



MOBILE APPLICATION FLUTTER TEXT



No Directionality widget found

- **problem**

flutter doesn't know whether the text is **LTR** or **RTL**.


- **Solve**

1) you need to tell the **TextDirection**

```
Text("Hello", textDirection: TextDirection.ltr)
```

2) you can just wrap the Text with a **Directionality Widget**

```
Directionality(  
  textDirection: TextDirection.ltr,  
  child: new Text('Hello')
```

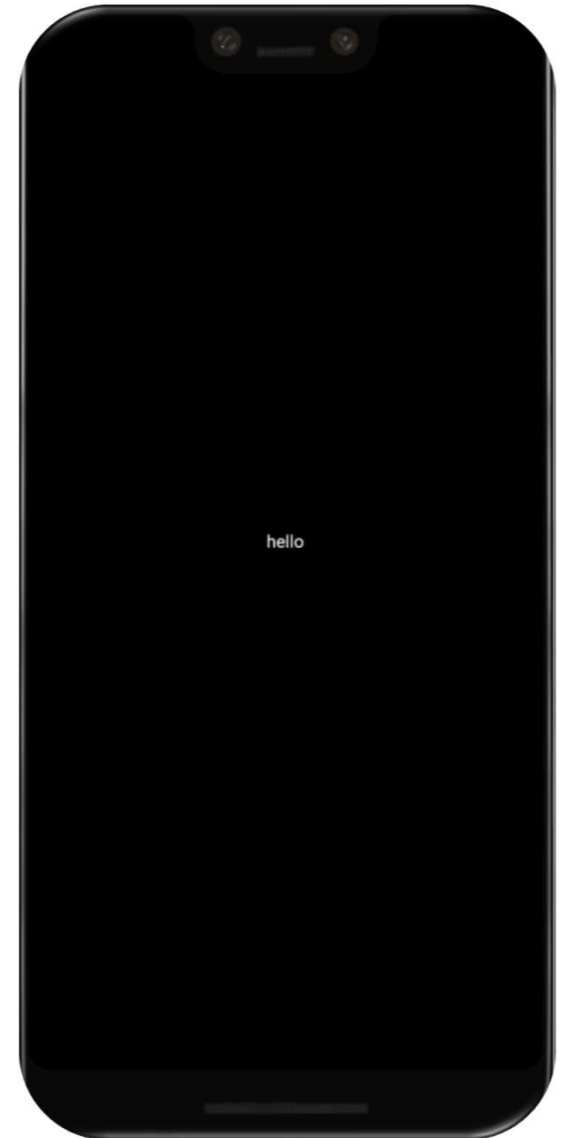


```
No Directionality widget found.  
RichText widgets require a  
Directionality widget ancestor.  
The specific widget that could  
not find a Directionality  
ancestor was:  
  RichText  
The ownership chain for the  
affected widget is: "RichText ←  
Text ← Center ← [root]"  
Typically, the Directionality  
widget is introduced by the  
MaterialApp or WidgetsApp widget  
at the top of your application  
widget tree. It determines the  
ambient reading direction and is  
used, for example, to determine  
how to lay out text, how to  
interpret "start" and "end"  
values, and to resolve  
EdgeInsetsDirectional,  
AlignmentDirectional, and other  
*Directional objects.  
See also: https://flutter.dev/  
docs/testing/errors
```

Flutter Text

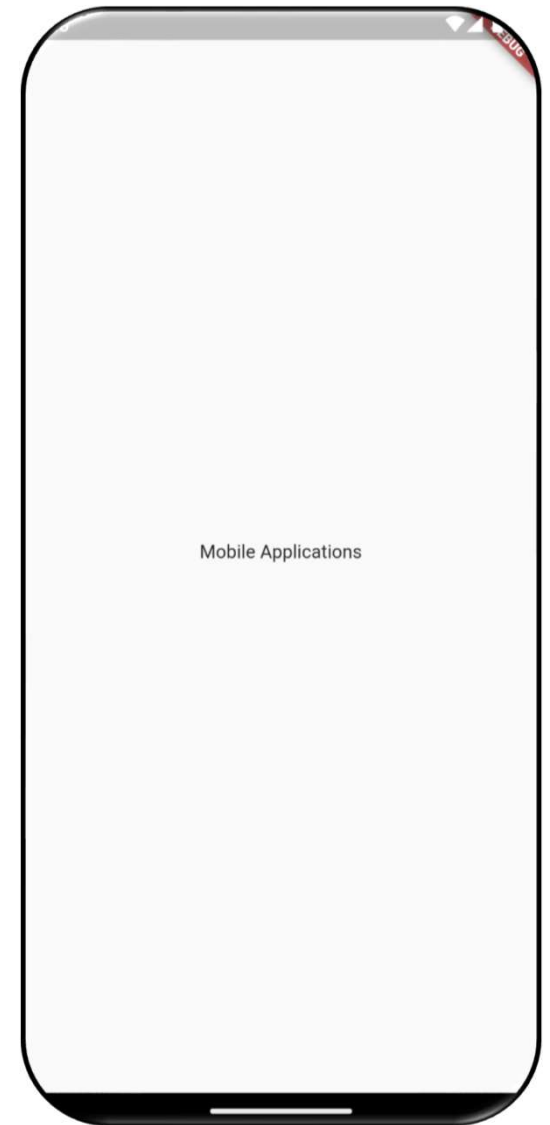
- A Text is a widget in Flutter that allows us to **display a string of text with a single line in our application**. Depending on the layout constraints,

```
import 'package:flutter/material.dart';  
void main(){  
  runApp(Center(  
    child: Text('hello',textDirection: TextDirection.ltr,  
  ),  
));  
}
```



Try this

```
import 'package:flutter/material.dart';
void main() {
  runApp(MyTextApp());
}
class MyTextApp extends StatelessWidget {
  const MyTextApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        body: Center(
          child: Text(
            "Mobile Applications",
          ),
        ),
      ),
    );
  }
}
```



1. Style: It is the most common property of this widget that allows developers to styling their text.

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.

```
style: TextStyle(  
fontSize: 34,  
fontWeight: FontWeight.bold,  
// color: Colors.indigo,  
foreground: Paint()  
..style = PaintingStyle.stroke //or use fill  
..strokeWidth = 2  
..color = Colors.red,
```

Use One of them
(color or foreground)

```
letterSpacing: 2,      wordSpacing: 5,  
shadows: [  
Shadow( blurRadius: 15.0,    color: Colors.blue,  
offset: Offset(0, 10),  
),  
],
```

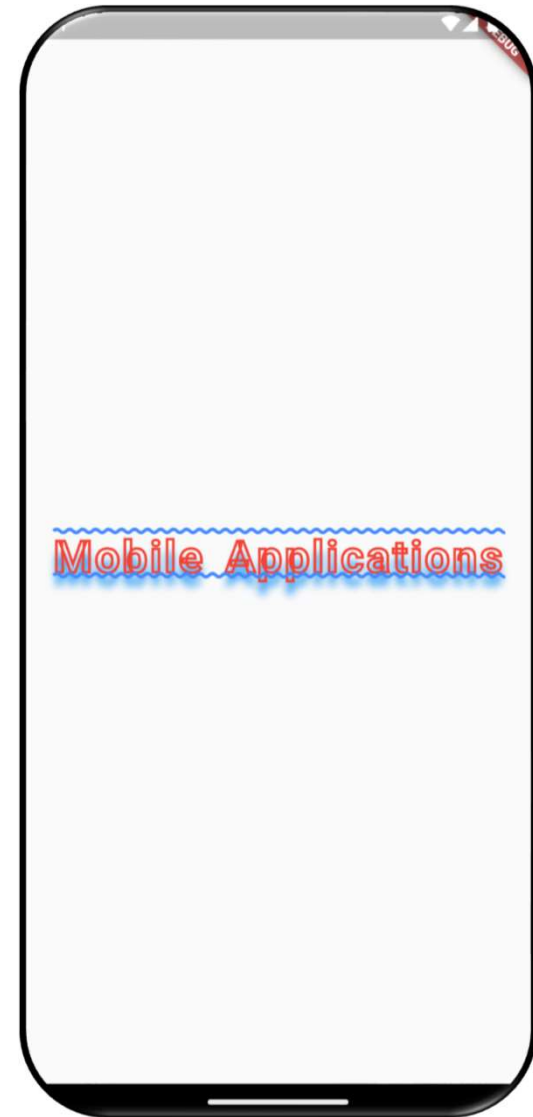
Offset(X , Y) try (10 , -10)

```
decoration: TextDecoration.combine([  
//TextDecoration.lineThrough,  
TextDecoration.overline,  
TextDecoration.underline ]),
```

```
decorationThickness: 1.5,  
decorationColor: Colors.blueAccent,  
decorationStyle: TextDecorationStyle.wavy,
```

Not Working without
TextDecoration

```
height: 2,  
// backgroundColor: Colors.amberAccent,  
),
```



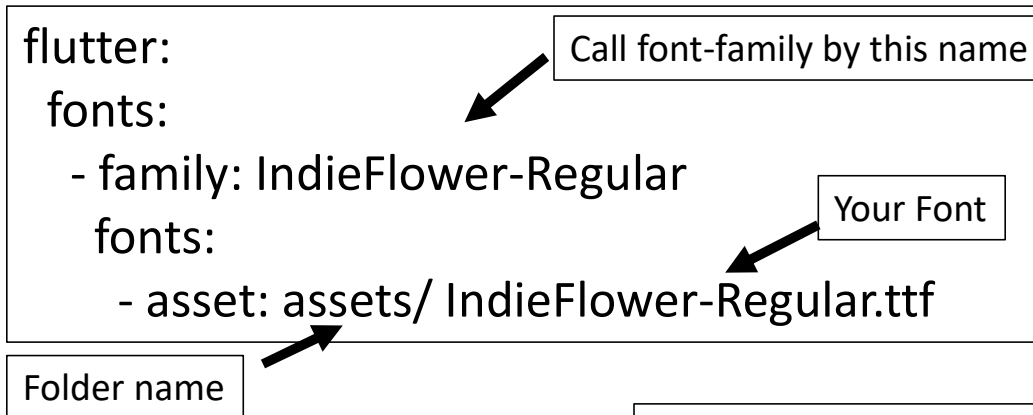
Custom Font Family

1. Add a font to your project

- Right click on your project folder and go to New > Directory. Call it ***assets or (your Folder)***. It should be in your projects root directory.
- Copy and paste ***your font*** into the new ***assets or (your Folder)*** directory. I'm just using a single font in my example, the regular **IndieFlower-Regular** font. I renamed it to **IndieFlower-Regular.ttf**.

2. Register the font

- Open your ***pubspec.yaml*** file.
- Add the fonts info under the ***flutter*** section. Indentation is mandatory.



Example:

```
Text(  
  'Hello World!',  
  style: TextStyle(  
    fontFamily: "IndieFlower-Regular",  
  ),  
),
```

This image below is a default in *pubspec.yaml*
You can selected and press **Ctrl+?** To remove #
then rename font family and path your fonts

```
# example:  
# fonts:  
#   - family: hazhar  
#     fonts:  
#       - asset: Myfonts/font1.ttf  
#       - asset: fonts/Schlyer-Italic.ttf  
#         style: italic
```


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

```
Text(  
  'Hello World!',  
  textAlign: TextAlign.center,  
),
```

3- 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.

```
Text(  
  'Hello World!',  
  textDirection: TextDirection.ltr,  
  //textDirection: TextDirection.rtl  
),
```

Directionality
?

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

5-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.

Wrap Overflow Text With **Clip:**

```
Text("longtext", //put your own long text here.  
    maxLines: 2,  
    overflow:TextOverflow.clip,  
)
```

Wrap Overflow Text with **Fade:**

```
Text("longtext", //put your own long text here.  
    maxLines: 2,  
    overflow:TextOverflow.fade,  
)
```


Wrap Overflow Text With **Ellipsis**

```
Text("longtext", //put your own long text here.  
    maxLines: 2,  
    overflow:TextOverflow.ellipsis,  
)
```

Flutter is an opensource framework by Google for building beautiful, natively compiled, multi-

Flutter is an opensource framework by Google for building beautiful, natively compiled, multi-

Flutter is an opensource framework by Google for building beautiful, natively compiled, multi-...

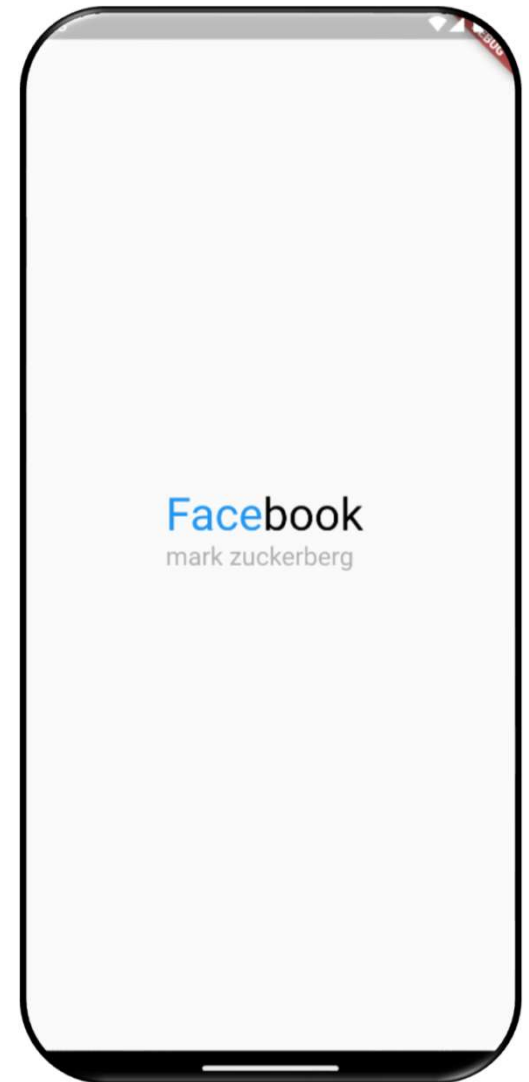


RichText & TextSpan

- RichText is a widget in Flutter used for **displaying a paragraph of text with multiple styles**. Inside the widget, you can have different styles by having multiple **TextSpan** widgets, each can set its own style
- **TextSpan** is an *immutable* span of text. It has style property to give style to the text. It is also having *children* property to add more text to this widget and give style to the children.

Example

```
Scaffold(  
  body: Center(  
    child: RichText(  
  
      text: TextSpan(  
style: TextStyle(color: Colors.black, fontSize: 36),  
      children: [  
  
TextSpan(text: 'Face', style: TextStyle(color: Colors.blue)),  
TextSpan(text: 'book '),  
TextSpan(text: '\n mark zuckerberg', style: TextStyle(  
          fontSize: 20,  
          color: Colors.black.withOpacity(0.3),  
        )),  
      ],  
    ),  
  ),  
),  
)
```



?