

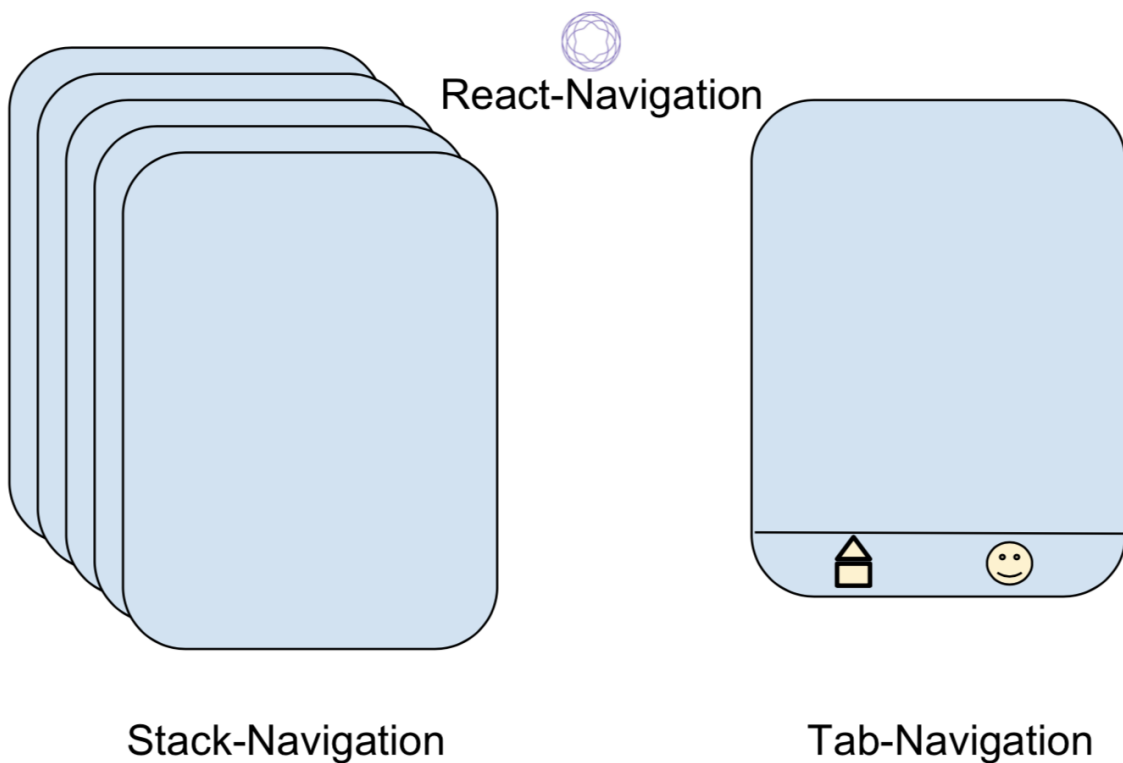
## Практика 4 - Навигация и меню приложения

Flutter позволяет создать несколько страниц приложения и далее осуществлять навигацию между ними. Мы научимся работать с навигацией и дополнительно поговорим относительно создания меню в приложении.

### Навигация «Stack Navigation»

Существуют разные форматы навигации для приложения. В ходе курса мы будем реализовывать одну из наиболее распространённых концепций навигации – Stack Navigation.

Суть этой навигации заключается в том, что каждая следующая сцена (страница) накладывается поверх прошлой. Пример такой реализации показан на фото ниже:



### Реализация навигации

Для реализации навигации необходимо создать несколько классов (страниц). После этого вы можете прописать маршрутизацию в главном файле и далее для переходов использовать класс Navigator.

#### Main.dart:

```
import 'package:flutter/material.dart';
import 'package:flutter_todo/pages/home.dart';
import 'package:flutter_todo/pages/main_screen.dart';

void main() => runApp(MaterialApp(
```

```

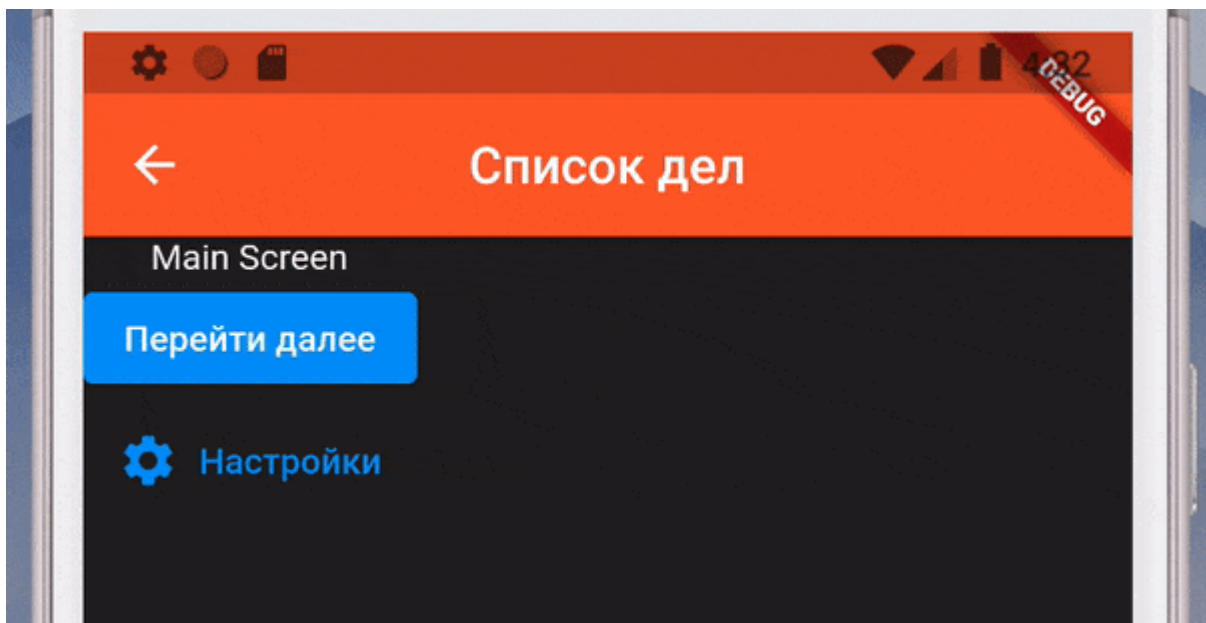
theme: ThemeData(
  primaryColor: Colors.deepOrangeAccent,
),
initialRoute: '/',
routes: {
  '/': (context) => MainScreen(),
  '/todo': (context) => Home(),
},
));

```

### Отдельная страница

Создайте страницу «/settings». Пропишите функционал для перехода с главной страницы «MainScene» на страницу «Settings».

Пример реализации:



### Home.dart:

```

import 'package:flutter/material.dart';

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

  @override
  _HomeState createState() => _HomeState();
}

class _HomeState extends State<Home> {
  String _userToDo;
  List todoList = [];

  @override
  void initState() {
    super.initState();

    todoList.addAll(['Buy milk', 'Wash dishes', 'Купить картошку']);
  }

  void _menuOpen() {
    Navigator.of(context).push(
      MaterialPageRoute(builder: (BuildContext context) {
        return Scaffold(

```

```

appBar: AppBar(title: Text('Меню'),),
body: Row(
  children: [
    ElevatedButton(onPressed: () {
      Navigator.pop(context);
      Navigator.pushNamedAndRemoveUntil(context, '/', (route) => false);
    }, child: Text('На главную')),
    Padding(padding: EdgeInsets.only(left: 15)),
    Text('Наше простое меню')
  ],
)
);
})
);
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey[900],
    appBar: AppBar(
      title: Text('Список дел'),
      centerTitle: true,
      actions: [
        IconButton(
          icon: Icon(Icons.menu_outlined),
          onPressed: _menuOpen,
        ),
      ],
    ),
    body: ListView.builder(
      itemCount: todoList.length,
      itemBuilder: (BuildContext context, int index) {
        return Dismissible(
          key: Key(todoList[index]),
          child: Card(
            child: ListTile(
              title: Text(todoList[index]),
              trailing: IconButton(
                icon: Icon(
                  Icons.delete_sweep,
                  color: Colors.deepOrangeAccent
                ),
                onPressed: () {
                  setState(() {
                    todoList.removeAt(index);
                  });
                },
              ),
            ),
          onDismissed: (direction) {
            // if(direction == DismissDirection.endToStart)
            setState(() {
              todoList.removeAt(index);
            });
          },
        );
      },
    ),
    floatingActionButton: FloatingActionButton(
      backgroundColor: Colors.greenAccent,
      onPressed: () {
        showDialog(context: context, builder: (BuildContext context) {
          return AlertDialog(
            title: Text('Добавить элемент'),
            content: TextField(
              onChanged: (String value) {

```

```

      _userToDo = value;
    },
  ),
  actions: [
    ElevatedButton(onPressed: () {
      setState(() {
        todoList.add(_userToDo);
      });

      Navigator.of(context).pop();
    }, child: Text('Добавить'))
  ],
);
});
},
child: Icon(
  Icons.add_box,
  color: Colors.white
),
),
);
}
}

```

## Main\_screen.dart:

```

import 'package:flutter/material.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[900],
      appBar: AppBar(
        title: Text('Список дел'),
        centerTitle: true,
      ),
      body: Column(
        children: [
          Text('Main Screen', style: TextStyle(color: Colors.white)),
          ElevatedButton(onPressed: () {
            Navigator.pushReplacementNamed(context, '/todo');
          }, child: Text('Перейти далее')),
          TextButton.icon(onPressed: () {
            // Новая кнопка и переход на новую страницу
            Navigator.pushNamed(context, '/settings');
          }, icon: Icon(Icons.settings), label: Text('Настройки')),
        ],
      ),
    );
  }
}

```

## Settings.dart:

```

import 'package:flutter/material.dart';

// Описываем страницу настроек
class Settings extends StatelessWidget {

```

```

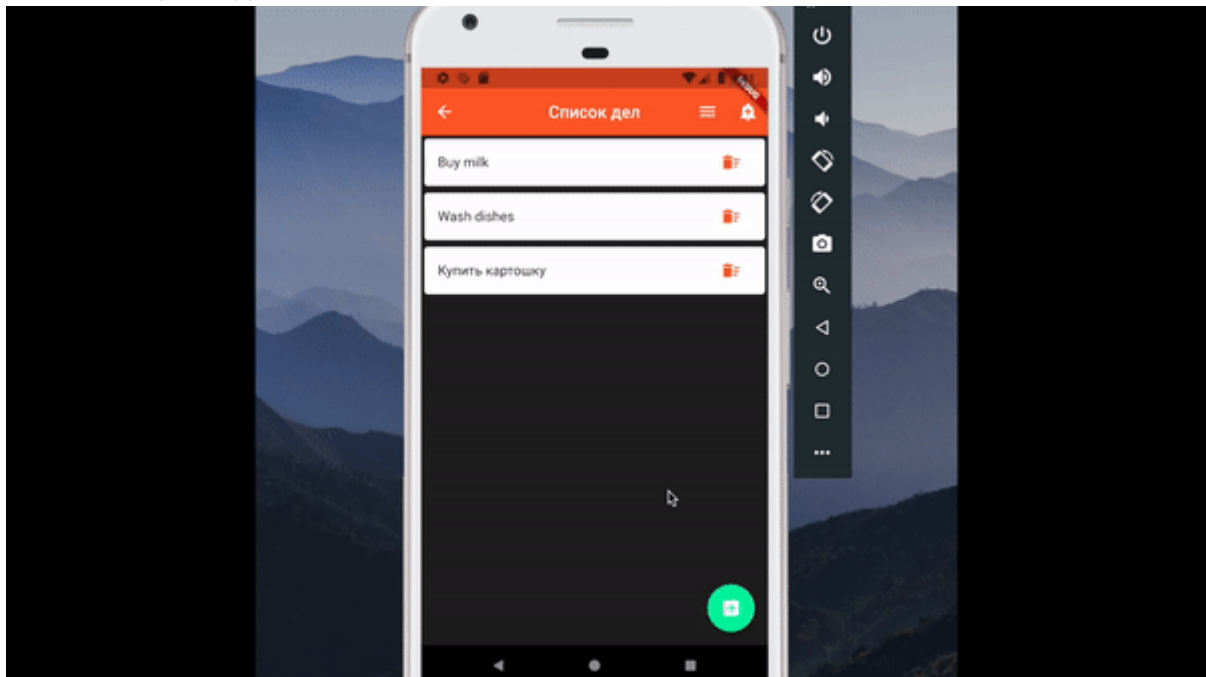
const Settings({Key key}) : super(key: key);

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey[900],
    appBar: AppBar(
      title: Text('Настройки'),
      centerTitle: true,
    ),
    body: Container(
      margin: EdgeInsets.all(50),
      child: Column(
        children: [
          Text('Страница с настройками', style: TextStyle(color: Colors.white)),
          // Кнопка для перехода на главную страницу
          TextButton.icon(onPressed: () {
            Navigator.pushNamed(context, '/');
          }, label: Text('Назад'), icon: Icon(Icons.arrow_back)),
        ],
      ),
    ),
  );
}

```

### Улучшенное меню

Добавьте в меню еще одну кнопку. При нажатии на неё должна показываться всплывающая подсказка.



Ниже приводим код для appBar. Код нужно было менять исключительно для шапки:

```

appBar: AppBar(
  title: Text('Список дел'),
  centerTitle: true,

```

```

actions: [
  IconButton(
    icon: Icon(Icons.menu_outlined),
    onPressed: _menuOpen,
  ),
  // Создание еще одной кнопки
  IconButton(
    icon: Icon(Icons.add_alert),
    tooltip: 'Подсказка', // Подсказка при наведении
    onPressed: () {
      // При нажатии показываем всплывающее окно
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Это подсказка! :))'))
    },
  ),
],
),

```

## Подключение базы данных Firebase

К любому Flutter проекту можно быстро подключить базу данных. За урок мы научимся подключать и работать с базой данных Firebase. Нами будет переделана программа «Список дел» и все данные из программы будут храниться в удаленной базе Firebase.

### Полезные ссылки:

- Официальный [сайт Firebase](#);
- Инструкции относительно [Cloud Firestore](#).

### Исправление ошибок

При запуске проекта может выдаваться ошибка связанная с тем, что FireBase не запущен. Такое может происходить по причине того, что его инициализацию нужно прописывать в основном методе.

Для устранения ошибки в файле «main.dart» пропишите:

```

// Подключени всех сцен
import 'package:flutter_todo/pages/home.dart';
import 'package:flutter_todo/pages/main_screen.dart';

import 'package:firebase_core/firebase_core.dart';

void main() async {
  // Инициализация в основной функции
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();

  runApp(MaterialApp(
    theme: ThemeData(
      primaryColor: Colors.deepOrangeAccent,

```

```
),  
initialRoute: '/',  
routes: {  
  // Маршрутизация  
  '/': (context) => MainScreen(),  
  '/todo': (context) => Home(),  
},  
));  
}
```

## База данных Firebase

При разработке Flutter программ вы можете использовать практически любую базу данных. В вашем распоряжении такие БД, как:

- SQLite;
- MySQL;
- MongoDB;
- Firebase;
- и множество других.

В ходе пракики мы использовали БД Firebase, так как она одна из наиболее простых для интеграции в Flutter и обладает широким спектром возможностей.

Благодаря тому, что Firebase была куплена компанией Google в 2014 году, то интеграция этой БД и Flutter стала легким и быстрым процессом. Напомним, что Flutter также принадлежит компании Google.

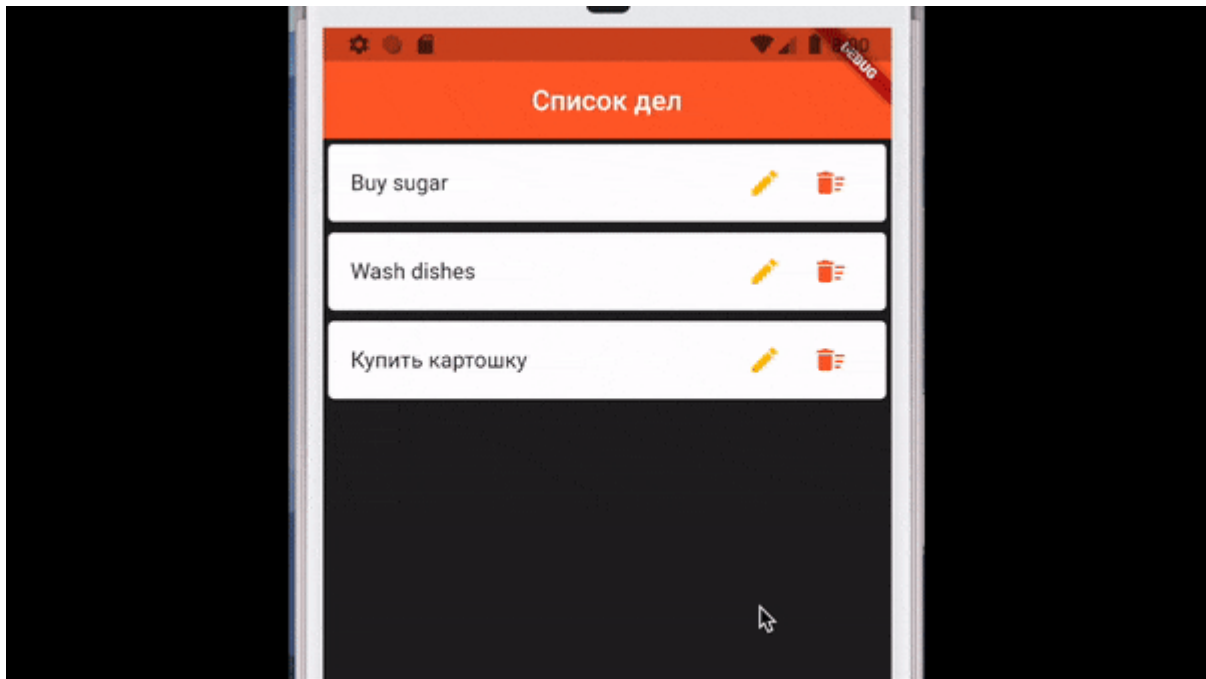
## Задание

Редактирование элементов

Ранее вы должны были создать кнопку для редактирования элементов.

Добавьте к этой кнопке функционал по редактированию элементов из базы данных.

Пример реализации:



Решение задания получилось большим, поэтому мы прикрепляем лишь измененную часть. Вам необходимо было изменить блок ListTile и в нем свойство «trailing»:

```
trailing: Wrap(
  children: [
    IconButton(
      icon: Icon(
        Icons.delete_sweep,
        color: Colors.deepOrangeAccent
      ),
      onPressed: () {
        FirebaseFirestore.instance.collection('items').doc(snapshot.data.docs[index].id).delete();
      },
    ),
    IconButton(
      icon: Icon(
        Icons.edit,
        color: Colors.amber
      ),
      onPressed: () {
        showDialog(context: context, builder:
(BuildContext context) {
          return AlertDialog(
            title: Text('Изменить элемент'),
            content: TextField(
              onChanged: (String value) {
                _userToDo = value;
              },
            ),
          ),
        },
      ),
    ),
  ],
),
```



```

        actions: [
          ElevatedButton(onPressed:
() {
          // Изменение элемента в БД

FirebaseFirestore.instance.collection('items').doc(snapshot.data.docs[index].id).set({'item': _userToDo});

Navigator.of(context).pop();
          }, child: Text('Изменить'))
        ],
      ),
    ),
  ],
),

```