

My Project link:

<https://zapp.run/edit/flutter-z2ay06j12az0?entry=lib/main.dart&file=lib/main.dart>

//Balbin, Edifer Jade D.

```
import 'package:flutter/material.dart';
void main() => runApp(MyApp());
class MyApp extends StatefulWidget {

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

class MyAppState extends State<MyApp> {
  double _numberFrom=0;
  String? _startMeasure;
  String? _convertedMeasure;
  String? _resultMessage = '';

  final Map<String, int> _measuresMap = { 'meters' : 0,
    'kilometers' : 1,
    'grams' : 2,
    'kilograms' : 3,
    'feet' : 4,
    'miles' : 5,
    'pounds (lbs)' : 6,
    'ounces' : 7,
  };

  final dynamic _formulas = {
    '0':[1,0.001,0,0,3.28084,0.000621371,0,0],
    '1':[1000,1,0,0,3280.84,0.621371,0,0],
    '2':[0,0,1,0.0001,0,0,0.00220462,0.035274],
    '3':[0,0,1000,1,0,0,2.20462,35.274],
    '4':[0.3048,0.0003048,0,0,1,0.000189394,0,0],
    '5':[1609.34, 1.60934,0,0,5280,1,0,0],
    '6':[0,0,453.592,0.453592,0,0,1,16],
    '7':[0,0,28.3495,0.0283495,3.28084,0,0.0625, 1],
  };
};
```

```

__@override
__void initState() {
__  __numberFrom = 0; super.initState();
__}

__void convert(double value, String from, String to) {

__  int nFrom = __measuresMap[from!];
__  int nTo = __measuresMap[to!];
__  var multiplier = __formulas[nFrom.toString()][nTo];
__  var result = value * multiplier;

__  if (result == 0) {
__resultMessage = 'This conversion cannot be performed';
__}
__else {
__resultMessage = '${__numberFrom.toString()} $__startMeasure
are${result.toString()} $__convertedMeasure';
__}
__setState(() {
__  __resultMessage = __resultMessage;
__});

__}

__final List<String> __measures = [ 'meters',
__  'kilometers', 'grams', 'kilograms', 'feet',
__  'miles', 'pounds (lbs)', 'ounces',
__];

__final TextStyle inputStyle = TextStyle(
__  fontSize: 20,
__  color: Colors.blue[900],
__);

__final TextStyle labelStyle = TextStyle(
__  fontSize: 24,
__  color: Colors.grey[700],
__);

```

```

Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Measures Converter',
    home: Scaffold(
      appBar: AppBar(
        title: Center(
          child: Text ('Measures Converter')),
        ),
      body: Container(
        padding: EdgeInsets.symmetric(horizontal: 20),
        child: SingleChildScrollView(
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: [
              SizedBox(height: 20),
              Center (
                child: Text(
                  'Value',
                  style: labelStyle,
                ),
              ),
              SizedBox(height: 10),
              TextField(
                style: inputStyle,
                decoration: InputDecoration(
                  hintText: "Please insert the measure to
be converted",
                ),
                onChanged: (text) {
                  var rv = double.tryParse(text);
                  if (rv != null) {
                    setState(() {
                      _numberFrom = rv;
                    });
                  }
                },
              ),
              SizedBox(height: 20),

```

```

Center (
  child: Text(
    'From',
    style: labelStyle,
  ),
),
DropdownButton<String>(
  isExpanded: true,
  items: _measures.map((String value) {
    return DropdownMenuItem<String>(
      value: value,
      child: Text(value, style: inputStyle),
    );
  }).toList(),
  onChanged: (value) {
    setState(() {
      _startMeasure = value;
    });
  },
  value: _startMeasure,
),
 SizedBox(height: 20),
  Center (
    child: Text(
      'To',
      style: labelStyle,
    ),
  ),
  DropdownButton<String>(
    isExpanded: true,
    style: inputStyle,
    items: _measures.map((String value) {
      return DropdownMenuItem<String>(
        value: value,
        child: Text(value, style: inputStyle),
      );
    }).toList(),
  ),

```

```

        onChanged: (value) {
          setState(() {
            _convertedMeasure = value;
          });
        },
        value: _convertedMeasure,
      ),

      SizedBox(height: 40),
      ElevatedButton(
        child: Text('Convert', style: inputStyle),
        onPressed: () {
          if (_startMeasure == null || _convertedMeasure
== null || _numberFrom == 0) {
            setState(() {
              _resultMessage = 'Please provide valid
inputs for conversion.';
            });
          } else {
            convert(_numberFrom, _startMeasure!,
_convertedMeasure!);
          }
        },
      ),

      SizedBox(height: 20),
      Text(
        (_resultMessage == null) ? '' : _resultMessage!,
        style: labelStyle,
        textAlign: TextAlign.center,
      ),
    ],
  ),
),
),
),
),
);
}
}

```