItems();

  @override
  Widget build(BuildContext context) {
    return SliverList(
      delegate: SliverChildBuilderDelegate(
        (context, index) => ListTile(
          leading: CircleAvatar(
            backgroundImage: NetworkImage(item.carItems[index].url),
          ),
          title: Text(item.carItems[index].title),
          subtitle: Text(item.carItems[index].subtitle),
        ),
        childCount: item.carItems.length,
      ),
    );
  }
}
```

GridView를 사용한 그리드 레이아웃 구현

- SliverGridDelegateWithFixedCrossAxisCount를 사용하여 그리드 설정
- itemCount로 그리드 항목 수 설정
- itemBuilder를 사용하여 각 그리드 항목 생성

GridView_example.dart

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

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

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

  @override
  Widget build(BuildContext context) {
    const title = 'MyAwesomeApp';

    return MaterialApp(
      title: title,
      home: Scaffold(
        appBar: AppBar(
          title: const Text(title),
        ),
        body: const MyGridViewBuilderWidget(),
      ),
    );
  }
}

class MyGridViewBuilderWidget extends StatelessWidget {
  const MyGridViewBuilderWidget({Key? key}) : super(key: key);
  final gridItems = 10; // 그리드 아이템의 총 개수

  @override
  Widget build(BuildContext context) {
    return GridView.builder(
      itemCount: gridItems, // 그리드 아이템의 총 개수 설정
      gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: 5 // 가로축에 5개의 아이템을 배치
      ),
      itemBuilder: (context, index) {
        return Padding(
          padding: const EdgeInsets.all(8.0),
          child: Container(
            height: 50,
            width: 50,
            color: Colors.blue,
            child: Center(child: Text(index.toString())), // 각 아이템의 텍스트 표시
          ),
        );
      },
    );
  }
}
```

SnackBar를 사용한 알림 메시지 표시

- ScaffoldMessenger를 사용하여 SnackBar 표시
- duration 속성으로 SnackBar 표시 시간 설정
- action 속성으로 SnackBar에 액션 추가

SnackBar_example.dart

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

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

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

  @override
  Widget build(BuildContext context) {
    const title = 'MyAwesomeApp';

    return MaterialApp(
      title: title,
      home: Scaffold(
        appBar: AppBar(
          title: const Text(title),
        ),
        body: MyListView(),
      ),
    );
  }
}

class ListViewData {
  final List<String> monthItems = [
    'January',
    'February',
    'March',
  ];
}

class MyListView extends StatelessWidget {
  MyListView({Key? key}) : super(key: key);
  final ListViewData items = ListViewData();

  @override
  Widget build(BuildContext context) {
    return ListView.builder(
      itemCount: items.monthItems.length,
      itemBuilder: (context, index) {
        return ListTile(
          title: Text(items.monthItems[index]),
          onTap: () {
            ScaffoldMessenger.of(context).showSnackBar(
              SnackBar(
                content: Text('You selected ${items.monthItems[index]}'),
              ),
            );
          },
        );
      },
    );
  }
}
```

요약

- ListView를 활용한 세로/가로 리스트 구현 방법 학습
- SliverAppBar와 SliverList를 통한 반응형 UI 구현
- GridView를 사용한 그리드 레이아웃 구성 방법 이해
- SnackBar를 활용한 알림 메시지 표시 기능 구현

END