

Mobile App Design and Development

Dart

- basic boilerplate

```
void main() {  
    //TODO  
}
```

- functions

```
dataType funcName(dataType param1, dataType param2) {  
    // TODO  
}
```

- data types

```
void main() {  
    String name;  
    int age;  
    double weight;  
    bool isVerified;  
}
```

- displaying and accepting inputs

```
import 'dart:io';  
  
void main() {  
    print(stdin.readLineSync());  
}
```

- variable declaration

```
var/ dataType variableName;
```

- object and class

```
class Student {  
    String? name;  
    int? age;  
  
    Student(this.name, this.age);  
  
}  
  
void main() {  
    var p1 = new Student("bes", 21);  
    print(p1.name);  
    print(p1.age);  
}
```

- inheritance

```
class Student {  
    String? name;  
    int? age;  
  
    Student(this.name, this.age);  
  
}  
  
class StudentCouncil extends Student {  
  
    String? role;  
  
    StudentCouncil(String name, int age, this.role) : super(  
  
}  
  
void main() {  
    var p1 = StudentCouncil("bes", 21, "president");  
}
```

```

    print(p1.name);
    print(p1.age);
    print(p1.role);
}

```

- named constructors

```

class Student {
    String? name;
    int? age;

    Student({this.name, this.age});
}

void main() {
    var p1 = Student(name: "bes", age: 21);
    print(p1.name);
    print(p1.age);
}

```

- private fields, methods and class

```

// private class
class _MyClass {

}

// private method
void _add(x, y) {
    return x + y;
}

void main() {
    // private field
    var _sum = _add(1, 2);
    print(_sum);
}

```

- enum

```
enum Gender {
    male,
    female
}

class Student {
    String name;
    int age;
    var gender;

    Student({required this.name, required this.age, required

}

void main() {
    var p1 = Student(name: "bes", age: 21, gender: Gender.ma
    print(p1.name);
    print(p1.age);
    print(p1.gender);
}
```

- multiple constructors

```
enum Gender {
    male,
    female
}

class Student {
    String name;
    int age;
    var gender;

    Student(this.name, this.age, this.gender);

    Student.named({required this.name, required this.age, re
```

```

}

void main() {
    var p1 = Student.named(name: "bes", age: 21, gender: Gen
    var p2 = Student("john", 20, Gender.male);
    print(p1.name);
    print(p1.age);
    print(p1.gender);
    print(p2.name);
    print(p2.age);
    print(p2.gender);
}

```

- map data structure

```

void main(){

    var data = {
        "H": 1,
        "He": 2,
        "B": 3,
        "C": 4,
        "N": 5,
        "O": 6,
    };

    data.forEach((k, v) => print("Key : $k, Value : $v"));
}

```

- final vs const

```

final // runtime constant
const // compile time constant

```

- immutable lists

```
var data = const [];
```

- getter

```
dataType get variableName {  
    return  
}
```

- mixins

```
mixin Flying {  
    void fly(){  
        print("flying...");  
    }  
}  
  
abstract class AirbornInsect with Flying {  
    void buzz() {  
        print("buzzing");  
    }  
    void doInsectThings() {  
        fly();  
        buzz();  
    }  
}  
  
abstract class Bird with Flying {  
    void chirp() {  
        print("chirp chirp");  
    }  
    void doInsectThings() {  
        fly();  
        chirp();  
    }  
}
```

- factory

```
// create Class Car
class Car {
    //class properties
    String name;
    String color;

    //constructor
    Car({ this.name, this.color});

    // factory constructor that returns a new instance
    factory Car.fromJson(Map json) {
        return Car(name : json['name'],
            color : json['color']);
    }
}
```

- Cascade operator

```
import 'dart:convert';

//An Example class with member attributes and methods
class Example{
    var a;
    var b;
    void bSetter(b)
    {
        this.b = b;
    }
    void printValues(){
        print(this.a);
        print(this.b);
    }
}

void main() {
    //Instantiating two Example objects
    Example eg1 = new Example();
    Example eg2 = new Example();
}
```

```

    //Using the .. operator for operations on Example object
    print("Example 1 results:");
    eg1
    ..a = 88
    ..bSetter(53)
    ..printValues();

    //The same operations as above but without the .. operator
    print("Example 2 results:");
    eg2.a = 88;
    eg2.bSetter(53);
    eg2.printValues();
}

```

Flutter Basics

- App boilerplate
 - Basic structure

```

// importing widgets from flutter package
import 'package:flutter/material.dart';

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

// class that wraps widgets in our app
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(home: Text('hello flutter (^
  }
}

```


- required named constructor

```
class Student {
    String? name;
    int? age;

    Student({@required this.name, @required this.age});
}

void main() {
    var p1 = Student(name: "bes", age: 21);
    print(p1.name);
    print(p1.age);
}
```

- widget tree

```
// importing widgets from flutter package
import 'package:flutter/material.dart';

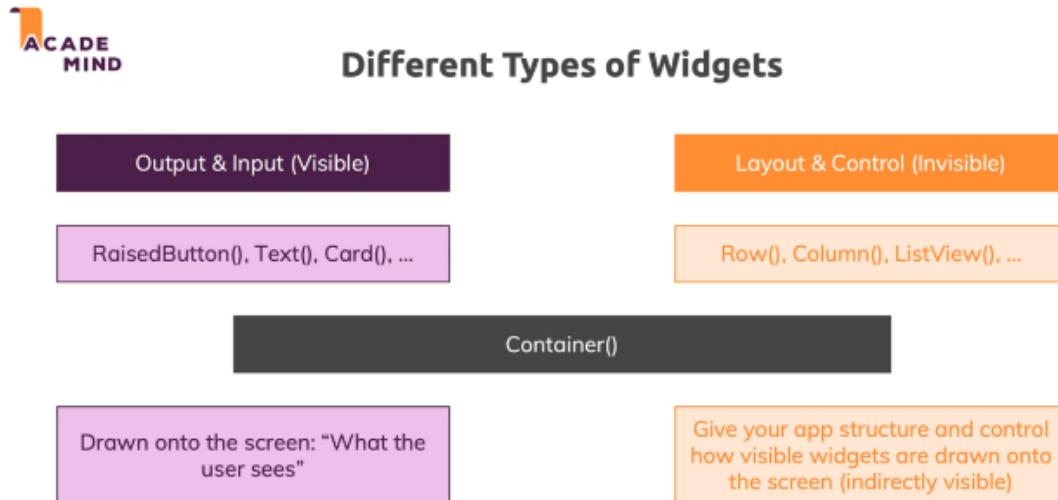
// main function
void main() {
    runApp(const MyApp());
}

// class that wraps widgets in our app
class MyApp extends StatelessWidget {
    const MyApp({super.key});

    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            home: Scaffold(
                appBar: AppBar(title: Text('Demo')),
                body: Text('hello flutter (^ - ^)')
            ),
        );
    }
}
```

```
}
}
```

- Types of widgets



- Output and input widgets (visible widgets)

- Text

```
Text(
  'txt',
  style: TextStyle(
    fontSize: 16
  ),
  textAlign: TextAlign.center
)
```

- Button

- ElevatedButton

```
void click() {
  print('clicked');
}

ElevatedButton(
```

```

        child: Text('Click me'),
        onPressed: click or () => {
            print('clicked');
        },
    ),
)

```

- Layout and control widgets (invisible widgets)

- Container

```

Container(
  width: double.infinity,
  margin: EdgeInsets.all(10),
  child: Text('yay')
)

```

-

- Row

```

// importing widgets from flutter package
import 'package:flutter/material.dart';

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

// class that wraps widgets in our app
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text('Demo')),
        body: Row(
          children: <Widget>[]

```

```

        ),
      ),
    );
  }
}

```

■ Column

```

// importing widgets from flutter package
import 'package:flutter/material.dart';

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

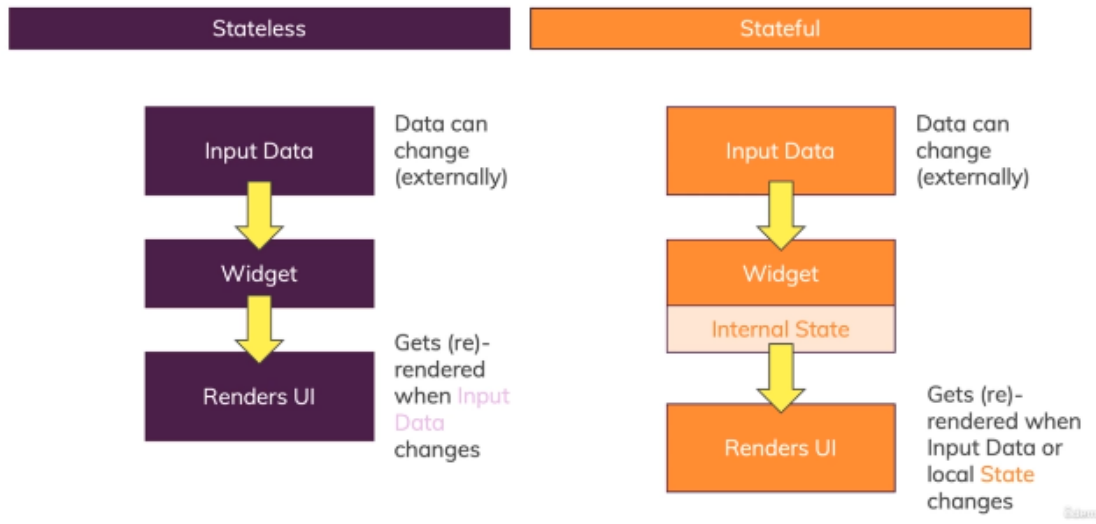
// class that wraps widgets in our app
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text('Demo')),
        body: Column(
          children: <Widget>[]
        )
      ),
    );
  }
}

```

- Statefull vs Stateless widgets

Stateless vs Stateful



- Stateless widget

```
// importing widgets from flutter package
import 'package:flutter/material.dart';

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

// class that wraps widgets in our app
class MyApp extends StatelessWidget {
  MyApp({super.key});

  var text = 'waiting';
  void click() {
    text = 'clicked';
    print(text);
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
```

```

        appBar: AppBar(
          title: Text('Demo'),
        ),
        body: Column(children: [
          Text(text),
          ElevatedButton(
            child: Text('Click me'),
            onPressed: click,
          )
        ]),
      ));
    }
  }
}

```

- Statefull widget

```

// importing widgets from flutter package
import 'package:flutter/material.dart';

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

// class that wraps widgets in our app
class MyApp extends StatefulWidget {
  MyApp({super.key});

  @override
  State<StatefulWidget> createState() {
    return MyAppState();
  }
}

class MyAppState extends State<MyApp> {
  var text = 'waiting';
  void click() {
    setState(() {

```

```

        text = 'clicked';
    });
    print(text);
}

@override
Widget build(BuildContext context) {
    return MaterialApp(
        home: Scaffold(
            appBar: AppBar(
                title: Text('Demo'),
            ),
            body: Column(children: [
                Text(text),
                ElevatedButton(
                    child: Text('Click me'),
                    onPressed: click,
                )
            ]),
        ));
}

```

- Custom widgets

```

/* mytext.dart */

import 'package:flutter/material.dart';

class MyText extends StatelessWidget {
    final String text;

    MyText({required this.text});

    @override
    Widget build(BuildContext context) {
        return Text(
            text,

```

```

        selectionColor: Color.fromRGBO(58, 102, 81, 1),
    );
}
}

/* mybutton.dart */

import 'package:flutter/material.dart';

class MyButton extends StatelessWidget {
    final VoidCallback callBackHandler;
    final String text;

    MyButton({this.callBackHandler, this.text});

    ButtonStyle style = ElevatedButton.styleFrom(
        backgroundColor: Color.fromARGB(255, 85, 108, 128));

    @override
    Widget build(BuildContext context) {
        return Container(
            width: double.infinity,
            margin: EdgeInsets.all(10),
            child: ElevatedButton(
                child: Text('Click me'),
                style: style,
                onPressed: callBackHandler,
            ),
        );
    }
}

/* main.dart */

// importing widgets from flutter package
import 'package:flutter/material.dart';

```



```

// importing custom widget
import './mytext.dart';
import './mybutton.dart';

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

// class that wraps widgets in our app
class MyApp extends StatefulWidget {
  MyApp({super.key});

  @override
  State<StatefulWidget> createState() {
    return MyAppState();
  }
}

class MyAppState extends State<MyApp> {
  var text = 'waiting';
  void click() {
    setState(() {
      text = 'clicked';
    });
    print(text);
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Demo'),
        ),
        body: Column(children: [
          MyText(text: text),
          MyButton(

```

```

        callbackHandler: click,
        text: 'click me'
      ),
    ]),
  ));
}
}

```

- Iterative rendering

```

// importing widgets from flutter package
import 'package:flutter/material.dart';

// importing custom widget
import './mytext.dart';
import './mybutton.dart';

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

// class that wraps widgets in our app
class MyApp extends StatefulWidget {
  MyApp({super.key});

  @override
  State<StatefulWidget> createState() {
    return MyAppState();
  }
}

class MyAppState extends State<MyApp> {
  var index = 0;

  var datas = [
    {
      'text': 'waiting',

```

```

        'buttonText': ['click me', 'nope']
      },
      {
        'text': 'terms and policies',
        'buttonText': ['accept', 'decline']
      },
    ],
  );

  void click() {
    setState(() {
      index++;
    });
    print(index);
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Demo'),
        ),
        body: Column(children: [
          MyText(text: (datas[index]['text'] as String)),
          ...(datas[index]['buttonText'] as List<String>).map(
            (txt) => MyButton(
              callbackHandler: click,
              text: txt,
            ),
          ),
        ]),
      ),
    );
  }
}

```

- conditional rendering

```

// importing widgets from flutter package
import 'package:flutter/material.dart';

```

```

// importing custom widget
import './mytext.dart';
import './mybutton.dart';

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

// class that wraps widgets in our app
class MyApp extends StatefulWidget {
  MyApp({super.key});

  @override
  State<StatefulWidget> createState() {
    return MyAppState();
  }
}

class MyAppState extends State<MyApp> {
  var index = 0;
  var isAccepted = false;

  var datas = [
    {
      'text': 'terms and policies',
      'buttonText': ['accept']
    },
  ];

  void click() {
    setState(() {
      isAccepted = true;
    });
  }

  @override
  Widget build(BuildContext context) {

```

```

return MaterialApp(
  home: Scaffold(
    appBar: AppBar(
      title: Text('Demo'),
    ),
    body: isAccepted
      ? (Center(
          child: Text('Welcome'),
        ))
      : (Column(children: [
          MyText(text: (datas[index]['text'] as String)
            ...(datas[index]['buttonText'] as List<String>
              return MyButton(callBackHandler: click, text:
            )).toList()
          ])),
    ));
}
}

```

•