Text

Description automatically generated

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Assignment No** | 1 |

Assignment Number - 01

**Title :** Design Flutter application to display your name, address and mobile number

**Theory :**

**Text Widget :** A Text is a widget in Flutter that allows us to display a string of text with a single line in application. Depending on the layout constraints, we can break the string across multiple lines or might all be displayed on the same line. If we do not specify any styling to the text widget, it will use the closest DefaultTextStyle class style. This class does not have any explicit style.

**Text Widget Constructor:**

The text widget constructor used to make the custom look and feel of our text in [Flutter](https://www.javatpoint.com/flutter):

|  |
| --- |
| **const** Text(String data,{      Key key,      TextStyle style,      StrutStyle strutStyle,      TextAlign textAlign,      TextDirection textDirection,      TextOverflow overflow,      bool softWrap,  **double** textScaleFactor,  **int** maxLines,      String semanticsLabel,      TextWidthBasis textWidthBasis,      TextHeightBehavior textHeightBehavior      }  ) |

The following are the essential properties of the Text widget :

**TextAlign:** It is used to specify how our text is aligned horizontally. It also controls the text location.

**TextDirection:** It is used to determine how textAlign values control the layout of our text. Usually, we write text from left to right, but we can change it using this parameter.

**Overflow:** It is used to determine when the text will not fit in the available space. It means we have specified more text than the available space.

**TextScaleFactor:** It is used to determine the scaling to the text displayed by the Text widget. Suppose we have specified the text scale factor as 1.5, then our text will be 50 percent larger than the specified font size.

**SoftWrap:** It is used to determine whether or not to show all text widget content when there is not enough space available. If it is true, it will show all content. Otherwise, it will not show all content.

**MaxLines:** It is used to determine the maximum number of lines displayed in the text widget.

**TextWidthBasis:** It is used to control how the text width is defined.

**TextHeightBehavior:** It is used to control how the paragraph appears between the first line and descent of the last line.

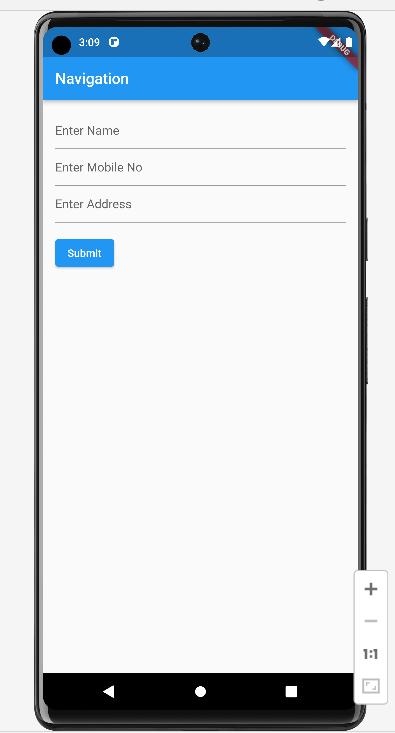
**Style:** It is the most common property of this widget that allows developers to styling their text. It can do styling by specifying the foreground and background color, font size, font weight, letter and word spacing, locale, shadows, etc. See the table to understand it more easily:

|  |  |
| --- | --- |
| **Attributes** | **Descriptions** |
| **foreground** | It determines the paint as a foreground for the text. |
| **background** | It determines the paint as a background for the text. |
| **fontWeight** | It determines the thickness of the text. |
| **fontSize** | It determines the size of the text. |
| **fontFamily** | It is used to specify the typeface for the font. For this, we need to download a typeface file in our project, and then keep this file into the assets/font folder. Finally, config the pubspec.yaml file to use it in the project. |
| **fontStyle** | It is used to style the font either in bold or italic form. |
| **Color** | It is used to determine the color of the text. |
| **letterSpacing** | It is used to determine the distance between the characters of the text. |
| **wordSpacing** | It is used to specify the distance between two words of the text. |
| **shadows** | It is used to paint underneath the text. |
| **decoration** | We use this to decorate text using the three parameters: decoration, decorationColor, decorationStyle. The decoration determines the location, decorationColor specify the color, decorationStyle determine the shape. |

**Source Code:-**

import 'package:flutter/material.dart';  
  
void main() => runApp(MyApp());  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'Multiple Text Fields',  
 home: FirstRoute(),  
 );  
 }  
}  
  
class FirstRoute extends StatefulWidget {*//stateful is base class* @override  
 \_FirstRouteState createState() => \_FirstRouteState();  
}  
  
class \_FirstRouteState extends State<FirstRoute> {  
 final \_formKey = GlobalKey<FormState>();*//what are global keys* final \_textController1 = TextEditingController();*//value remains same(final)* final \_textController2 = TextEditingController();  
 final \_textController3 = TextEditingController();  
  
 @override  
 void dispose() {  
 \_textController1.dispose();  
 \_textController2.dispose();  
 \_textController3.dispose();  
 super.dispose();  
 }  
  
 void \_submitForm() {  
 if (\_formKey.currentState!.validate()) {  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => SecondRoute(  
 name: \_textController1.text,  
 mobileNo: \_textController2.text,  
 address: \_textController3.text,  
 ),  
 ),  
 );  
 }  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Navigation'),  
 ),  
 body: Padding(  
 padding: EdgeInsets.all(16.0),  
 child: Form(  
 key: \_formKey,  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: <Widget>[  
 TextFormField(  
 controller: \_textController1,  
 decoration: InputDecoration(  
 hintText: 'Enter Name',  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Please enter name ';  
 }  
 return null;  
 },  
 ),  
 TextFormField(  
 controller: \_textController2,  
 decoration: InputDecoration(  
 hintText: 'Enter Mobile No',  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Please enter Mobile No';  
 }  
 return null;  
 },  
 ),  
 TextFormField(  
 controller: \_textController3,  
 decoration: InputDecoration(  
 hintText: 'Enter Address',  
 ),  
 validator: (value) {  
 if (value == null || value.isEmpty) {  
 return 'Please enter address';  
 }  
 return null;  
 },  
 ),  
 SizedBox(height: 16.0),  
 ElevatedButton(  
 onPressed: \_submitForm,  
 child: Text('Submit'),  
 ),  
 ],  
 ),  
 ),  
 ),  
 );  
 }  
}  
  
class SecondRoute extends StatelessWidget {  
 final String name;  
 final String mobileNo;  
 final String address;  
  
 SecondRoute({  
 required this.name,  
 required this.mobileNo,  
 required this.address,  
 });  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Next Route'),  
 ),  
 body: Center(  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Text(  
 'Name: $name',  
 style: TextStyle(  
 color: Colors.*black*,  
 fontSize: 35,  
 ),  
 ),  
 Text(  
 'Mobile No: $mobileNo',  
 style: TextStyle(  
 color: Colors.*deepOrange*,  
 fontSize: 35,  
 ),  
 ),  
 Text(  
 'Address: $address',  
 style: TextStyle(  
 color: Colors.*green*,  
 fontSize: 35,  
 ),  
 ),  
 ],  
 ),  
 ),  
 );  
 }  
}

**Output:**



**Conclusion : in this assignment 1 learn about different attributes**