# تعلم برمجة تطبيقات الجوال بالفلاتر والدارت

# مقدمة

الجزء الأول

# تثبيت وإعداد بيئة البرمجة

To get started right away, just tap any placeholder text (such as this) and start typing.

In this first section, we give some useful details about the development environment that will use during this series of flutter courses.

## First Application

بالنسبة لتطبيق الأيفون، لنا الخيار إما أن يكون الكود باستعمال لغة البرمجة Swift أو Objective C.

يمكن تحديد ذلك في أمر إنشاء التطبيق من خلال الخيار -i swift أو -i objc

كذلك بالنسبة لتطبيق الأندرويد، لنا الخيار بين الجافا Java أو Kotlin.

ويمكن تحديد ذلك في أمر الإنشاء من خلال الخيار -a java أو -a kotlin

لمعرفة جمبع الإمكانيات المتاحة، ما عليك إلا أن تكتب:

flutter create --help

We have a main function which the entry point of the program.

With build function, we render something to the screen.

# الأزرار Button

## Overview

We start by exploring button widget.

Flutter offers mainly three types of buttons: Raised Button, Flat Button and Icon Button.

### RaisedButton

RaisedButton is one of the most widely used widget in the flutter material library. It is actually a simple button which can handle normal click event. But what make it so great is its easiness of development.

#### Example 1

We create an empty new project using Visual Studio as shown in the previous section.

flutter create --androidx -t app --org com.ah.book -a java -i swift raisedbutton

The code of the main.dart file is given below:

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter RisedButton Demo'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

String \_value = 'Welcome';

void \_onPressed() {

setState((){

\_value = 'Welcome in KSA';

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text(widget.title),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Column(

children: <Widget>[

Text(\_value),

RaisedButton(onPressed: \_onPressed, child: new Text('Click me'),)

],

),

),

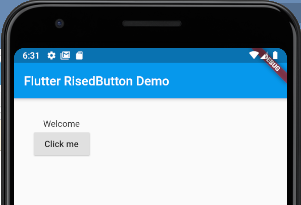
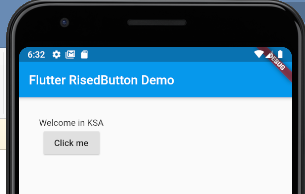
);

}

}

In this example, we have a string variable called \_value initialized with value “Welcome”. This \_value is assigned to the Text widget. We have a function called \_onPressed() used to change the value of the \_value variable.

We have a RaisedButton. When we click on that button, the text will change according to the new value.

#### Example 2

In this second example, we pass the text to display as a parameter of the function \_onPressed.

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter RisedButton Demo'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

String \_value = 'Welcome';

void \_onPressed(String v) {

setState((){

\_value = v;

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text(widget.title),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Column(

children: <Widget>[

Text(\_value),

RaisedButton(onPressed: () => \_onPressed('Saudi Arabia'), child: new Text('Click me'),)

],

),

),

);

}

}

#### Example 3

##### Tip 1

Here, we add a new parameter color to the RaisedButton Widget:

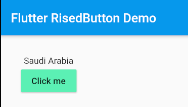
RaisedButton(

color: Colors.greenAccent,

onPressed: () => \_onPressed('Saudi Arabia'),

child: new Text('Click me'),

),



There are other coloring arguments to RaisedButton. They are focusColor, highlightColor and hoverColor.

##### Tip 2

If the onPressed is null, the button will be disabled. We can change the color of the button and its text while disabled using:

RaisedButton(

disabledColor: Colors.greenAccent[100],

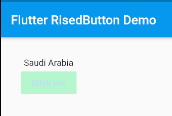
disabledTextColor: Colors.blue[100],

color: Colors.greenAccent,

onPressed: null,

child: new Text('Click me'),

),



##### Tip 3

The elevation will give a feeling of rise from its background.

RaisedButton(

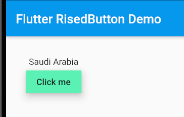
color: Colors.greenAccent,

onPressed: () => \_onPressed('Saudi Arabia'),

child: new Text('Click me'),

elevation: 10.0,

),



##### Tip 4

We can set a splash color when the button is pressed:

RaisedButton(

splashColor: Colors.yellow[200],

color: Colors.greenAccent,

onPressed: () => \_onPressed('Saudi Arabia'),

child: new Text('Click me'),

elevation: 10.0,

),

##### Tip 5

We can change the shape of the RaisedButton. For an example, we use RoundedRectangleBorder:

RaisedButton(

shape: RoundedRectangleBorder(

borderRadius: new BorderRadius.circular(18.0),

side: BorderSide(color: Colors.black),

),

splashColor: Colors.yellow[200],

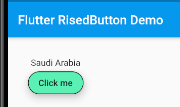
color: Colors.greenAccent,

onPressed: () => \_onPressed('Saudi Arabia'),

child: new Text('Click me'),

elevation: 10.0,

),



##### Tip 6

We can make square button using RoundedRectangleBorder and set 0.0 to borderRadius:

RaisedButton(

padding: EdgeInsets.all(

5.0), //distance of all the sides to the //child

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(0.0),

side: BorderSide(color: Colors.black)),

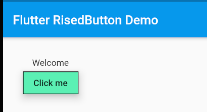
color: Colors.greenAccent,

onPressed: () => \_onPressed('Saudi Arabia'),

child: new Text('Click me'),

elevation: 10.0,

),



##### Tip 7

onHighlightChanged callback: Triggered when pressed or released the RaisedButton.

RaisedButton(

onHighlightChanged: (boolValue) => print(boolValue),

//returns true when pressed the button, false when released

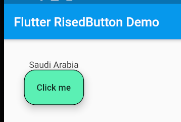
…

##### Tip 8

padding: Padding is used to set the distance between sides of the RaisedButton and it’s child.

RaisedButton(

padding: EdgeInsets.all(20.0),//distance of all the sides to the //child



In this paragraph, we learned how to use RaisedButton and how to change some of its properties.

<https://medium.com/flutter-community>

### FlatButton

We explore in this second flutter widget FlatButton. In Flutter mobile app, FlatButton is usually used to display buttons that lead to secondary functionalities of the application like viewing all files of Gallery, opening Camera, changing permissions etc.

We create another project:

flutter create --androidx -t app --org com.ah.book -a java -i swift flatbutton

The code of the main.dart file is given below:

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter FlatButton Demo'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

String \_value = 'Welcome';

void \_onPressed() {

setState((){

\_value = new DateTime.now().toString();

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text(widget.title),

),

body: Container(

padding: const EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(\_value),

SizedBox(height: 16.0,),

RaisedButton(

onPressed: \_onPressed,

child: new Text('RB Click me'),

),

SizedBox(height: 16.0,),

FlatButton(

onPressed: \_onPressed,

child: Text('FB Click me'),

)

],

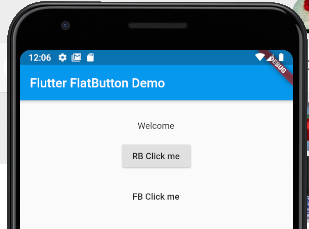
),

)),

);

}

}



As you can see, we have two buttons: a RaisedButton and a Flatbutton. By default, there is no color to the FlatButton and text is black. Also, FlatButton does not have an elevation unlike RaisedButton.

##### Tip 1

We can add color to the text and button using textColor and color respectively.

FlatButton(

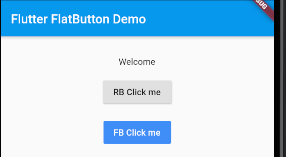
color: Colors.blueAccent,

textColor: Colors.white,

onPressed: \_onPressed,

child: Text('FB Click me'),

),



In this paragraph, we learned how to use FlatButton and how to change some of its properties.

### IconButton

We move to another interested button: the Iconbutton. It acts just like a button, but with an icon instead of an usual button.

flutter create --androidx -t app --org com.ah.book -a java -i swift iconbutton

The code of the main.dart file is given below:

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

int \_value = 0;

void \_add() {

setState(() {

\_value++;

});

}

void \_subtract() {

setState(() {

\_value--;

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text(widget.title),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text('Value = ${\_value}'),

SizedBox(height: 16.0,),

IconButton(icon: new Icon(Icons.add), onPressed: \_add),

IconButton(icon: new Icon(Icons.remove), onPressed: \_subtract)

],

),

)),

);

}

}



You can change many properties like size, color, background shape/color, etc.



Change Background of IconButton

You can change the background of IconButton by wrapping it around Ink widget as shown below.

Ink(

decoration: const ShapeDecoration(

color: Colors.lightBlue,

shape: CircleBorder(),

),

child: IconButton(

icon: Icon(

Icons.local\_airport,

),

iconSize: 100,

onPressed: () {},

),

),



#### تلميح Tooltips

الزر الأيقونة عبارة عن صورة جامدة. فيمكن إضافة نص عندما يمر المستخدم إصبعه فوق الأيقونة ويضغط عليها. وهي تسمى تلميح ToolTip.

class \_MyHomePageState extends State<MyHomePage> {

String \_value = 'Nothing Yet';

void \_onPressed() => setState(() => \_value = DateTime.now().toString());

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Tooltip'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(\_value),

IconButton(

icon: Icon(Icons.timer),

onPressed: \_onPressed,

tooltip: 'Click me',

)

],

),

),

),

);

}

}

للعلم، يوجد أداة اسمها تلميح Tooltip widget يمكن إضافتها لأي عنصر آخر لتعرض النص.

Tooltip(message: "DateTime", child: new Text(\_value)),

### MaterialButton

MaterialButton(

minWidth: 200.0,

height: 42.0,

onPressed: () {},

color: Colors.lightBlueAccent,

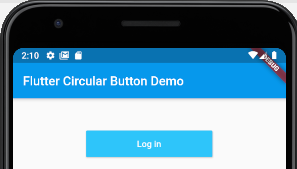
child: Text(

'Log in',

style: TextStyle(color: Colors.white),

),

),



If we want to set rounded border to my MaterialButton, we need to Warp the button with Material Widget.

Material(

//Wrap with Material

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(22.0)),

elevation: 18.0,

color: Color(0xFF801E48),

clipBehavior: Clip.antiAlias, // Add This

child: MaterialButton(

minWidth: 200.0,

height: 35,

color: Color(0xFF801E48),

child: new Text('Sign up',

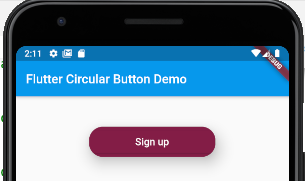
style:

new TextStyle(fontSize: 16.0, color: Colors.white)),

onPressed: () {},

),

),



If we wrap MaterialButton with ClipRRect(), it will looks like:

ClipRRect(

borderRadius: BorderRadius.all(Radius.circular(30.0)),

child: MaterialButton(

minWidth: 200.0,

height: 35,

color: Color(0xFF801E48),

child: new Text('Sign up',

style:

new TextStyle(fontSize: 16.0, color: Colors.white)),

onPressed: () {},

),

),



You can find more documentation on ClipRRect() flowing this link:

<https://api.flutter.dev/flutter/widgets/ClipRRect-class.html>

<https://kodestat.gitbook.io/flutter/20-flutter-iconbutton>

### الأزرار التبديلية Toggle Buttons

الأزرار التبديلية أو ToggleButtons هي مجموعة من الأزرار المتلاصقة تستعمل في عدة مجالات.

نقوم بإنشاء مشروع جديد عن طريق تنفيذ الأمر التالي:

flutter create --androidx -t app --org com.ah.book -a java -i swift togglebuttons

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

List<bool> \_selections = List.generate(3, (\_) => false);

@override

Widget build(BuildContext context) {

return MaterialApp(

home: Scaffold(

appBar: AppBar(

title: Center(child: Text('ToggleButtons')),

),

body: Column(

children: [

Padding(

padding: const EdgeInsets.only(top:16.0),

child: Center(

child: ToggleButtons(

children: <Widget>[

Icon(Icons.camera),

Icon(Icons.airline\_seat\_individual\_suite),

Icon(Icons.add\_location),

],

isSelected: \_selections,

onPressed: (int index) {

setState(() {

\_selections[index] = !\_selections[index];

});

},

),

),

),

],

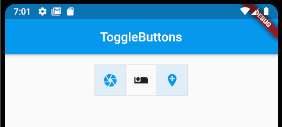
),

),

);

}

}



### Quiz

Question 1:

What does setState do?

* Sets the application as a state
* Clears the state
* Allows you to store variables in the current state
* Modifies variables

Question 2:

What does an Icon Button display?

* Text
* Icons
* Emails
* URLs

Question 3:

What is a Flat Button

* A button what visually appears flat
* A button without any color
* A raised button without any style
* A custom widget

Question 4:

What is the correct syntax for SetState

* () SetState
* SetState()
* setState((){})
* setState{}

### Exercise

1. If you are developing a mobile app in Flutter. How can you add a rounded button?

There are many ways of doing it. I am listing few here.

1) Using RoundedRectangleBorder

RaisedButton(

shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(20)),

onPressed: () {},

child: Text("Button"),

)

(2) Using ClipRRect

ClipOval(

child: RaisedButton(

onPressed: () {},

child: Text("Button"),

),

)

(4) Using ButtonTheme

ButtonTheme(

shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(20)),

child: RaisedButton(

onPressed: () {},

child: Text("Button"),

),

)

(5) Using StadiumBorder

RaisedButton(

shape: StadiumBorder(),

onPressed: () {},

child: Text("Button"),

)

2. Circular Button

Center(

child: SizedBox.fromSize(

size: Size(80, 80), // button width and height

child: ClipOval(

child: Material(

color: Colors.pink[300], // button color

child: InkWell(

splashColor: Colors.yellow, // splash color

onTap: () {}, // button pressed

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Icon(Icons.linked\_camera), // icon

Text("Picture"), // text

],

),

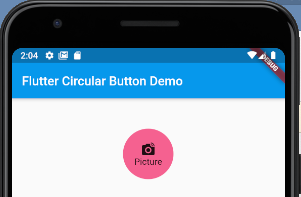
),

),

),

),

),



# أدوات الإدخال Input Widgets

## نظرة عامة

بعد أن تعرفنا إلى مختلف أنواع الأزرار المستعملة في الفلاتر، ننتقل الآن إلى أدوات الإدخال التي تمكن المستخدم من إدخال البيانات مثل الاسم وكلمة المرور إلى غير ذلك من البيانات. سنستعرض الأدوات التالية: TextField، TextformField،

## حقل النص TextField

هذه الأداة تستعمل غالبا لإدخال بيانات النص مثل كتابة الاسم وكلمة المرور. فلاتر يعتمد على TextField و TextFormField.

في هذه الفقرة، سندرس أداة TextField. ثم في الفقرة الموالية، سندرس أداة TextFormField.

لكن قبل ذلك سنتعرض فيما يلي إلى الفرق بين هذين الأداتين:

للحصول على إدخال المستخدم TextField كافية. لكن، إذا كنت تقوم بعمل نموذج Form بحيث تحتاج إلى حفظ العمليات save أو إعادة تعيينها reset أو التحقق من صحتها validate، فاستخدم TextFormField.

إذن، TextField هو حقل النص الأساسي بدون استعمال النموذج.

flutter create --androidx -t app --org com.ah.book -a java -i swift textfield

ملف main.dart يحتوي على الكود التالي:

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

String \_value = '';

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Name here'),

),

body: new Container(

padding: new EdgeInsets.all(32.0),

child: new Center(

child: new Column(

children: <Widget>[

Text(\_value),

TextField(

decoration: new InputDecoration(

labelText: 'Welcome ',

hintText: 'Hint',

icon: Icon(Icons.people)),

autocorrect: true,

autofocus: true,

keyboardType: TextInputType.text,

onChanged: (value) {

setState(() => \_value = 'Change: ${value}');

},

onSubmitted: (value) {

setState(() => \_value = 'Submit: ${value}');

},

)

],

),

)),

);

}

}

يمكن تحسين الكود عن طريق إضافة دالتين:

void \_onChange(String value) {

setState(() => \_value = 'Change: ${value}');

}

void \_onSubmit(String value) {

setState(() => \_value = 'Submit: ${value}');

}

فيصبح الكود كالآتي:

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

String \_value = '';

void \_onChange(String value) {

setState(() => \_value = 'Change: ${value}');

}

void \_onSubmit(String value) {

setState(() => \_value = 'Submit: ${value}');

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Name here'),

),

body: new Container(

padding: new EdgeInsets.all(32.0),

child: new Center(

child: new Column(

children: <Widget>[

Text(\_value),

TextField(

decoration: new InputDecoration(

labelText: 'Welcome ',

hintText: 'Hint',

icon: Icon(Icons.people)

),

autocorrect: true,

autofocus: true,

keyboardType: TextInputType.text,

onChanged: \_onChange,

onSubmitted: \_onSubmit,

)

],

),

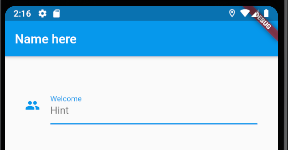
)

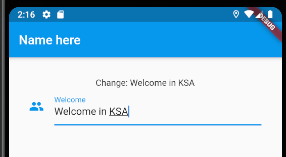
),

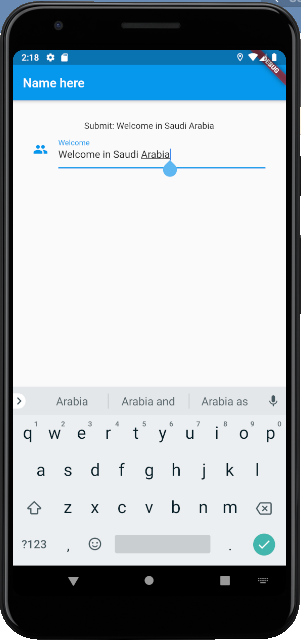
);

}

}







يستدعي حقل النص TextField رد الاتصالcallback onChanged عندما يقوم المستخدم بتغيير النص في الحقل. إذا أشار المستخدم إلى أنه انتهى من الكتابة في الحقل (على سبيل المثال، عن طريق الضغط على زر على لوحة المفاتيح الإلكترونية)، فإن حقل النص يستدعي رد الاتصال onSubmitted callback.

### استعمال TextEditingController

في هذه الفقرة، نتعرف على كيفية استرداد النص الذي أدخله المستخدم في حقل النص باستخدام TextEditingController و ذلك باتباع الخطوات التالية:

1. إنشاء TextEditingController.
2. إلحاق TextEditingController إلى TextField.
3. عرض القيمة الحالية لحقل النص.

هذه الخطوات مجموعة في الكود التالي:

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

// Create a text controller and use it to retrieve the current value

// of the TextField.

final myController = TextEditingController();

@override

void dispose() {

// Clean up the controller when the widget is disposed.

myController.dispose();

super.dispose();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Retrieve Text Input'),

),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: TextField(

controller: myController,

),

),

floatingActionButton: FloatingActionButton(

// When the user presses the button, show an alert dialog containing

// the text that the user has entered into the text field.

onPressed: () {

return showDialog(

context: context,

builder: (context) {

return AlertDialog(

// Retrieve the text the that user has entered by using the

// TextEditingController.

content: Text(myController.text),

);

},

);

},

tooltip: 'Show me the value!',

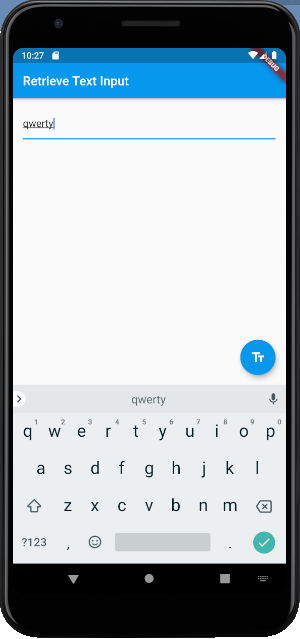
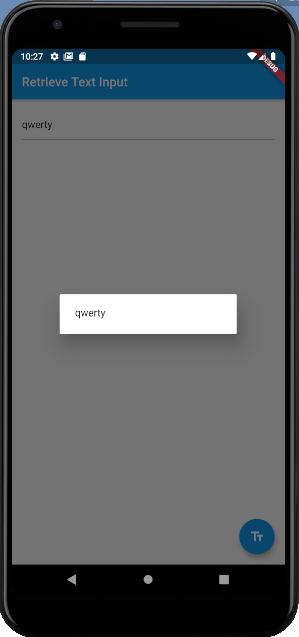
child: Icon(Icons.text\_fields),

),

);

}

}

### استعمال AutoFocus و FocusNode

بعض واجهات الجوال تعتمد على عدة حقول نص. ومن المناسب عندما تظهر هذه الواجهة أن يكون النص الأول يحتوي على بداية الكتابة ثم إذا انتهى المستخدم وضغط على Enter في لوحة المفاتيح أن ينتقل مباشرة إلى الحقل الثاني الموالي.

فيما يلي سنستعرض مثالين لهذا الاستعمال:

#### المثال الأول

ملف main.dart يحتوي على الكود التالي المتكون من حقلين نص.

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

FocusNode \_focusNode;

@override

void initState() {

\_focusNode = FocusNode();

super.initState();

}

@override

void dispose() {

\_focusNode.dispose();

super.dispose();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Focus Node'),

),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(

children: [

TextField(

autofocus: true,

onEditingComplete: () => \_focusNode.requestFocus(),

),

SizedBox(height: 16.0,),

TextField(

focusNode: \_focusNode,

),

],

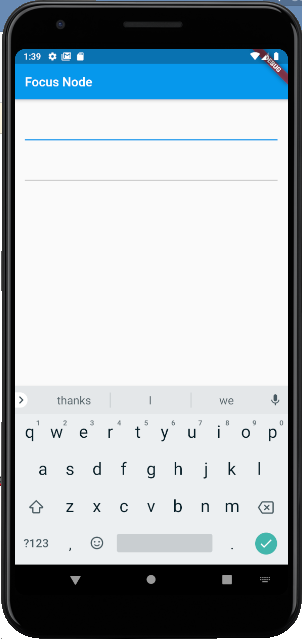
),

),

);

}

}



في هذا المثال، عندنا متغير \_focusNode نقوم بإنشائه في الدالة initState() ونقوم بحذفه في الدالة dispose().

هذا المتغير نربطه بالحقل الثاني.

في الحقل الأول، نقوم بإعطاء الخاصية autofocus يساوي true وذلك لكي يأخذ النص بداية الكتابة عند ظهور الواجهة.

والخاصية onEditingComplete نعطيها الكود \_focusNode.requestFocus() الذي من خلاله يتمكن الحقل الثاني من أخذ بداية الكتابة عندما يضغط المستخدم على Enter من الحقل الأول.

بهذه الطريقة يتم التنقل من الحقل الأول إلى الحقل الثاني.

#### المثال الثاني

ملف main.dart يحتوي على ثلاث حقول نص:

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('AutoFocus'),

),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(

children: [

TextField(

autofocus: true,

onEditingComplete: () => FocusScope.of(context).nextFocus(),

),

SizedBox(height: 16.0,),

TextField(

onEditingComplete: () => FocusScope.of(context).nextFocus(),

),

SizedBox(height: 16.0,),

TextField(

onEditingComplete: () => FocusScope.of(context).unfocus(),

),

],

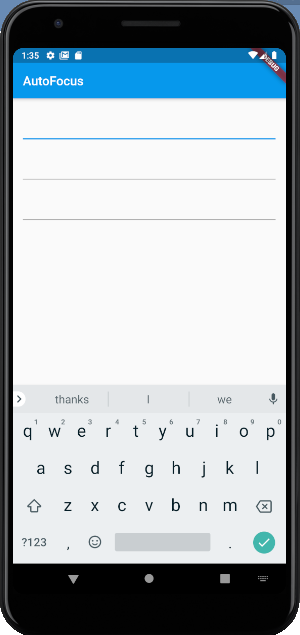
),

),

);

}

}



في البداية عند ظهور الواجهة، الحقل الأول يحتوي على بداية الكتابة (ما يسمى ال focus). وكل مرة يضغط المستخدم على Enter، ينتقل إلي الحقل الموالي وذلك باستعمال FocusScope.of(context).nextFocus()

في الحقل الأخير ينفذ الأمر FocusScope.of(context).unfocus() و ذلك لتختفي لوحة المفاتيح وينتهي ال focus.

هنا يمكن مثلا تنفيذ أوامر أخرى مثل تسجيل البيانات والانتقال إلى واجهة أخرى.

### إفادة

keyboardType: TextInputType.number,

keyboardType: TextInputType.emailAddress,

keyboardType: TextInputType.phone,

keyboardType: TextInputType.url,

### روابط مفيدة

<https://flutter.dev/docs/cookbook/forms/retrieve-input>

<https://flutter.dev/docs/cookbook/forms/focus>

## حقل النص TextFormField

هذه الأداة تشبه الأداة السابقة لكنها تُستعمل داخل النموذج Form ولديها ميزات خاصة تحتاج إلى حفظ العمليات save أو إعادة تعيينها reset أو التحقق من صحتها validate.

غالبا نحتاج إلى الخطوات التالية:

* إنشاء نموذج باستخدام GlobalKey
* إضافة TextFormField مع كيفية التحقق من صحة البيانات
* إنشاء زر للتحقق من النموذج وإرساله.

لإنشاء المشروع، نقوم بتنفيذ الأمر التالي:

flutter create --androidx -t app --org com.ah.book -a java -i swift textformfield

ملف الدارت يتكون من الكود:

import 'package:flutter/material.dart';

import 'package:fluttertoast/fluttertoast.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

final \_formKey = GlobalKey<FormState>();

String name = '';

@override

Widget build(BuildContext context) {

// Build a Form widget using the \_formKey created above.

return Scaffold(

appBar: AppBar(title: Text('Form')),

body: Form(

key: \_formKey,

child: Column(

children: <Widget>[

// Add TextFormFields and RaisedButton here.

TextFormField(

decoration: const InputDecoration(

icon: Icon(Icons.person),

hintText: 'What do people call you?',

labelText: 'Name \*',

),

onChanged: (text) {

name = text;

},

onSaved: (String value) {

// This optional block of code can be used to run

// code when the user saves the form.

},

// The validator receives the text that the user has entered.

validator: (value) {

if (value.isEmpty) {

return 'Please enter some text.';

}

if (value.contains('@')) {

return 'Do not use the @ char.';

}

return null;

},

),

SizedBox(

height: 16.0,

),

RaisedButton(

onPressed: () {

final \_form = \_formKey.currentState;

// Validate returns true if the form is valid, otherwise false.

if (\_form.validate()) {

// If the form is valid, display a toast message. In the real world,

// you'd often call a server or save the information in a database.

\_form.save();

Fluttertoast.showToast(

msg: "Processing Data: " + name,

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.CENTER,

timeInSecForIosWeb: 1,

backgroundColor: Colors.red,

textColor: Colors.white,

fontSize: 16.0);

}

},

child: Text('Submit'),

),

],

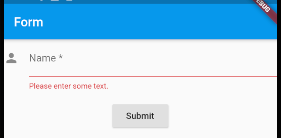
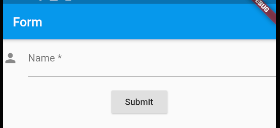
),

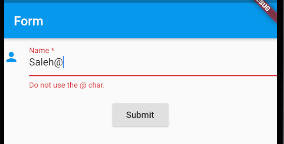
),

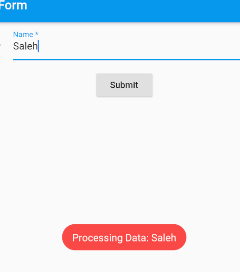
);

}

}





نحتاج إلى معرف موحد Unique Identifier لاستعمال النموذج. يتيح لنا GlobalKey إنشاء معرف موحد على مستوى التطبيق يمكننا ربطه بنموذجنا:

final \_formKey = GlobalKey<FormState>();

body: Form(

key: \_formKey,

للتحقق من صحة البيانات المدخلة، نستعمل "المدقق" validator.

validator: (value) {

if (value.isEmpty) {

return 'Please enter some text.';

}

if (value.contains('@')) {

return 'Do not use the @ char.';

}

return null;

},

في الأخير، نضيف زر يؤدي إلى:

* التحقق من صحة النموذج
* إذا كانت البيانات صحيحة، نقوم بحفظ البيانات مثلا وتشغيل جميع وظائف onSaved في كل أداة من النموذج.

في المثال السابق، نقوم بعرض البيانات في رسالة تظهر على الشاشة.

RaisedButton(

onPressed: () {

final \_form = \_formKey.currentState;

// Validate returns true if the form is valid, otherwise false.

if (\_form.validate()) {

// If the form is valid, display a toast message. In the real world,

// you'd often call a server or save the information in a database.

\_form.save();

Fluttertoast.showToast(

msg: "Processing Data: " + name,

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.CENTER,

timeInSecForIosWeb: 1,

backgroundColor: Colors.red,

textColor: Colors.white,

fontSize: 16.0);

}

},

child: Text('Submit'),

),

يمكن إضافة زر لإعادة النموذج من جديد وذلك باستعمال الأمر \_form.reset();

Row(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

RaisedButton(

onPressed: () {

final \_form = \_formKey.currentState;

// Validate returns true if the form is valid, otherwise false.

if (\_form.validate()) {

// If the form is valid, display a toast message. In the real world,

// you'd often call a server or save the information in a database.

//\_form.save();

Fluttertoast.showToast(

msg: "Processing Data: " + name,

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.CENTER,

timeInSecForIosWeb: 1,

backgroundColor: Colors.red,

textColor: Colors.white,

fontSize: 16.0);

}

},

child: Text('Submit'),

),

SizedBox(

width: 16.0,

),

RaisedButton(

onPressed: () {

final \_form = \_formKey.currentState;

\_form.reset();

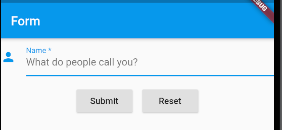
},

child: Text('Reset'),

),

],

),



### مثال 1 واجهة تسجيل الدخول LoginForm

كثير من تطبيقات الجوال تعتمد على تسجيل دخول المستخدم من خلال إدخال اسم المستخدم وكلمة المرور.

فيما يلي نستعرض كود لواجهة تحتوي على نموذج الدخول.

flutter create --androidx -t app --org com.ah.book -a java -i swift loginform

import 'package:flutter/material.dart';

import 'package:flutter/services.dart';

final kHintTextStyle = TextStyle(

color: Colors.white54,

fontFamily: 'OpenSans',

);

final kLabelStyle = TextStyle(

color: Colors.white,

fontWeight: FontWeight.bold,

fontFamily: 'OpenSans',

);

final kBoxDecorationStyle = BoxDecoration(

color: Color(0xFF6CA8F1),

borderRadius: BorderRadius.circular(10.0),

boxShadow: [

BoxShadow(

color: Colors.black12,

blurRadius: 6.0,

offset: Offset(0, 2),

),

],

);

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: LoginPage(),

);

}

}

class LoginPage extends StatefulWidget {

@override

\_LoginPageState createState() => new \_LoginPageState();

// State<StatefulWidget> createState() {

// return \_LoginPageState();

// }

}

class \_LoginPageState extends State<LoginPage> {

final scaffoldKey = new GlobalKey<ScaffoldState>();

final formKey = new GlobalKey<FormState>();

// FormType \_formType = FormType.login;

String \_userName = "";

String \_password = "";

void validateAndSave() {

final form = formKey.currentState;

if (form.validate()) {

form.save();

performLogin();

print(\_userName + ' - ' + \_password);

}

}

void performLogin() {

final snackbar = new SnackBar(

content: new Text("Username : $\_userName, password : $\_password"),

);

scaffoldKey.currentState.showSnackBar(snackbar);

}

Widget \_buildUserNameTF() {

return Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: <Widget>[

Text(

'Username',

style: kLabelStyle,

),

SizedBox(height: 10.0),

Container(

alignment: Alignment.centerLeft,

decoration: kBoxDecorationStyle,

height: 60.0,

child: TextFormField(

keyboardType: TextInputType.text,

autovalidate: false,

style: TextStyle(

color: Colors.white,

fontFamily: 'OpenSans',

),

decoration: InputDecoration(

border: InputBorder.none,

contentPadding: EdgeInsets.only(top: 14.0),

prefixIcon: Icon(

Icons.account\_circle,

color: Colors.white,

),

hintText: 'Enter your Username',

hintStyle: kHintTextStyle,

),

validator: (value) {

return value.isEmpty ? 'Username is Required.' : null;

},

onSaved: (value) {

return \_userName = value;

},

),

),

],

);

}

Widget \_buildPasswordTF() {

return Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: <Widget>[

Text(

'Password',

style: kLabelStyle,

),

SizedBox(height: 10.0),

Container(

alignment: Alignment.centerLeft,

decoration: kBoxDecorationStyle,

height: 60.0,

child: TextFormField(

obscureText: true,

autovalidate: false,

style: TextStyle(

color: Colors.white,

fontFamily: 'OpenSans',

),

decoration: InputDecoration(

border: InputBorder.none,

contentPadding: EdgeInsets.only(top: 14.0),

prefixIcon: Icon(

Icons.lock,

color: Colors.white,

),

hintText: 'Enter your Password',

hintStyle: kHintTextStyle,

),

validator: (String value) {

if (value.isEmpty) {

return 'Password is Required.';

}

if (value.length < 6) {

return 'Password too short.';

}

return null;

// return value.isEmpty ? 'Password is Required.' : null;

// || value.length < 6 ? 'Password too short' : null;

},

onSaved: (String value) {

return \_password = value;

},

// validator: (val) =>

// val.length < 6 ? 'Password too short' : null,

// onSaved: (val) => \_password = val,

),

),

],

);

}

Widget \_buildLoginBtn() {

return Container(

padding: EdgeInsets.symmetric(vertical: 25.0),

width: double.infinity,

child: Form(

child: RaisedButton(

elevation: 5.0,

onPressed: () {

validateAndSave();

},

// => print('Login Button Pressed'),

padding: EdgeInsets.all(15.0),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(30.0),

),

color: Colors.white,

child: Text(

'LOGIN',

style: TextStyle(

color: Color(0xFF527DAA),

letterSpacing: 1.5,

fontSize: 18.0,

fontWeight: FontWeight.bold,

fontFamily: 'OpenSans',

),

),

),

),

);

}

@override

Widget build(BuildContext context) {

return Scaffold(

key: scaffoldKey,

body: AnnotatedRegion<SystemUiOverlayStyle>(

value: SystemUiOverlayStyle.light,

child: GestureDetector(

onTap: () => FocusScope.of(context).unfocus(),

child: Stack(

children: <Widget>[

Container(

height: double.infinity,

width: double.infinity,

decoration: BoxDecoration(

gradient: LinearGradient(

begin: Alignment.topCenter,

end: Alignment.bottomCenter,

colors: [

Color(0xFF73AEF5),

Color(0xFF61A4F1),

Color(0xFF478DE0),

Color(0xFF398AE5),

],

stops: [0.1, 0.4, 0.7, 0.9],

),

),

),

Container(

height: double.infinity,

child: SingleChildScrollView(

physics: AlwaysScrollableScrollPhysics(),

padding: EdgeInsets.symmetric(

horizontal: 40.0,

vertical: 120.0,

),

child: Form(

key: formKey,

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

'Sign In',

style: TextStyle(

color: Colors.white,

fontFamily: 'OpenSans',

fontSize: 30.0,

fontWeight: FontWeight.bold,

),

),

SizedBox(height: 30.0),

\_buildUserNameTF(),

SizedBox(

height: 30.0,

),

\_buildPasswordTF(),

\_buildLoginBtn(),

],

),

),

),

)

],

),

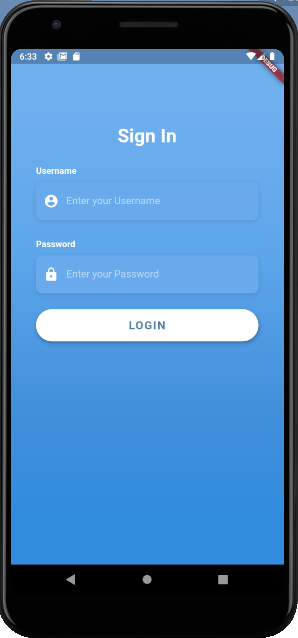
),

),

);

}

}



### مثال 2 واجهة تسجيل الدخول LoginScreen

نستعرض فيما يلي واجهة أخرى لتسجيل الدخول. تحتوي على:

* حقل نص لكتابة اسم المستخدم
* حقل نص لكتابة كلمة المرور
* FlatButton في حالة نسيان كلمة المرور
* RaisedButton للدخول
* FlatButton لتسجيل مستخدم جديد

لإنشاء مشروع جديد، ننفذ الأمر التالي:

flutter create --androidx -t app --org com.ah.book -a java -i swift loginscreen

كود الواجهة في الملف التالي:

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

TextEditingController nameController = TextEditingController();

TextEditingController passwordController = TextEditingController();

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Sample App: Login Screen'),

),

body: Padding(

padding: EdgeInsets.all(10),

child: Column(

children: <Widget>[

Container(

alignment: Alignment.center,

padding: EdgeInsets.all(10),

child: Text(

'Flutter Book',

style: TextStyle(

color: Colors.blue,

fontWeight: FontWeight.w500,

fontSize: 30),

)),

Container(

alignment: Alignment.center,

padding: EdgeInsets.all(10),

child: Text(

'Sign in',

style: TextStyle(fontSize: 20),

)),

Container(

padding: EdgeInsets.all(10),

child: TextField(

controller: nameController,

decoration: InputDecoration(

border: OutlineInputBorder(),

labelText: 'Username',

),

),

),

Container(

padding: EdgeInsets.fromLTRB(10, 10, 10, 0),

child: TextField(

obscureText: true,

controller: passwordController,

decoration: InputDecoration(

border: OutlineInputBorder(),

labelText: 'Password',

),

),

),

FlatButton(

onPressed: () {

//forgot password screen

},

textColor: Colors.blue,

child: Text('Forgot Password'),

),

Container(

height: 50,

width: double.infinity,

padding: EdgeInsets.fromLTRB(10, 0, 10, 0),

child: RaisedButton(

textColor: Colors.white,

color: Colors.blue,

child: Text('Login'),

onPressed: () {

print(nameController.text);

print(passwordController.text);

},

)),

Container(

child: Row(

children: <Widget>[

Text('Does not have account?'),

FlatButton(

textColor: Colors.blue,

child: Text(

'Sign in',

style: TextStyle(fontSize: 20),

),

onPressed: () {

//signup screen

},

)

],

mainAxisAlignment: MainAxisAlignment.center,

),

),

],

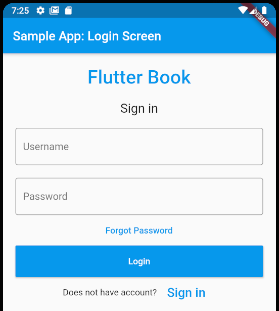
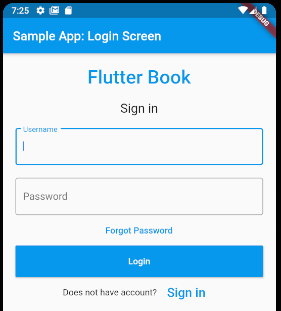
),

),

);

}

}

### روابط مفيدة

<https://flutter.dev/docs/cookbook/forms>

<https://flutter.dev/docs/cookbook/forms/validation>

## خانة الاختيار CheckBox

flutter create --androidx -t app --org com.ah.book -a java -i swift checkbox

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

bool \_value1 = false;

bool \_value2 = false;

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Checkbox'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text('\_value1: ${\_value1}'),

Checkbox(

value: \_value1,

onChanged: (bool value) => setState(() => \_value1 = value),

),

CheckboxListTile(

value: \_value2,

onChanged: (bool value) => setState(() => \_value2 = value),

title: new Text('Faculty of KSA'),

controlAffinity: ListTileControlAffinity.leading,

subtitle: new Text('Student'),

secondary: new Icon(Icons.archive),

activeColor: Colors.red,

),

Text('\_value2: ${\_value2}'),

],

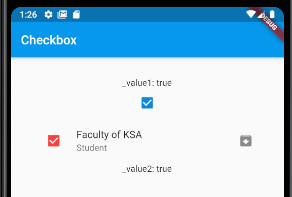
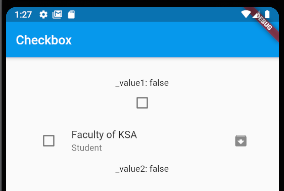
),

)),

);

}

}

### مثال واجهة تسجيل الدخول

الهدف من هذا المثال:

* تعلم ترتيب الواجهة
* إدخال النص جانب الخيار في تغيير القيمة

class \_MyHomePageState extends State<MyHomePage> {

bool \_rememberMeFlag = false;

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Checkbox'),

),

body: Container(

color: Colors.black45,

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Container(

margin: EdgeInsets.symmetric(vertical: 5.0, horizontal: 35.0),

color: Colors.white70,

height: 50.0,

),

Container(

margin: EdgeInsets.symmetric(vertical: 5.0, horizontal: 35.0),

color: Colors.white70,

height: 50.0,

),

Container(

margin: EdgeInsets.symmetric(vertical: 5.0, horizontal: 20.0),

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: <Widget>[

Row(

children: <Widget>[

GestureDetector(

child: Row(

children: <Widget>[

Checkbox(

value: \_rememberMeFlag,

onChanged: (value) => setState(() {

\_rememberMeFlag = !\_rememberMeFlag;

}),

),

Text(

"Remember me",

style: TextStyle(color: Colors.white70),

)

],

),

onTap: () => setState(() {

\_rememberMeFlag = !\_rememberMeFlag;

}),

),

],

),

Container(

margin: EdgeInsets.only(right: 15.0),

child: Text(

"Forgot password ?",

style: TextStyle(color: Colors.white70),

),

)

],

)),

Container(

margin: EdgeInsets.symmetric(vertical: 5.0, horizontal: 35.0),

color: Colors.orange,

height: 50.0,

),

],

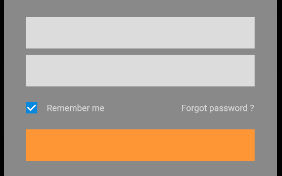
),

),

);

}

}



هنا، مجرد الضغط على نص Remember me يغير قيمة الخيار.

وتم ذلك عن طريق إضافة GestureDetector

GestureDetector(

child: Row(

children: <Widget>[

Checkbox(

value: \_rememberMeFlag,

onChanged: (value) => setState(() {

\_rememberMeFlag = !\_rememberMeFlag;

}),

),

Text(

"Remember me",

style: TextStyle(color: Colors.white70),

)

],

),

onTap: () => setState(() {

\_rememberMeFlag = !\_rememberMeFlag;

}),

),

### مجموعة خانات الاختيار CheckboxGroup

تمتاز منصة فلاتر بوجود مكتبات وأكواد جاهزة لإضافتها في التطبيق. هذه المكتبات تم برمجتها من عدة مبرمجين حول العالم وهي تسهل الأمور. مثلا قد تحتاج إلى خدمة معينة، فعوض أن تبذل جهد ووقت إضافي، ممكن تجد الخدمة تم برمجتها من طرف آخر ووضعها متاحة للجميع للاستخدام في تطبيقاتهم.

المكتبات التي يمكن تحميلها موجودة بالأساس في هذا الموقع: <https://pub.dev> الذي يقدم خدمة البحث والتحميل حسب ما هو مبين في الصورة التالية:



نستعرض فيما يلي مكتبة: grouped\_buttons وهي مكتبة تسهل تجميع خانات الاختيار:

لتثبيت هذه المكتبة، لابد من إضافة السطر التالي في ملف: pubspec.yaml

dependencies:

grouped\_buttons: ^1.0.4

ثم تنفيذ الأمر:

**$** flutter pub get

بعد ذلك إضافة السطر في أعلي صفحة ملف الدارت:

**import** 'package:grouped\_buttons/grouped\_buttons.dart';

هذه المكتبة تحتوي على عنصر CheckboxGroup يُمكّن من إضافة مجموعة من خانات الاختيار كما هو مبين في الكود التالي:

import 'package:flutter/material.dart';

import 'package:grouped\_buttons/grouped\_buttons.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: HomePage(),

);

}

}

class HomePage extends StatefulWidget {

@override

\_HomePageState createState() => \_HomePageState();

}

class \_HomePageState extends State<HomePage> {

List<String> \_checked = ["A", "B"];

@override

Widget build(BuildContext context){

return Scaffold(

appBar: AppBar(

title: Text("Grouped Buttons Example"),

),

body: \_body(),

);

//

}

Widget \_body(){

return ListView(

children: <Widget>[

//--------------------

//SIMPLE USAGE EXAMPLE

//--------------------

//BASIC CHECKBOXGROUP

Container(

padding: const EdgeInsets.only(left: 14.0, top: 14.0),

child: Text("Basic CheckboxGroup",

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: 20.0

),

),

),

CheckboxGroup(

labels: <String>[

"Sunday",

"Monday",

"Tuesday",

"Wednesday",

"Thursday",

"Friday",

"Saturday",

],

disabled: [

"Wednesday",

"Friday"

],

onChange: (bool isChecked, String label, int index) => print("isChecked: $isChecked label: $label index: $index"),

onSelected: (List<String> checked) => print("checked: ${checked.toString()}"),

),

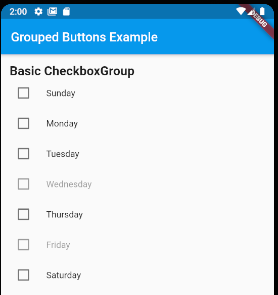
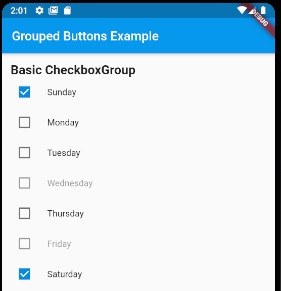
]

);

}

}

تشغيل التطبيق في الجوال الافتراضي يعطي هذه الواجهة:

عند الضغط على يوم الأحد، يطبع النص التالي:

I/flutter ( 9662): isChecked: true label: Sunday index: 0

I/flutter ( 9662): checked: [Sunday]

ثم عند الضغط على يوم السبت، يطبع النص التالي:

I/flutter ( 9662): isChecked: true label: Saturday index: 6

I/flutter ( 9662): checked: [Sunday, Saturday]

### روابط مفيدة

<https://api.flutter.dev/flutter/material/Checkbox-class.html>

## زر الراديو Radio Widget

يستعمل زر الراديو مثلا في الخيارات المتعددة بحيث يستطيع المستخدم أن يختار الإجابة الصحيحة للسؤال.

flutter create --androidx -t app --org com.ah.book -a java -i swift radiobutton

يأخذ عنصر زر الراديو ثلاث معلمات parameters: value، groupValue و onChanged

تأخذ value رقم الترتيب. مثلا إذا عندنا ثلاث أزرار على التوالي، ستكون قيمة كل منها: 0 و 1 و 2

يمثل groupValue القيمة المختارة لمجموعة الأزرار بأكملها.

و onChanged هو callback الذي يستدعى وينفذ في كل مرة حصل تغيير في الزر.

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

int \_value1 = 0;

void \_setValue1(int value) => setState(() => \_value1 = value);

Widget makeRadios() {

List<Widget> list = List<Widget>();

for (int i = 0; i < 3; i++) {

list.add(Radio(

value: i,

groupValue: \_value1,

onChanged: \_setValue1,

));

}

Column column = Column(

children: list,

);

return column;

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Radio buttons'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(

'${\_value1}',

style: new TextStyle(

color: new Color.fromARGB(255, 117, 117, 117),

fontSize: 25.0,

fontWeight: FontWeight.bold),

),

makeRadios(),

],

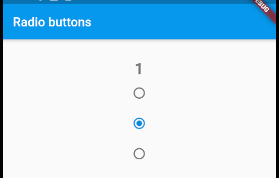
),

)),

);

}

}



في هذا المثال، أضفنا ثلاث أزرار راديو باستعمال التكرار loop for

وفي كل مرة نضغط على زر معين، يستعرض لنا رقم الترتيب: 0 أو 1 أو 2

هناك طريقة ثانية تتمثل في استعمال RadioListTile:

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage2(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage2 extends StatefulWidget {

MyHomePage2({Key key, this.title}) : super(key: key);

final String title;

@override

\_MyHomePage2State createState() => \_MyHomePage2State();

}

class \_MyHomePage2State extends State<MyHomePage2> {

int \_value2 = 0;

void \_setValue2(int value) => setState(() => \_value2 = value);

Widget makeRadioTiles() {

List<Widget> list = List<Widget>();

for (int i = 0; i < 3; i++) {

list.add(RadioListTile(

value: i,

groupValue: \_value2,

onChanged: \_setValue2,

activeColor: Colors.green,

controlAffinity: ListTileControlAffinity.trailing,

title: Text('Item: ${i}'),

subtitle: Text('sub title'),

));

}

Column column = Column(

children: list,

);

return column;

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Radio buttons'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(

'${\_value2}',

style: new TextStyle(

color: new Color.fromARGB(255, 117, 117, 117),

fontSize: 25.0,

fontWeight: FontWeight.bold),

),

makeRadioTiles(),

],

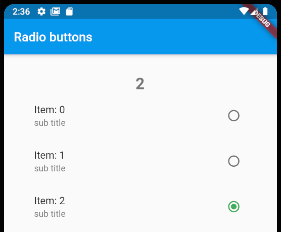
),

)),

);

}

}



### مثال لأسئلة اختبار

import 'package:flutter/material.dart';

import 'package:fluttertoast/fluttertoast.dart';

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

class MyApp extends StatefulWidget {

@override

\_MyAppState createState() => new \_MyAppState();

}

class \_MyAppState extends State<MyApp> {

int \_radioValue1 = -1;

int correctScore = 0;

int \_radioValue2 = -1;

int \_radioValue3 = -1;

void \_handleRadioValueChange1(int value) {

setState(() {

\_radioValue1 = value;

switch (\_radioValue1) {

case 0:

Fluttertoast.showToast(

msg: 'Correct !',

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.TOP,

timeInSecForIosWeb: 1,

backgroundColor: Colors.green,

textColor: Colors.white,

fontSize: 16.0);

correctScore++;

break;

case 1:

Fluttertoast.showToast(

msg: 'Try again !',

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.TOP,

timeInSecForIosWeb: 1,

backgroundColor: Colors.red,

textColor: Colors.white,

fontSize: 16.0);

break;

case 2:

Fluttertoast.showToast(

msg: 'Try again !',

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.TOP,

timeInSecForIosWeb: 1,

backgroundColor: Colors.red,

textColor: Colors.white,

fontSize: 16.0);

break;

}

});

}

void \_handleRadioValueChange2(int value) {

setState(() {

\_radioValue2 = value;

switch (\_radioValue2) {

case 0:

Fluttertoast.showToast(

msg: 'Try again !',

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.TOP,

timeInSecForIosWeb: 1,

backgroundColor: Colors.red,

textColor: Colors.white,

fontSize: 16.0);

break;

case 1:

Fluttertoast.showToast(

msg: 'Correct !',

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.TOP,

timeInSecForIosWeb: 1,

backgroundColor: Colors.green,

textColor: Colors.white,

fontSize: 16.0);

correctScore++;

break;

case 2:

Fluttertoast.showToast(

msg: 'Try again !',

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.TOP,

timeInSecForIosWeb: 1,

backgroundColor: Colors.red,

textColor: Colors.white,

fontSize: 16.0);

break;

}

});

}

void \_handleRadioValueChange3(int value) {

setState(() {

\_radioValue3 = value;

switch (\_radioValue3) {

case 0:

Fluttertoast.showToast(

msg: 'Try again !',

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.TOP,

timeInSecForIosWeb: 1,

backgroundColor: Colors.red,

textColor: Colors.white,

fontSize: 16.0);

break;

case 1:

Fluttertoast.showToast(

msg: 'Correct !',

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.TOP,

timeInSecForIosWeb: 1,

backgroundColor: Colors.green,

textColor: Colors.white,

fontSize: 16.0);

correctScore++;

break;

case 2:

Fluttertoast.showToast(

msg: 'Try again !',

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.TOP,

timeInSecForIosWeb: 1,

backgroundColor: Colors.red,

textColor: Colors.white,

fontSize: 16.0);

break;

}

});

}

@override

Widget build(BuildContext context) {

return new MaterialApp(

home: new Scaffold(

appBar: AppBar(

title: new Text('Kids Quiz App'),

centerTitle: true,

backgroundColor: Colors.blue,

),

body: new Container(

padding: EdgeInsets.all(8.0),

child: new Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

new Text(

'Select correct answers from below:',

style:

new TextStyle(fontSize: 20.0, fontWeight: FontWeight.bold),

),

new Padding(

padding: new EdgeInsets.all(8.0),

),

new Divider(height: 5.0, color: Colors.black),

new Padding(

padding: new EdgeInsets.all(8.0),

),

new Text(

'Lion is :',

style: new TextStyle(

fontWeight: FontWeight.bold,

fontSize: 18.0,

),

),

new Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

new Radio(

value: 0,

groupValue: \_radioValue1,

onChanged: \_handleRadioValueChange1,

),

new Text(

'Carnivore',

style: new TextStyle(fontSize: 16.0),

),

new Radio(

value: 1,

groupValue: \_radioValue1,

onChanged: \_handleRadioValueChange1,

),

new Text(

'Herbivore',

style: new TextStyle(

fontSize: 16.0,

),

),

new Radio(

value: 2,

groupValue: \_radioValue1,

onChanged: \_handleRadioValueChange1,

),

new Text(

'Omnivore',

style: new TextStyle(fontSize: 16.0),

),

],

),

new Divider(

height: 5.0,

color: Colors.black,

),

new Padding(

padding: new EdgeInsets.all(8.0),

),

new Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

new Text(

'Giraffe is :',

style: new TextStyle(

fontWeight: FontWeight.bold,

fontSize: 18.0,

),

),

new Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

new Radio(

value: 0,

groupValue: \_radioValue2,

onChanged: \_handleRadioValueChange2,

),

new Text(

'Carnivore',

style: new TextStyle(fontSize: 16.0),

),

new Radio(

value: 1,

groupValue: \_radioValue2,

onChanged: \_handleRadioValueChange2,

),

new Text(

'Herbivore',

style: new TextStyle(fontSize: 16.0),

),

new Radio(

value: 2,

groupValue: \_radioValue2,

onChanged: \_handleRadioValueChange2,

),

new Text(

'Omnivore',

style: new TextStyle(fontSize: 16.0),

),

],

),

new Divider(

height: 5.0,

color: Colors.black,

),

new Padding(

padding: new EdgeInsets.all(8.0),

),

new Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

new Text(

'Elephant is :',

style: new TextStyle(

fontWeight: FontWeight.bold,

fontSize: 18.0,

),

),

new Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

new Radio(

value: 0,

groupValue: \_radioValue3,

onChanged: \_handleRadioValueChange3,

),

new Text(

'Carnivore',

style: new TextStyle(fontSize: 16.0),

),

new Radio(

value: 1,

groupValue: \_radioValue3,

onChanged: \_handleRadioValueChange3,

),

new Text(

'Herbivore',

style: new TextStyle(fontSize: 16.0),

),

new Radio(

value: 2,

groupValue: \_radioValue3,

onChanged: \_handleRadioValueChange3,

),

new Text(

'Omnivore',

style: new TextStyle(fontSize: 16.0),

),

],

),

new Divider(

height: 5.0,

color: Colors.black,

),

new Padding(

padding: new EdgeInsets.all(8.0),

),

new RaisedButton(

onPressed: validateAnswers,

child: new Text(

'Check Final Score',

style: new TextStyle(

fontSize: 16.0,

fontWeight: FontWeight.normal,

color: Colors.white),

),

color: Theme.of(context).accentColor,

shape: new RoundedRectangleBorder(

borderRadius: new BorderRadius.circular(20.0)),

),

new Padding(

padding: EdgeInsets.all(4.0),

),

new RaisedButton(

onPressed: resetSelection,

child: new Text(

'Reset Selection',

style: new TextStyle(

fontWeight: FontWeight.normal,

fontSize: 16.0,

color: Colors.white),

),

color: Theme.of(context).accentColor,

shape: new RoundedRectangleBorder(

borderRadius: new BorderRadius.circular(20.0)),

)

],

),

],

)

],

),

),

),

);

}

void resetSelection() {

\_handleRadioValueChange1(-1);

\_handleRadioValueChange2(-1);

\_handleRadioValueChange3(-1);

correctScore = 0;

}

void validateAnswers() {

if (\_radioValue1 == -1 && \_radioValue2 == -1 && \_radioValue3 == -1) {

Fluttertoast.showToast(

msg: 'Please select at least one answer',

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.TOP,

timeInSecForIosWeb: 1,

backgroundColor: Colors.red,

textColor: Colors.white,

fontSize: 16.0);

} else {

Fluttertoast.showToast(

msg: 'Your total score is: $correctScore out of 3',

toastLength: Toast.LENGTH\_SHORT,

gravity: ToastGravity.TOP,

timeInSecForIosWeb: 1,

backgroundColor: Colors.green,

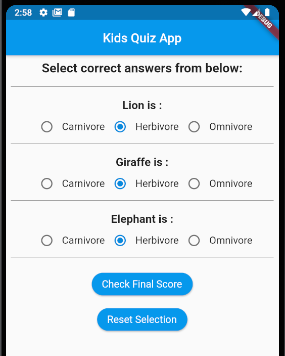
textColor: Colors.white,

fontSize: 16.0);

}

}

}



### روابط مفيدة

<https://api.flutter.dev/flutter/material/Radio-class.html>

## زر المفتاح Switch Widget

flutter create --androidx -t app --org com.ah.book -a java -i swift switchbutton

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

bool \_value1 = false;

bool \_value2 = false;

void \_onChanged1(bool value) => setState(() => \_value1 = value);

void \_onChanged2(bool value) => setState(() => \_value2 = value);

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Switch Button'),

),

body: Container(

padding: const EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(

'${\_value1}',

style: TextStyle(

fontSize: 20.0,

fontWeight: FontWeight.bold,

color: Colors.blue),

),

Switch(

value: \_value1,

onChanged: (bool value) => setState(() => \_value1 = value),

activeTrackColor: Colors.lightGreenAccent,

activeColor: Colors.green,

),

Divider(

color: Colors.teal.shade100,

thickness: 3.0,

),

Text(

'${\_value2}',

style: TextStyle(

fontSize: 20.0,

fontWeight: FontWeight.bold,

color: Colors.blue),

),

SwitchListTile(

value: \_value2,

onChanged: \_onChanged2,

title: Text(

'Are you student?',

style: TextStyle(

fontWeight: FontWeight.bold, color: Colors.blue),

),

)

],

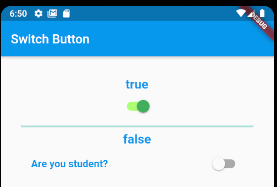
),

)),

);

}

}



## المنزلق Slider Widget

flutter create --androidx -t app --org com.ah.book -a java -i swift slider

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

double \_value = 0.0;

//void \_setvalue(double newvalue) => setState(() => \_value = newvalue);

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Slider'),

),

body: Container(

padding: const EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(

'Value: ${(\_value \* 100).round()}',

style: TextStyle(

fontSize: 20.0,

color: Colors.black87,

),

),

Slider(

value: \_value,

onChanged: (double newvalue) =>

setState(() => \_value = newvalue),

//Slider(value: \_value, onChanged: \_setvalue),

),

],

),

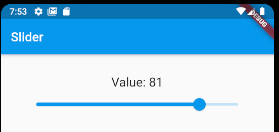
),

),

);

}

}



ممكن إضافة ألوان وهيئات أخرى عن طريق إضافة SliderTheme:

SliderTheme(

data: SliderTheme.of(context).copyWith(

activeTrackColor: Colors.red[700],

inactiveTrackColor: Colors.red[100],

trackShape: RectangularSliderTrackShape(),

trackHeight: 4.0,

thumbColor: Colors.redAccent,

thumbShape: RoundSliderThumbShape(enabledThumbRadius: 12.0),

overlayColor: Colors.red.withAlpha(32),

overlayShape: RoundSliderOverlayShape(overlayRadius: 28.0),

),

child: Slider(

value: \_value,

onChanged: (value) {

setState(() {

\_value = value;

});

},

),

),



وتعديل آخر:

SliderTheme(

data: SliderTheme.of(context).copyWith(

activeTrackColor: Colors.red[700],

inactiveTrackColor: Colors.red[100],

trackShape: RoundedRectSliderTrackShape(),

trackHeight: 4.0,

thumbShape: RoundSliderThumbShape(enabledThumbRadius: 12.0),

thumbColor: Colors.redAccent,

overlayColor: Colors.red.withAlpha(32),

overlayShape: RoundSliderOverlayShape(overlayRadius: 28.0),

tickMarkShape: RoundSliderTickMarkShape(),

activeTickMarkColor: Colors.red[700],

inactiveTickMarkColor: Colors.red[100],

valueIndicatorShape: PaddleSliderValueIndicatorShape(),

valueIndicatorColor: Colors.redAccent,

valueIndicatorTextStyle: TextStyle(

color: Colors.white,

),

),

child: Slider(

value: \_value,

min: 0,

max: 100,

divisions: 10,

label: '$\_value',

onChanged: (value) {

setState(

() {

\_value = value;

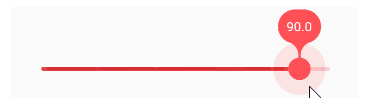
},

);

},

),

),



### روابط مفيدة

<https://api.flutter.dev/flutter/material/Slider-class.html>

## تحديد التاريخ DatePicker

flutter create --androidx -t app --org com.ah.book -a java -i swift datepicker

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

String \_value = '';

Future \_selectDate() async {

DateTime picked = await showDatePicker(

context: context,

initialDate: new DateTime.now(),

firstDate: new DateTime(2016),

lastDate: new DateTime(2099),

);

if (picked != null) setState(() => \_value = picked.toString());

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Date Picker'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(

\_value,

style: TextStyle(

fontSize: 20.0,

color: Colors.black87,

),

),

RaisedButton(

onPressed: \_selectDate,

child: new Text('Show DatePicker'),

)

],

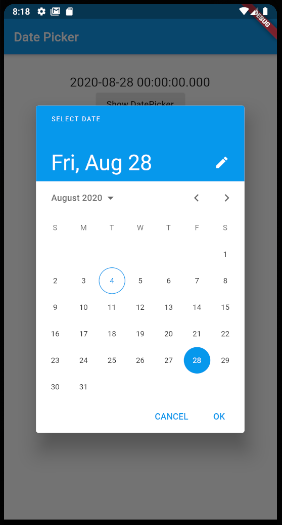
),

)),

);

}

}



## تحديد الوقت TimePicker

flutter create --androidx -t app --org com.ah.book -a java -i swift timepicker

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

TimeOfDay selectedTime = TimeOfDay.now();

Future \_selectDate() async {

TimeOfDay picked\_s = await showTimePicker(

context: context,

initialTime: selectedTime,

builder: (BuildContext context, Widget child) {

return MediaQuery(

data: MediaQuery.of(context).copyWith(alwaysUse24HourFormat: false),

child: child,

);

});

if (picked\_s != null && picked\_s != selectedTime)

setState(() {

selectedTime = picked\_s;

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Date Picker'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(

("Current Time: ${selectedTime.format(context)}"),

style: TextStyle(

fontSize: 20.0,

color: Colors.black87,

),

),

RaisedButton(

onPressed: \_selectDate,

child: new Text('Show TimePicker'),

)

],

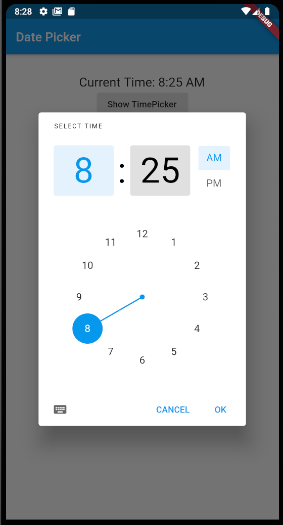
),

)),

);

}

}



### CupertinoDatePicker

يوجد كلاس فلاتر تسمى CupertinoDatePicker

import 'package:flutter/cupertino.dart';

Container(

height: 200,

child: CupertinoDatePicker(

mode: CupertinoDatePickerMode.date,

initialDateTime: DateTime(2020, 1, 1),

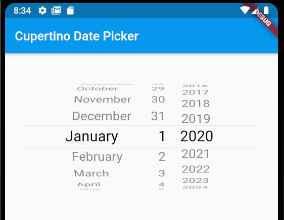
onDateTimeChanged: (DateTime newDateTime) {

// Do something

},

),

),



إضافة الوقت:

Container(

height: 200,

child: CupertinoDatePicker(

mode: CupertinoDatePickerMode.dateAndTime,

initialDateTime: DateTime(2020, 1, 1, 10, 33),

onDateTimeChanged: (DateTime newDateTime) {

//Do Some thing

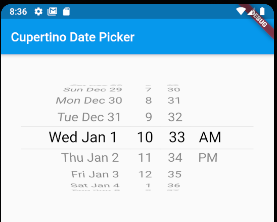
},

use24hFormat: false,

minuteInterval: 1,

),

),



### مكتبة flutter\_datetime\_picker

dependencies:

flutter:

sdk: flutter

flutter\_datetime\_picker:

git:

url: https://github.com/ditheshthegreat/flutter\_datetime\_picker # Repo

ref: patch-1 # Branch name

jiffy: ^3.0.1

import 'package:flutter/material.dart';

import 'package:flutter/cupertino.dart';

import 'package:flutter\_datetime\_picker/flutter\_datetime\_picker.dart';

import 'package:jiffy/jiffy.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

DateTime dateTime;

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Date Picker'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Container(

height: 50,

child: Padding(

padding: const EdgeInsets.only(bottom: 5.0),

child: FlatButton(

materialTapTargetSize: MaterialTapTargetSize.shrinkWrap,

color: Colors.blueAccent,

textColor: Colors.white,

onPressed: () {

DatePicker.showDateTimePicker(context,

showTitleActions: true,

minTime: DateTime(2020, 1, 1),

maxTime: DateTime(2040, 12, 31),

theme: DatePickerTheme(

headerColor: Colors.orange,

backgroundColor: Colors.blue,

itemStyle: TextStyle(

color: Colors.white,

fontWeight: FontWeight.bold,

fontSize: 18),

doneStyle: TextStyle(

color: Colors.white,

fontSize: 16)), onChanged: (date) {

//print('2change $date in time zone ' +

// date.timeZoneOffset.inHours.toString());

}, onConfirm: (date) {

//print('2confirm $date');

setState(() {

dateTime = date;

});

},

currentTime: DateTime.now(),

locale: LocaleType.ar);

},

child: Text(

Jiffy(dateTime).yMMMMEEEEdjm,

//style: TextStyle(color: Colors.blue),

)),

),

),

],

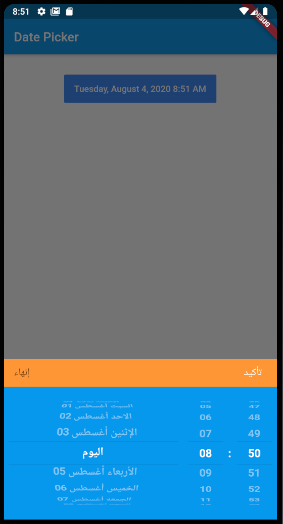
),

)),

);

}

}



## أسئلة وتمارين

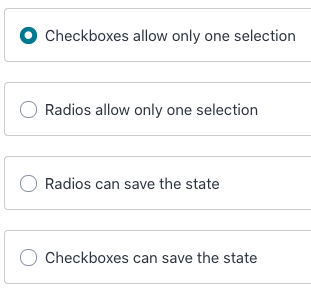
Add a Text widget, a Textfield widget, and a RaisedButton widget - allow the user to enter text and when they click the button, the Text will display what they wrote in the TextField

Questions for this assignment

User Input allows the user to?

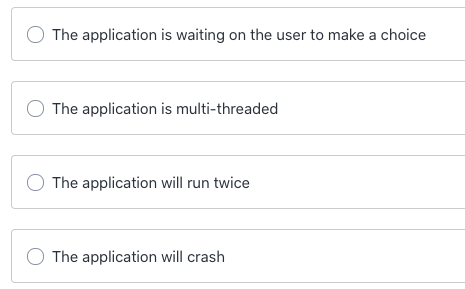
Question 1:

Whats the difference between a Radio and a Checkbox



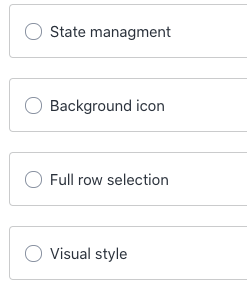
Question 2:

Why do we have to use async programming to display a Date picker?



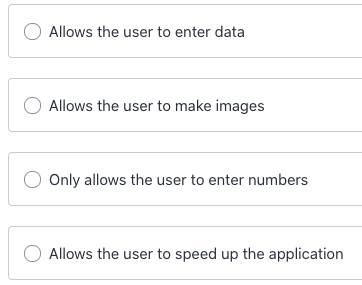
Question 3:

What is the difference between a Combobox and a Switch



Question 4:

What does a Textfield do?



# السقالة Scaffold

## شريط التطبيقات AppBar

يوجد شريط التطبيقات في أعلى واجهة تطبيق الجوال. في هذا المثال، يحتوي الشريط على عنوان Title وأزرار إجراءات Actions.

ويمكن تغيير لون الخلفية عن طريق backgroundColor

flutter create –androidx -t app –org com.ah.book -a java -i swift appbar

import ‘package:flutter/material.dart’;

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: ‘Flutter Demo’,

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: ‘Flutter Demo Home Page’),

debugShowCheckedModeBanner: false,

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

int \_value = 0;

void \_add() => setState(() => \_value++);

void \_remove() => setState(() => \_value--);

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text(‘Plus – Minus’),

backgroundColor: Colors.red,

actions: <Widget>[

IconButton(icon: Icon(Icons.add), onPressed: \_add),

IconButton(icon: Icon(Icons.remove), onPressed: \_remove)

],

),

body: Container(

padding: const EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(

\_value.toString(),

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: 40.0,

),

)

],

),

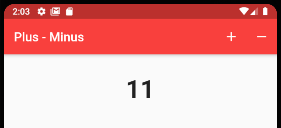
),

),

);

}

}



### القائمة المنسدلة DropDown Menu

يمكن إضافة القائمة المنسدلة إلى شريط التطبيقات.

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

debugShowCheckedModeBanner: false,

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

Choice \_selectedChoice = choices[0]; // The app's "state".

void \_select(Choice choice) {

// Causes the app to rebuild with the new \_selectedChoice.

setState(() {

\_selectedChoice = choice;

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('DropDown Menu'),

backgroundColor: Colors.green,

actions: <Widget>[

// overflow menu

PopupMenuButton<Choice>(

onSelected: \_select,

itemBuilder: (BuildContext context) {

return choices.skip(2).map((Choice choice) {

return PopupMenuItem<Choice>(

value: choice,

child: Row(

children: [

IconButton(

icon: Icon(

choice.icon,

color: Colors.blue,

),

onPressed: () {},

),

Text(choice.title),

],

),

);

}).toList();

},

),

],

),

body: Container(

padding: const EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(

\_selectedChoice.title,

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: 40.0,

),

)

],

),

),

),

);

}

}

class Choice {

const Choice({this.title, this.icon});

final String title;

final IconData icon;

}

const List<Choice> choices = const <Choice>[

const Choice(title: 'Car', icon: Icons.directions\_car),

const Choice(title: 'Bicycle', icon: Icons.directions\_bike),

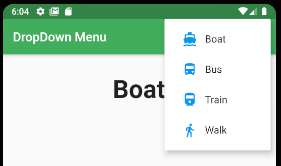
const Choice(title: 'Boat', icon: Icons.directions\_boat),

const Choice(title: 'Bus', icon: Icons.directions\_bus),

const Choice(title: 'Train', icon: Icons.directions\_railway),

const Choice(title: 'Walk', icon: Icons.directions\_walk),

];



### شريط التطبيقات يشبه Facebook AppBar

النقاط المتشابهة بين هذين التطبيقين هي:

استخدم اللون الأبيض ل AppBar

استخدم شعار الصورة كعنوان

استخدم صورهم المصممة بدلاً من IconButton الأساسية للإجراءات

Scaffold(

appBar: AppBar(

backgroundColor: Colors.white,

title: Row(

mainAxisAlignment: MainAxisAlignment.start,

children: <Widget>[

Image.asset(

'assets/images/facebook\_icon.png',

fit: BoxFit.cover,

height: 35.0,

),

],

),

actions: <Widget>[

IconButton(

padding: EdgeInsets.all(5.0),

icon: Image.asset('assets/images/facebook\_search.png'),

onPressed: () {

// Implement navigation to shopping cart page here...

print('Click Search');

},

),

IconButton(

padding: EdgeInsets.all(5.0),

icon: Image.asset('assets/images/facebook\_msg.png'),

onPressed: () {

// Implement navigation to shopping cart page here...

print('Click Message');

},

),

],

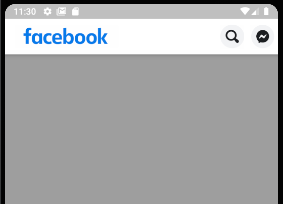
),

body: Container(

decoration: BoxDecoration(color: Colors.grey),

),

),



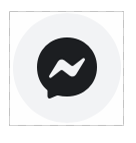
facebook\_icon.png



facebook\_search.png



facebook\_msg.png



إضافة صورة

Image.asset(

'assets/images/facebook\_icon.png',

fit: BoxFit.cover,

height: 35.0,

),

إضافة زر يحتوي على صورة

IconButton(

padding: EdgeInsets.all(5.0),

icon: Image.asset('assets/images/facebook\_search.png'),

onPressed: () {

// Implement navigation to shopping cart page here...

print('Click Search');

},

),

### روابط مفيدة

<https://api.flutter.dev/flutter/material/AppBar-class.html>

<https://flutter.dev/docs/catalog/samples/basic-app-bar>

## زر الإجراء العائم Floating Action Button

flutter create --androidx -t app --org com.ah.book -a java -i swift flotingactionbutton

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

String \_value = '';

void \_onClicked() => setState(() => \_value = new DateTime.now().toString());

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Floting Action Button'),

),

floatingActionButton: FloatingActionButton(

onPressed: \_onClicked,

backgroundColor: Colors.red,

mini: false,

child: Icon(Icons.timer),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[Text(\_value)],

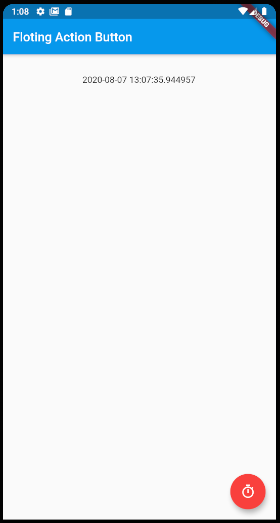
),

)),

);

}

}



### تغيير مكان الزر

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Floting Action Button'),

),

floatingActionButton: FloatingActionButton(

onPressed: \_onClicked,

backgroundColor: Colors.red,

mini: false,

child: Icon(Icons.timer),

),

floatingActionButtonLocation: FloatingActionButtonLocation.centerFloat,

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[Text(\_value)],

),

)),

);

}



### روابط مفيدة

<https://api.flutter.dev/flutter/material/FloatingActionButton-class.html>

<https://pub.dev/packages/fab_circular_menu>

## الدرج Drawer

flutter create --androidx -t app --org com.ah.book -a java -i swift drawer

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Drawer'),

),

drawer: Drawer(

child: Container(

padding: EdgeInsets.all(32.0),

child: Column(

children: <Widget>[

Text('App Drawer'),

RaisedButton(

onPressed: () => Navigator.pop(context),

child: Text('Close'),

),

],

),

),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[Text('Add Widgets Here')],

),

)),

);

}

}

### درج أكثر واقعية

drawer: Drawer(

child: ListView(

padding: EdgeInsets.zero,

children: const <Widget>[

DrawerHeader(

decoration: BoxDecoration(

color: Colors.blue,

),

child: Text(

'Drawer Header',

style: TextStyle(

color: Colors.white,

fontSize: 24,

),

),

),

ListTile(

leading: Icon(Icons.message),

title: Text('Messages'),

),

ListTile(

leading: Icon(Icons.account\_circle),

title: Text('Profile'),

),

ListTile(

leading: Icon(Icons.settings),

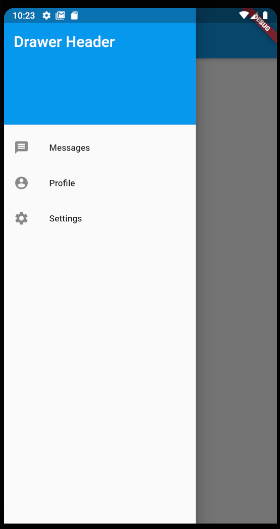
title: Text('Settings'),

),

],

),

),



### روابط مفيدة

<https://flutter.dev/docs/cookbook/design/drawer>

<https://api.flutter.dev/flutter/material/Drawer-class.html>

## أزرار التذييل Footer Buttons

أزرار التذييل هي عبارة عن مجموعة من الأزرار التي يتم عرضها أسفل السقالة Scaffold أي أسفل واجهة الجوال.

flutter create --androidx -t app --org com.ah.book -a java -i swift footerbuttons

### مثال أول

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

String \_value = '';

void \_onClick(String value) => setState(() => \_value = value);

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Footer Buttons'),

),

persistentFooterButtons: <Widget>[

IconButton(

icon: Icon(Icons.timer),

onPressed: () => \_onClick('Button 1'),

),

IconButton(

icon: Icon(Icons.people),

onPressed: () => \_onClick('Button 2'),

),

IconButton(

icon: Icon(Icons.map),

onPressed: () => \_onClick('Button 3'),

),

],

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(

\_value,

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: 40.0,

),

),

],

),

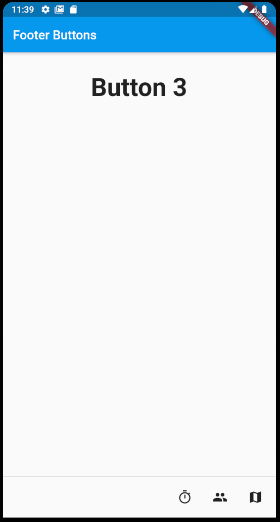
),

),

);

}

}



### مثال ثاني

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

String \_value = '';

void \_onClick(String value) => setState(() => \_value = value);

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Footer Buttons'),

),

persistentFooterButtons: <Widget>[

RaisedButton(

elevation: 10.0,

onPressed: () => \_onClick('btn 1'),

color: Colors.green,

child: Icon(

Icons.print,

color: Colors.white,

),

),

RaisedButton(

elevation: 10.0,

onPressed: () => \_onClick('btn 1'),

color: Colors.blueGrey,

child: Icon(

Icons.send,

color: Colors.white,

),

),

],

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(

\_value,

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: 40.0,

),

),

],

),

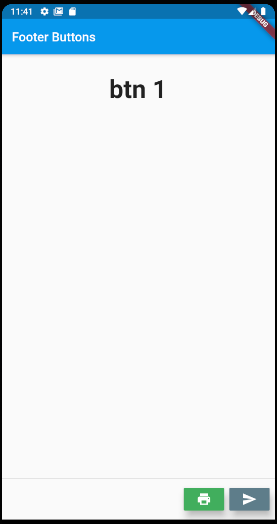
),

),

);

}

}



## شريط التنقل السفلي Button Navigation Bar

يتم استخدام Bottom Navigation Bar لعرض شريط تنقل أسفل السقالة.

flutter create --androidx -t app --org com.ah.book -a java -i swift bottomnavigationbar

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

List<BottomNavigationBarItem> \_items;

String \_value = '';

int \_index = 0;

@override

void initState() {

\_items = List();

\_items.add(BottomNavigationBarItem(

icon: Icon(Icons.home), title: Text('Home')));

\_items.add(BottomNavigationBarItem(

icon: Icon(Icons.search), title: Text('Search')));

\_items.add(BottomNavigationBarItem(

icon: Icon(Icons.add), title: Text('Add')));

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Bottom Navigation Bar'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(

\_value,

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: 30.0,

),

),

],

),

)),

bottomNavigationBar: BottomNavigationBar(

items: \_items,

fixedColor: Colors.blue,

currentIndex: \_index,

onTap: (int item) {

setState(() {

\_index = item;

\_value = "Current value is: ${\_index.toString()}";

});

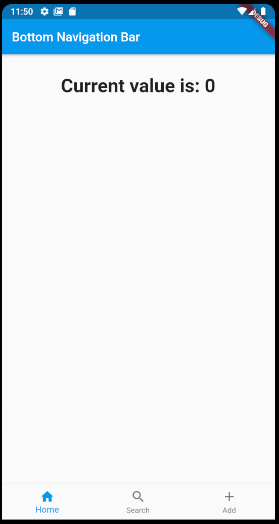
},

),

);

}

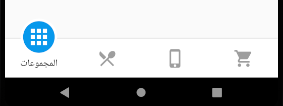
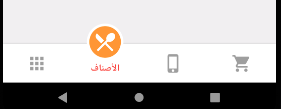
}



### مكتبة circular\_bottom\_navigation

يوجد في موقع المكتبات التابعة للفلاتر، مكتبة تسمى circular\_bottom\_navigation.

هذه المكتبة تعطي شريط تنقل سفلي متميز كما هو مبين في هذا المثال:

## لون خلفية السقالة backgroundColor

تحدد هذه الخاصية لون خلفية السقالة:

return Scaffold(

backgroundColor: Colors.white70,

appBar: AppBar(

title: Text('Bottom Navigation Bar'),

),

body:



## إضافة زر للعودة إلى الخلف

### الطريقة الأولى: تلقائيا

وذلك باستعمال

appBar: AppBar(

automaticallyImplyLeading: true,

//`true` if you want Flutter to automatically add Back Button when needed,

//or `false` if you want to force your own back button every where

title: Text('Bottom Navigation Bar'),

),

body:

### الطريقة الثانية: يدويا

appBar: AppBar(

title: Text('Bottom Navigation Bar'),

leading: IconButton(

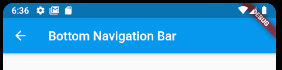
icon: Icon(Icons.arrow\_back, color: Colors.white),

onPressed: () => Navigator.pop(context, false),

),

),

body:



# التنبيهات

## Bottom Sheet

flutter create --androidx -t app --org com.ah.book -a java -i swift bottomsheet

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

void \_showBottom() {

showModalBottomSheet<void>(

context: context,

builder: (BuildContext context) {

return Container(

padding: EdgeInsets.all(15.0),

child: Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

'Some info here',

style:

TextStyle(color: Colors.red, fontWeight: FontWeight.bold),

),

IconButton(

icon: Icon(Icons.close),

onPressed: () => Navigator.pop(context),

),

],

),

);

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Bottom Sheet'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text('Add Widgets Here'),

RaisedButton(

onPressed: \_showBottom,

child: Text('Click me'),

),

],

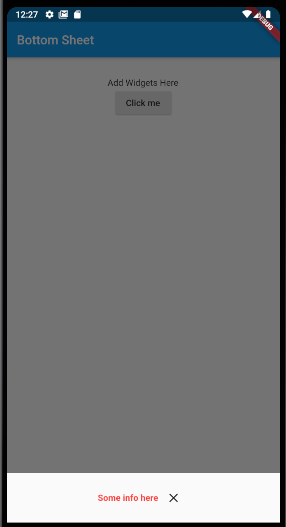
),

)),

);

}

}



### تنفيذ أوامر بعد غلق الإشعار

void \_showBottom() {

showModalBottomSheet<void>(

context: context,

builder: (BuildContext context) {

return Container(

padding: EdgeInsets.all(15.0),

child: Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

'Some info here',

style:

TextStyle(color: Colors.red, fontWeight: FontWeight.bold),

),

IconButton(

icon: Icon(Icons.close),

onPressed: () => Navigator.pop(context),

),

],

),

);

}).whenComplete(() {

print('Hey there, I\'m calling after hide bottomSheet');

});

}

I/flutter (10754): Hey there, I'm calling after hide bottomSheet

<https://api.flutter.dev/flutter/material/showModalBottomSheet.html>

## Snack Bar

flutter create --androidx -t app --org com.ah.book -a java -i swift snackbar

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

final GlobalKey<ScaffoldState> \_scaffoldstate = GlobalKey<ScaffoldState>();

void \_showbar() {

\_scaffoldstate.currentState.showSnackBar(SnackBar(content: Text('Waiting ...')));

}

@override

Widget build(BuildContext context) {

return Scaffold(

key: \_scaffoldstate,

appBar: AppBar(

title: Text('Snack Bar'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text('Add Widgets Here'),

RaisedButton(onPressed: \_showbar, child: Text('Click me'),)

],

),

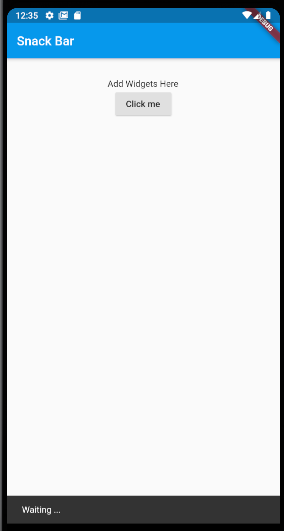
)

),

);

}

}



### مكتبة flushbar

<https://pub.dev/packages/flushbar>



## Alert Dialog

flutter create --androidx -t app --org com.ah.book -a java -i swift alertdialog

import 'dart:async';

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

Future \_showAlert(BuildContext context, String message) async {

return showDialog(

context: context,

child: AlertDialog(

title: Text(message),

actions: <Widget>[

FlatButton(

child: Text('Buy'),

onPressed: () {},

),

FlatButton(

onPressed: () => Navigator.pop(context),

child: Text('Cancel'),

),

],

));

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Alert Dialog'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text('Add Widgets Here'),

RaisedButton(

onPressed: () =>

\_showAlert(context, 'Continue purchase?'),

child: Text('Click me'),

)

],

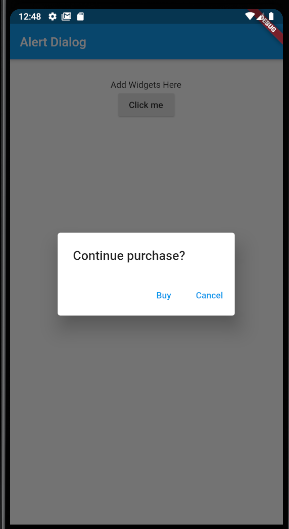
),

)),

);

}

}



## Simple Dialog

flutter create --androidx -t app --org com.ah.book -a java -i swift simpledialog

import 'dart:async';

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

enum Answers{YES,NO,MAYBE}

class \_MyHomePageState extends State<MyHomePage> {

String \_value = '';

void \_setValue(String value) => setState(() => \_value = value);

Future \_askUser() async {

switch(

await showDialog(

context: context,

child: SimpleDialog(

title: Text('Do you like Flutter?'),

children: <Widget>[

SimpleDialogOption(child: Text('Yes!!!'),onPressed: (){Navigator.pop(context, Answers.YES);},),

SimpleDialogOption(child: Text('NO :('),onPressed: (){Navigator.pop(context, Answers.NO);},),

SimpleDialogOption(child: Text('Maybe :|'),onPressed: (){Navigator.pop(context, Answers.MAYBE);},),

],

)

)

) {

case Answers.YES:

\_setValue('Yes');

break;

case Answers.NO:

\_setValue('No');

break;

case Answers.MAYBE:

\_setValue('Maybe');

break;

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Name here'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(\_value),

RaisedButton(onPressed: \_askUser, child: Text('Click me'),)

],

),

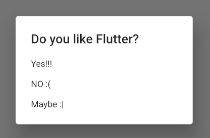
)

),

);

}

}



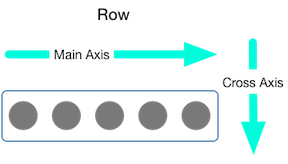
# ترتيب الواجهات Layouts

## العمود والصف Column and Row

العمود هو أداة واجهة Widget يُستعمل لترتيب عناصر واجهة آخرين بطريقة عمودية. على عكس الصف الذي يرتب العناصر بطريقة أفقية. العمود والصف لهما نفس الخصائص.

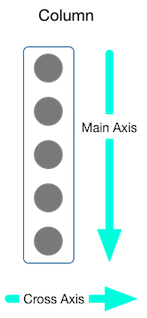
**For Row:**

mainAxisAlignment = Horizontal Axis  
crossAxisAlignment = Vertical Axis

[](https://i.stack.imgur.com/aypHr.png)

**For Column:**

mainAxisAlignment = Vertical Axis  
crossAxisAlignment = Horizontal Axis

[](https://i.stack.imgur.com/eseWF.png)

flutter create --androidx -t app --org com.ah.book -a java -i swift rowcolumn

### مثال ١

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Row'),

),

body: Row(

mainAxisAlignment: MainAxisAlignment.start,

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Container(color: Colors.blue, height: 50.0, width: 50.0),

Icon(Icons.local\_taxi, size: 50.0, color: Colors.pink),

Icon(Icons.card\_travel, size: 50.0, color: Colors.purple),

Container(color: Colors.orange, height: 50.0, width: 50.0),

Icon(Icons.local\_airport, size: 50.0, color: Colors.cyan),

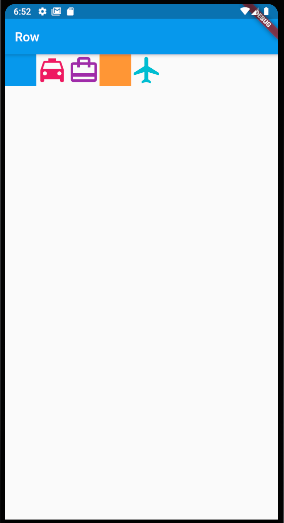
],

),

);

}

}



### مثال ٢

body: Row(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Container(color: Colors.blue, height: 50.0, width: 50.0),

Icon(Icons.local\_taxi, size: 50.0, color: Colors.pink),

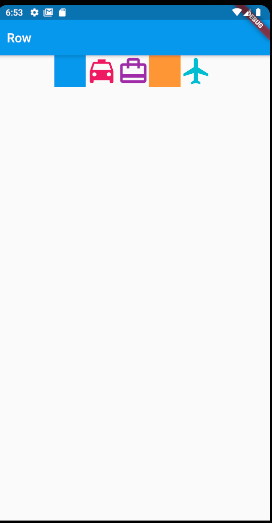
Icon(Icons.card\_travel, size: 50.0, color: Colors.purple),

Container(color: Colors.orange, height: 50.0, width: 50.0),

Icon(Icons.local\_airport, size: 50.0, color: Colors.cyan),

],

),



#### ملاحظة

لا يشغل الصف كل المساحة الرأسية المتاحة افتراضيًا. يأخذ فقط المساحة المطلوبة (لكنه يأخذ كل المساحة الأفقية).

ولذلك لو أنا غيرنا: crossAxisAlignment: CrossAxisAlignment.start

إلى crossAxisAlignment: CrossAxisAlignment.center

فإن الواجهة لا تتغير ولا تعطي النتيجة المتوقعة.

ولكن لو أننا نضيف SizedBox.expand:

### مثال ٣

body: SizedBox.expand(

child: Row(

mainAxisAlignment: MainAxisAlignment.start,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Container(color: Colors.blue, height: 50.0, width: 50.0),

Icon(Icons.local\_taxi, size: 50.0, color: Colors.pink),

Icon(Icons.card\_travel, size: 50.0, color: Colors.purple),

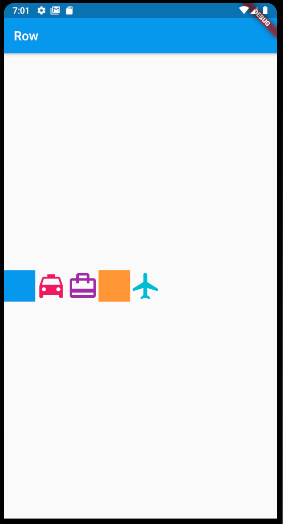
Container(color: Colors.orange, height: 50.0, width: 50.0),

Icon(Icons.local\_airport, size: 50.0, color: Colors.cyan),

],

),

),



### مثال ٤

وكذلك هذا الترتيب:

body: SizedBox.expand(

child: Row(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Container(color: Colors.blue, height: 50.0, width: 50.0),

Icon(Icons.local\_taxi, size: 50.0, color: Colors.pink),

Icon(Icons.card\_travel, size: 50.0, color: Colors.purple),

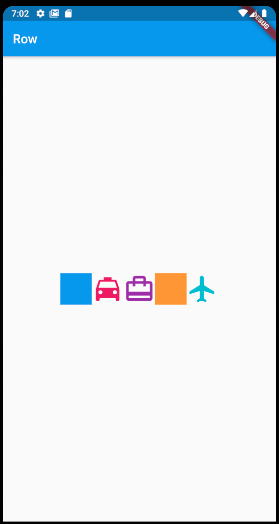
Container(color: Colors.orange, height: 50.0, width: 50.0),

Icon(Icons.local\_airport, size: 50.0, color: Colors.cyan),

],

),

),



### مثال ٥

body: Column(

mainAxisAlignment: MainAxisAlignment.start,

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Container(color: Colors.blue, height: 50.0, width: 50.0),

Icon(Icons.local\_taxi, size: 50.0, color: Colors.pink),

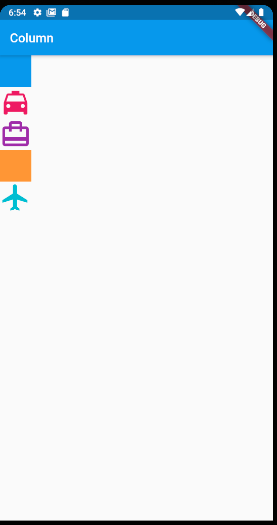
Icon(Icons.card\_travel, size: 50.0, color: Colors.purple),

Container(color: Colors.orange, height: 50.0, width: 50.0),

Icon(Icons.local\_airport, size: 50.0, color: Colors.cyan),

],

),



### مثال ٦

body: Column(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Container(color: Colors.blue, height: 50.0, width: 50.0),

Icon(Icons.local\_taxi, size: 50.0, color: Colors.pink),

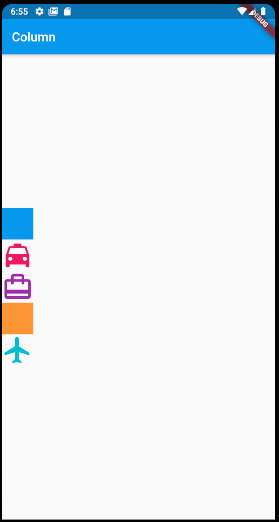
Icon(Icons.card\_travel, size: 50.0, color: Colors.purple),

Container(color: Colors.orange, height: 50.0, width: 50.0),

Icon(Icons.local\_airport, size: 50.0, color: Colors.cyan),

],

),



### mainAxisAlignment: MainAxisAlignment.spaceAround

تضع مساحات خالية بالتساوي بين العناصر الفرعية.

#### الصف

body: SizedBox.expand(

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceAround,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Container(color: Colors.blue, height: 50.0, width: 50.0),

Icon(Icons.local\_taxi, size: 50.0, color: Colors.pink),

Icon(Icons.card\_travel, size: 50.0, color: Colors.purple),

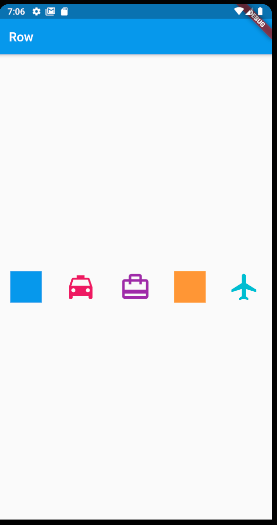
Container(color: Colors.orange, height: 50.0, width: 50.0),

Icon(Icons.local\_airport, size: 50.0, color: Colors.cyan),

],

),

),



#### العمود

body: SizedBox.expand(

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Container(color: Colors.blue, height: 50.0, width: 50.0),

Icon(Icons.local\_taxi, size: 50.0, color: Colors.pink),

Icon(Icons.card\_travel, size: 50.0, color: Colors.purple),

Container(color: Colors.orange, height: 50.0, width: 50.0),

Icon(Icons.local\_airport, size: 50.0, color: Colors.cyan),

],

),

),



### Expanded

#### مثال ١

body: SizedBox.expand(

child: Row(

mainAxisAlignment: MainAxisAlignment.start,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Expanded(child: Container(color: Colors.blue, height: 50.0, width: 50.0)),

Icon(Icons.local\_taxi, size: 50.0, color: Colors.pink),

Icon(Icons.card\_travel, size: 50.0, color: Colors.purple),

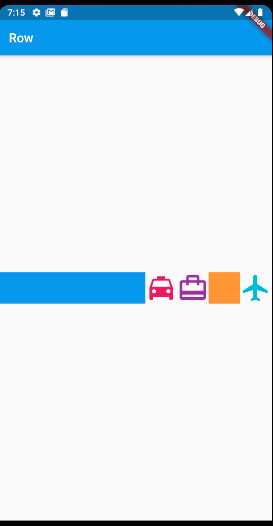
Container(color: Colors.orange, height: 50.0, width: 50.0),

Icon(Icons.local\_airport, size: 50.0, color: Colors.cyan),

],

),

),



#### مثال ٢

body: SizedBox.expand(

child: Column(

mainAxisAlignment: MainAxisAlignment.start,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Container(color: Colors.blue, height: 50.0, width: 50.0),

Icon(Icons.local\_taxi, size: 50.0, color: Colors.pink),

Icon(Icons.card\_travel, size: 50.0, color: Colors.purple),

Expanded(

child: Container(color: Colors.red, height: 50.0, width: 50.0),

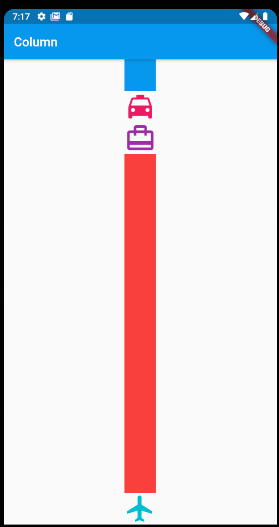
),

Icon(Icons.local\_airport, size: 50.0, color: Colors.cyan),

],

),

),



### خلط بين الأعمدة والصفوف

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

TextEditingController \_user = TextEditingController();

TextEditingController \_pass = TextEditingController();

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Column and Row'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text('Please Login'),

Row(

children: <Widget>[

Text('Username: '),

Expanded(

child: TextField(

controller: \_user,

))

],

),

Row(

children: <Widget>[

Text('Password: '),

Expanded(

child: TextField(

controller: \_pass,

obscureText: true,

))

],

),

Padding(

padding: EdgeInsets.all(32.0),

child: RaisedButton(

onPressed: () => print('Login ${\_user.text}'),

child: Text('Click me'),

),

),

],

),

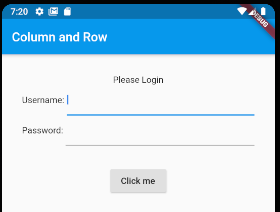
),

),

);

}

}



## بطاقة Card

flutter create --androidx -t app --org com.ah.book -a java -i swift card

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Card'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Card(

//elevation: 5,

child: Container(

padding: EdgeInsets.all(32.0),

child: Column(

children: <Widget>[

Text('Card 1'),

],

),

),

),

Card(

//elevation: 10,

child: Container(

padding: EdgeInsets.all(32.0),

child: Column(

children: <Widget>[

Text('Card 2'),

],

),

),

),

Card(

//elevation: 20,

child: Container(

padding: EdgeInsets.all(32.0),

child: Column(

children: <Widget>[

Text('Card 3'),

],

),

),

)

],

),

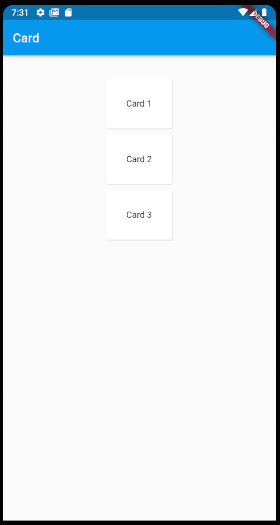
)

),

);

}

}



### إضافة ارتفاع

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Card(

elevation: 5,

child: Container(

padding: EdgeInsets.all(32.0),

child: Column(

children: <Widget>[

Text('Card 1'),

],

),

),

),

Card(

elevation: 10,

child: Container(

padding: EdgeInsets.all(32.0),

child: Column(

children: <Widget>[

Text('Card 2'),

],

),

),

),

Card(

elevation: 20,

child: Container(

padding: EdgeInsets.all(32.0),

child: Column(

children: <Widget>[

Text('Card 3'),

],

),

),

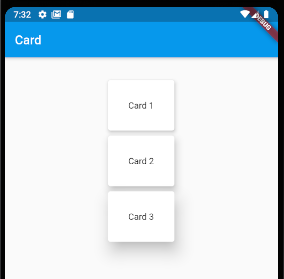
)

],

),

)

),



### إضافة خصائص أخرى

body: Container(

height: 200.0,

width: double.maxFinite,

padding: EdgeInsets.all(32.0),

child: Card(

shape: new RoundedRectangleBorder(

side: new BorderSide(

color: Theme.of(context).accentColor, width: 3.0),

borderRadius: BorderRadius.circular(20.0)),

elevation: 10.0,

child: Container(

padding: EdgeInsets.all(32.0),

child: Column(

children: <Widget>[

Text('Card 1'),

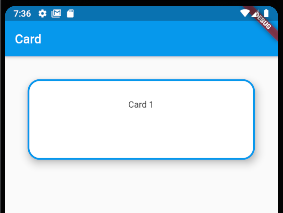
],

),

),

),

),



### ترتيب العناصر داخل البطاقة

body: Container(

width: 200,

child: Card(

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(15.0),

),

color: Colors.pink,

elevation: 10,

child: Column(

mainAxisSize: MainAxisSize.min,

children: <Widget>[

const ListTile(

leading: Icon(Icons.account\_box, size: 70),

title:

Text('First Name', style: TextStyle(color: Colors.white)),

subtitle: Text('Student', style: TextStyle(color: Colors.white)),

),

ButtonBar(

children: <Widget>[

FlatButton(

child: const Text('Edit',

style: TextStyle(color: Colors.white)),

onPressed: () {},

),

FlatButton(

child: const Text('Delete',

style: TextStyle(color: Colors.white)),

onPressed: () {},

),

],

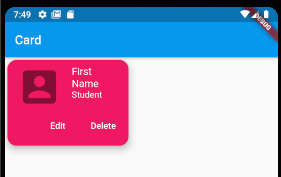
),

],

),

),

),



### مكتبة awesome\_card

<https://pub.dev/packages/awesome_card>

CreditCard(

cardNumber: "5450 7879 4864 7854",

cardExpiry: "10/25",

cardHolderName: "Card Holder",

cvv: "456",

bankName: "Axis Bank",

cardType:

CardType.masterCard, // Optional if you want to override Card Type

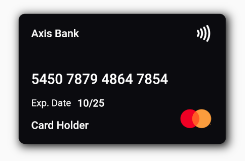
showBackSide: false,

frontBackground: CardBackgrounds.black,

backBackground: CardBackgrounds.white,

showShadow: true,

),



## ListView

### الطريقة الأولى

flutter create --androidx -t app --org com.ah.book -a java -i swift listview

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('ListView'),

),

body: Center(

child: ListView(

padding: const EdgeInsets.all(8),

children: <Widget>[

Container(

height: 50,

color: Colors.amber[600],

child: const Center(child: Text('Entry A')),

),

Container(

height: 50,

color: Colors.amber[500],

child: const Center(child: Text('Entry B')),

),

Container(

height: 50,

color: Colors.amber[100],

child: const Center(child: Text('Entry C')),

),

],

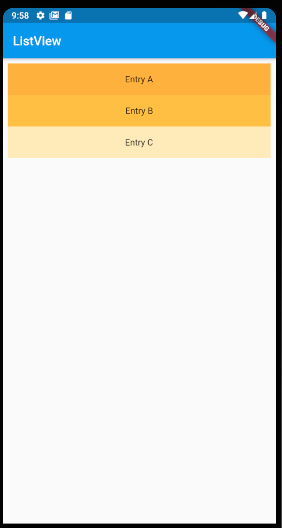
),

),

);

}

}



#### تغيير الاتجاه scrollDirection: Axis.horizontal,

body: Center(

child: ListView(

scrollDirection: Axis.horizontal,

padding: const EdgeInsets.all(8),

children: <Widget>[

Container(

height: 50,

color: Colors.amber[600],

child: const Center(child: Text('Entry A')),

),

Container(

height: 50,

color: Colors.amber[500],

child: const Center(child: Text('Entry B')),

),

Container(

height: 50,

color: Colors.amber[100],

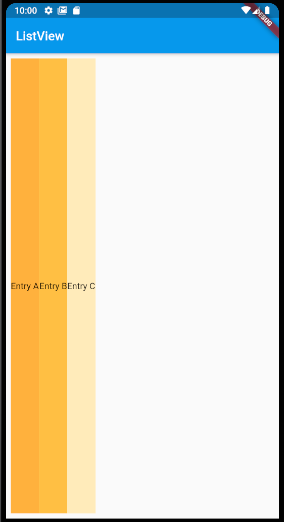
child: const Center(child: Text('Entry C')),

),

],

),

),



### الطريقة الثانية: استعمال ListView.builder

flutter create --androidx -t app --org com.ah.book -a java -i swift listviewbuilder

import 'package:flutter/material.dart';

import 'package:http/http.dart' as http;

import 'dart:async';

import 'dart:convert';

import 'dart:io';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

Map \_countries = Map();

void \_getData() async {

var url = 'http://country.io/names.json';

var response = await http.get(url);

if (response.statusCode == 200) {

setState(() => \_countries = json.decode(response.body));

print('Loaded ${\_countries.length} countries');

} else {

print("Status code: ${response.statusCode}");

}

}

@override

void initState() {

\_getData();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('ListView.builder'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Text(

'Countries',

style: TextStyle(fontWeight: FontWeight.bold),

),

SizedBox(height: 10.0),

Expanded(

child: ListView.builder(

shrinkWrap: true,

itemCount: \_countries.length,

itemBuilder: (BuildContext context, int index) {

String key = \_countries.keys.elementAt(index);

return Container(

margin: const EdgeInsets.all(5),

padding: const EdgeInsets.all(8.0),

color: Colors.blue[200],

child: Row(

children: <Widget>[

Text(

'${key} : ',

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: 15.0,

),

),

Text(

\_countries[key],

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: 15.0,

),

)

],

),

);

},

),

),

],

),

),

),

);

}

}



## GridView

### مثال ١

flutter create --androidx -t app --org com.ah.book -a java -i swift gridview

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('GridView'),

),

body: Container(

height: 200,

child: GridView.count(

scrollDirection: Axis.horizontal,

crossAxisCount: 2,

children: List.generate(5, (index) {

return Container(

child: Card(

color: Colors.amber,

),

);

}),

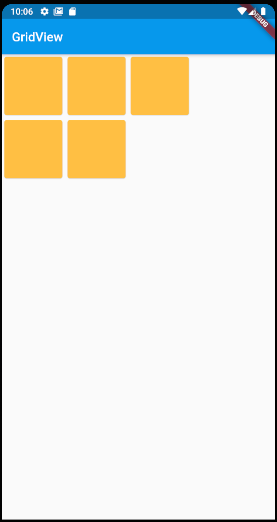
),

),

);

}

}



### مثال ٢

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

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

final String title;

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('GridView'),

),

body: Container(

//height: 200,

child: GridView.count(

//scrollDirection: Axis.horizontal,

crossAxisCount: 2,

children: List.generate(5, (index) {

return Container(

child: Card(

color: Colors.amber,

),

);

}),

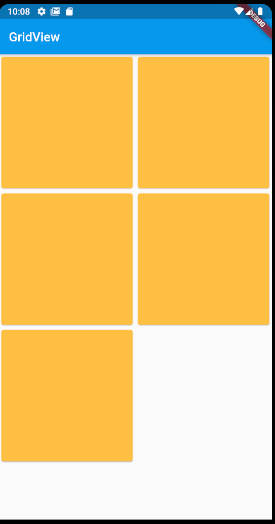
),

),

);

}

}



## Expansion Panel

flutter create --androidx -t app --org com.ah.book -a java -i swift expansionpanel

class TheItem {

bool isExpanded;

final String header;

final Widget body;

TheItem(this.isExpanded, this.header, this.body);

}

class \_MyHomePageState extends State<MyHomePage> {

List<TheItem> \_items = List<TheItem>();

@override

void initState() {

for(int i = 0 ; i< 10; i++) {

\_items.add(TheItem(

false,

'Item ${i}',

Container(

padding: EdgeInsets.all(10.0),

child: Text('Welcome'),

)

));

}

}

ExpansionPanel \_createitem(TheItem item) {

return ExpansionPanel(

headerBuilder: (BuildContext context, bool isExpanded) {

return Container(

padding: EdgeInsets.all(5.0),

child: Text('Header ${item.header}'),

);

},

body: item.body,

isExpanded: item.isExpanded

);

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Expansion Panel'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: ListView(

children: <Widget>[

ExpansionPanelList(

expansionCallback: (int index, bool isExpanded) {

setState(() {

\_items[index].isExpanded = !\_items[index].isExpanded;

});

},

children: \_items.map(\_createitem).toList(),

),

],

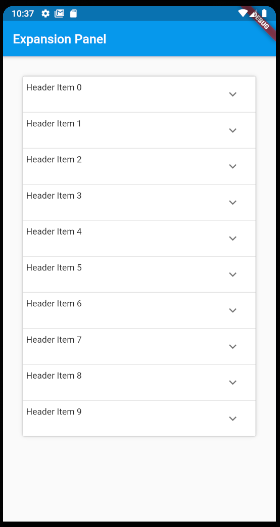
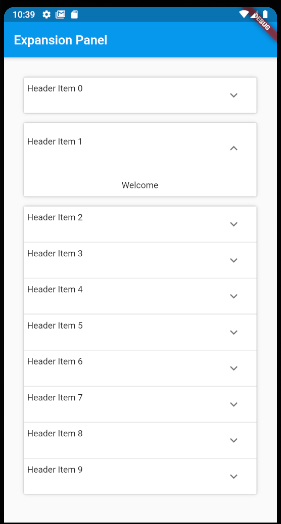
),

),

);

}

}

## شريط التبويب Tabbar

flutter create --androidx -t app --org com.ah.book -a java -i swift tabbar

class Choice {

final String title;

final IconData icon;

const Choice({this.title, this.icon});

}

class \_MyHomePageState extends State<MyHomePage> with SingleTickerProviderStateMixin {

TabController \_controller;

List<Choice> \_items = const <Choice>[

const Choice(title: 'CAR', icon: Icons.directions\_car),

const Choice(title: 'BICYCLE', icon: Icons.directions\_bike),

const Choice(title: 'BOAT', icon: Icons.directions\_boat),

const Choice(title: 'BUS', icon: Icons.directions\_bus),

const Choice(title: 'TRAIN', icon: Icons.directions\_railway),

const Choice(title: 'WALK', icon: Icons.directions\_walk),

];

@override

void initState() {

\_controller = TabController(length: \_items.length, vsync: this);

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Tabbar'),

bottom: PreferredSize(

preferredSize: const Size.fromHeight(48.0),

child: Theme(

data: Theme.of(context).copyWith(accentColor: Colors.white),

child: Container(

height: 48.0,

alignment: Alignment.center,

child: TabPageSelector(controller: \_controller,),

)

),

),

),

body: TabBarView(

controller: \_controller,

children: \_items.map((Choice item){

return Container(

padding: EdgeInsets.all(25.0),

child: Center(

child: Column(

children: <Widget>[

Text(item.title),

Icon(item.icon, size: 120.0,)

],

),

),

);

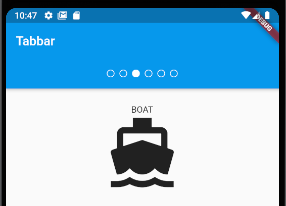
}).toList(),

),

);

}

}



## Stepper

flutter create --androidx -t app --org com.ah.book -a java -i swift stepper

class \_MyHomePageState extends State<MyHomePage> {

List<Step> \_steps;

int \_current;

@override

void initState() {

\_current = 0;

\_steps = <Step>[

Step(title: Text('Step 1'), content: Text('Do Something'), isActive: true),

Step(title: Text('Step 2'), content: Text('Do Something'), isActive: false),

Step(title: Text('Step 3'), content: Text('Do Something'), isActive: false),

];

}

void \_stepContinue() {

setState(() {

\_current++;

if(\_current >= \_steps.length) \_current = \_steps.length - 1;

});

}

void \_stepCancel() {

setState(() {

\_current--;

if(\_current < 0) \_current = 0;

});

}

void \_stepTap(int index) {

setState(() {

\_current = index;

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Stepper'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Column(

children: [

Stepper(

steps: \_steps,

type: StepperType.vertical,

currentStep: \_current,

onStepCancel: \_stepCancel,

onStepContinue: \_stepContinue,

onStepTapped: \_stepTap,

),

],

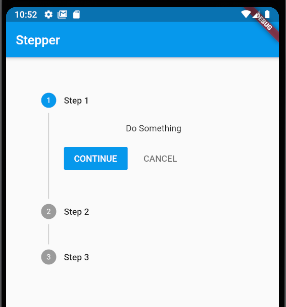
),

),

);

}

}



# الرسوم البيانية Charts

# أدوات أخرى

## Chip Widget

Chips أو الرقائق هي عناصر صغيرة تمثل سمة Attribute أو نصًا Text أو كيانًا Entity أو إجراءً Action.

flutter create --androidx -t app --org com.ah.book -a java -i swift chip

class \_MyHomePageState extends State<MyHomePage> {

int counter = 0;

List<Widget> \_list = List<Widget>();

@override

void initState() {

for (int i = 0; i < 5; i++) {

Widget child = \_newItem(i);

\_list.add(child);

}

}

void \_onClicked() {

Widget child = \_newItem(counter);

setState(() => \_list.add(child));

}

Widget \_newItem(int i) {

Key key = Key('item\_${i}');

Container child = Container(

key: key,

padding: EdgeInsets.all(10.0),

child: Chip(

label: Text('${i} Name here', style: TextStyle(fontSize: 20.0)),

materialTapTargetSize: MaterialTapTargetSize.shrinkWrap,

deleteIconColor: Colors.red,

deleteButtonTooltipMessage: 'Delete',

onDeleted: () => \_removeItem(key),

avatar: CircleAvatar(

backgroundColor: Colors.grey.shade800,

child: Text(i.toString()),

),

shape:

RoundedRectangleBorder(borderRadius: BorderRadius.circular(12.0)),

elevation: 6.0,

//backgroundColor: Colors.white,

),

);

counter++;

return child;

}

void \_removeItem(Key key) {

for (int i = 0; i < \_list.length; i++) {

Widget child = \_list.elementAt(i);

if (child.key == key) {

setState(() => \_list.removeAt(i));

print('Removing ${key.toString()}');

}

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Chips'),

),

floatingActionButton: FloatingActionButton(

onPressed: \_onClicked,

child: Icon(Icons.add),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: \_list,

),

)),

);

}

}

## Sliders and Progress Indicators

flutter create --androidx -t app --org com.ah.book -a java -i swift slider

class \_MyHomePageState extends State<MyHomePage> {

double \_value = 0.0;

void \_onChanged(double value) => setState(() => \_value = value);

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Slider and Progress indicators'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Column(

children: <Widget>[

Slider(value: \_value, onChanged: \_onChanged),

Container(

padding: EdgeInsets.all(32.0),

child: LinearProgressIndicator(

value: \_value,

valueColor: AlwaysStoppedAnimation<Color>(Colors.green),

),

),

Container(

padding: EdgeInsets.all(32.0),

child: CircularProgressIndicator(

value: \_value,

),

)

],

),

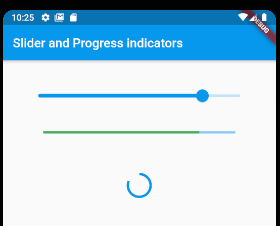
)

),

);

}

}



## زر القائمة المنبثقة PopupMenuButton

flutter create --androidx -t app --org com.ah.book -a java -i swift popupmenubutton

enum Animals{Cat, Dog, Bird, Lizard, Fish}

class \_MyHomePageState extends State<MyHomePage> {

Animals \_selected = Animals.Cat;

String \_value = 'Make a Selection';

List<PopupMenuEntry<Animals>> \_items = List<PopupMenuEntry<Animals>>();

@override

void initState() {

for(Animals animal in Animals.values) {

\_items.add(PopupMenuItem(

child: Text(\_getDisplay(animal),),

value: animal,

));

}

}

void \_onSelected(Animals animal) {

setState((){

\_selected = animal;

\_value = 'You Selected ${\_getDisplay(animal)}';

});

}

String \_getDisplay(Animals animal) {

int index = animal.toString().indexOf('.');

index++;

return animal.toString().substring(index);

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('PopupMenuButton'),

),

body: Container(

padding: EdgeInsets.all(32.0),

child: Center(

child: Row(

children: <Widget>[

Container(

padding: EdgeInsets.all(5.0),

child: Text(\_value),

),

PopupMenuButton<Animals>(

child: Icon(Icons.input),

initialValue: Animals.Cat,

onSelected: \_onSelected,

itemBuilder: (BuildContext context) {

return \_items;

}

),

],

),

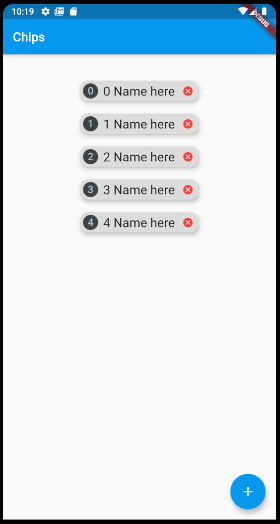
),

),

);

}

}



الجزء الثاني

A theme is a widget that you can insert anywhere in your widget tree. It overrides the current theme with custom values Try this:

Theme(

data: Theme.of(context).copyWith(accentColor: Colors.yellow),

child: new CircularProgressIndicator(),

);