Text

Description automatically generated

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Assignment No** | 2 |

Assignment Number - 02

**Title :** Design a simple calculator by using flutter

**Theory :**

**Converting a String to an Integer**

In Flutter, you can convert a string to an integer using the int.parse(string) method. This method takes a string as an argument and returns the corresponding integer.

A string can be cast to an integer using the int.parse() method in Dart. The method takes a string as an argument and converts it into an integer.

****

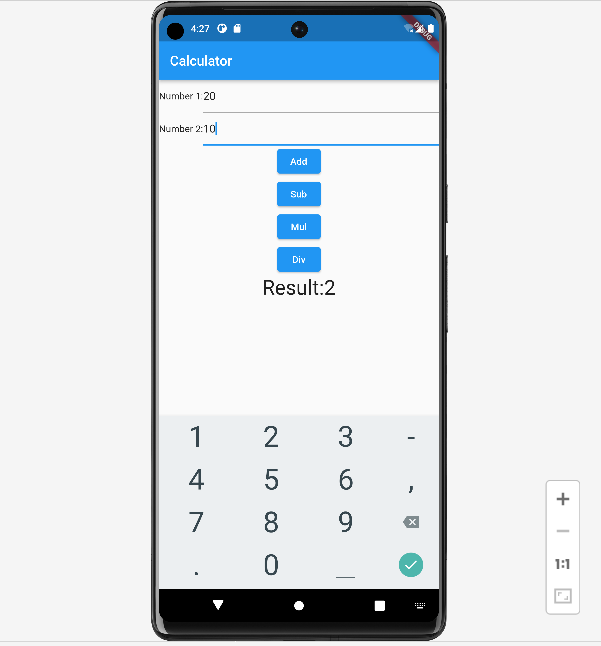
**tryParse() method**

When using tryParse() method, we just need to provide the input String. In case the input is invalid, the program will return null. This is the difference between tryParse() and parse() (which throws exception if we don’t handle onError method).

**Sourcecode**

import 'package:flutter/material.dart';  
  
void main() => runApp(MyApp());  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 final appTitle = 'Calculator';  
  
 return MaterialApp(  
 title: appTitle,  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text(appTitle),  
 ),  
 body: Calculator(),  
 ),  
 );  
 }  
}  
class Calculator extends StatefulWidget {  
 @override  
 \_CalculatorState createState() => \_CalculatorState();  
}  
  
  
  
  
class \_CalculatorState extends State<Calculator> {  
 TextEditingController num1controller = new TextEditingController();  
 TextEditingController num2controller = new TextEditingController();  
 String result = "0";  
 @override  
 Widget build(BuildContext context) {  
 return Container(  
 child: Column(  
 children: <Widget>[  
 Row(  
 children: <Widget>[  
 Text("Number 1:"),  
 new Flexible(  
 child: new TextField(  
 keyboardType: TextInputType.*number*,  
 controller: num1controller,  
 ),  
 ),  
 ],  
 ),  
 Row(  
 children: <Widget>[  
 Text("Number 2:"),  
 new Flexible(  
 child: new TextField(  
 keyboardType: TextInputType.*number*,  
 controller: num2controller,  
 ),  
 ),  
 ],  
 ),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 ElevatedButton(  
 child: Text("Add"),  
 onPressed : () {  
 setState(() {  
 int sum = int.*parse*(num1controller.text) + int.*parse*(num2controller.text);  
 result = sum.toString();  
 });  
 },  
 )  
 ],  
 ),  
  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 ElevatedButton(  
 child: Text("Sub"),  
 onPressed: (){  
 setState(() {  
 int Sub = int.*parse*(num1controller.text) - int.*parse*(num2controller.text);  
 result = Sub.toString();  
 });  
 },  
 )  
 ],  
 ),  
  
  
  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 ElevatedButton(  
 child: Text("Mul"),  
 onPressed: (){  
 setState(() {  
 int Mul = int.*parse*(num1controller.text) \* int.*parse*(num2controller.text);  
 result = Mul.toString();  
 });  
 },  
 )  
 ],  
 ),  
  
  
  
  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 ElevatedButton(  
 child: Text("Div"),  
 onPressed: (){  
 setState(() {  
 int Div = int.*parse*(num1controller.text) ~/ int.*parse*(num2controller.text);  
 result = Div.toString();  
 });  
 },  
 )  
 ],  
 ),  
  
  
  
  
  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Text("Result:",  
 style: TextStyle(  
 fontSize: 30,  
 ),),  
 Text(result,  
 style: TextStyle(  
 fontSize: 30,  
 ),),  
 ],  
 ),  
 ],  
 ),  
 );  
 }  
}

**Output:**



**Conclusion : In this assignment I have learn about int.parse which parse a string into an int in flutter . Parse source as a, possibly signed, integer literal and return its value.**

**The source must be a non-empty sequence of base-radix digits, optionally prefixed with a minus or plus sign ('-' or '+').**