Flutter中的文本框主要有两个组件：

- Text
- RichText

一、Text

Text是显示文本的Widget

代码位置：

Flutter_widget_demo/lib/text/TextWidget.dart

1、构造函数

```
const Text(
 this.data,
 {
 Key key,
 this.style,
 this.strutStyle,
 this.textAlign,
 this.textDirection,
 this.locale,
  this.softWrap,
  this.overflow,
```

```
  this.textScaleFactor,
  this.maxLines,
  this.semanticsLabel,
  this.textWidthBasis,
}) : assert(
  data != null,
  'A non-null String must be provided to a Text widget.',
  ),
  textSpan = null,
  super(key: key);
```

从构造函数来看，参数分为两类

- 必选参数，data
- 可选命名参数，{ ... }

1、必选参数data

data是Text的必选参数，表示文本内容，类型为String

2、可选命名参数

参数名字	参数类型	意义	必选 or 可选
`data`	String	要显示的文字	必选
`key`	Key	Widget 的标识	可选
`style`	TextStyle	文本样式	可选
`strutStyle`	StrutStyle	设置每行的最小行高	可选
`textAlign`	TextAlign	文本的对齐方式	可选
`textDirection`	TextDirection	文字方向	可选
`locale`	Locale	用于选择用户语言和格式设置首选项的标识符	可选
`softWrap`	bool	是否支持软换行符 如果是 false 的话，这个文本只有一行，水平方向是无限的	可选
`overflow`	TextOverflow	文本的截断方式	可选
`textScaleFactor`	double	代表文本相对于当前字体大小的缩放因子 默认值为1.0	可选
`maxLines`	int	显示的最大行数	可选
`semanticsLabel`	String	给文本加上一个语义标签 没有实际用处	可选

(1) style

TextStyle类型，用于定义文字颜色、大小、背景等

```
Text(
 "Hello Flutter",
 style: TextStyle(
 color: Colors.red,
 fontSize: 20.0,
 background: new Paint()..color = Colors.yellow,
 ),
 )
```

其中fontSize的单位是pixel

TextStyle的构造函数如下：

```
class TextStyle extends Diagnosticable {

 const TextStyle({
 this.inherit = true,
 this.color,
 this.fontSize,
 this.fontWeight,
 this.fontStyle,
 this.letterSpacing,
  this.wordSpacing,
  this.textBaseline,
  this.height,
  this.locale,
  this.foreground,
  this.background,
  this.shadows,
  this.decoration,
  this.decorationColor,
  this.decorationStyle,
  this.debugLabel,
  String fontFamily,
  List<String> fontFamilyFallback,
  String package,
  }) : fontFamily = package == null ? fontFamily : 'packages/$package/$fo
ntFamily',
  _fontFamilyFallback = fontFamilyFallback,
```

```
  _package = package,
  assert(inherit != null),
  assert(color == null || foreground == null, _kColorForegroundWarning);

  ...
}
```

参数名字	参数类型	意义	必选 or 可选
inherit	bool	是否继承父 Text 的样式，默认为 true 如果为false，且样式没有设置具体的值，则采用默认值：白色、字体大小 10px、sans-serif 字体	可选
color	Color	文字的颜色	可选
fontSize	double	文字的大小	可选
fontWeight	FontWeight	字体粗细	可选
fontStyle	FontStyle	是否在字体中倾斜字形	可选
letterSpacing	double	字母之间的间隔	可选
wordSpacing	double	单词之间的间隔	可选
textBaseline	TextBaseline	用于对齐文本的水平线	可选
height	double	文本的高度 但它并不是一个绝对值，而是一个因子，具体的行高等于fontSize*height。	可选
locale	Locale	用于选择用户语言和格式设置首选项的标识符	可选
foreground	Paint	文本的前景色	可选
background	Paint	文本的背景色	可选
shadows	List<ui.Shadow>	在文本下方绘制阴影	可选
decoration	TextDecoration	文本的线条	可选
decorationColor	Color	TextDecoration 线条的颜色	可选
decorationStyle	TextDecorationStyle	TextDecoration 线条的样式	可选
debugLabel	String	文本样式的描述 无实际用处	可读
fontFamil	String	用于设置使用哪种自定义字体	可读

y			
fontFamilyFallback	String	字体列表，当前面的字体找不到时，会在这个列表里依次查找	可读
package	String	用于设置使用哪种自定义字体	可读

可以看到 TextStyle 只有可选参数，没有必选参数。

- TextDecoration:文本的线条TextDecoration.underline：下划线

1. TextDecoration.overline：上划线
2. TextDecoration.lineThrough：中划线
3. TextDecoration.none：不划线

- decorationStyle：文本线条的种类，即TextDecoration的线条类型

1. TextDecorationStyle.solid：实线
2. TextDecorationStyle.double：两条线
3. TextDecorationStyle.dotted：点虚线
4. TextDecorationStyle.dashed：间隔虚线（比点要长）
5. TextDecorationStyle.wave：波浪线

（2）textAlign

表示文本对齐方式

常见的文本对齐方式，共有六种：

1. TextAlign.left：左对齐
2. TextAlign.right：右对齐
3. TextAlign.center：居中对齐
4. TextAlign.start：从文字开始的那个方向对齐，如果文字方向从左到右，就左对齐，否则是右对齐。
5. TextAlign.end：从文字开始的相反方向对齐，如果文字方向从左到右，就右对齐，否则是左对齐。
6. TextAlign.justify

（3）textDirection

表示文字方向，支持从左到右、从右到左

1. TextDirection.ltr：文字方向从左到右
2. TextDirection.ltr：文字方向从右到左

（4）overflow

表示文本的截断方式

当要显示的内容超了之后，文本的截断方式有三种：

1. TextOverflow.ellipsis：多余文本截断后以省略符“...”表示

2. TextOverflow.clip：剪切多余文本，多余文本不显示

3. TextOverflow.fade：将多余的文本设为透明

（5）Text.rich()

Text.rich（）需要传入TextSpan，可以表示多种样式的Text，类似于Android中的SpanString

二、RichText

一个富文本Text，可以显示多种样式的text

代码位置：

flutter_widget_demo/lib/text/RichTextWidget.dart

1、构造函数

```
RichText(
 text: TextSpan(children: [
 TextSpan(text: "Hello", style: TextStyle(color: Colors.blue)),
 TextSpan(text: "Flutter", style: TextStyle(color: Colors.red))
 ]),
 )
```

RichText构造函数中只有一个必须参数就是TextSpan数组

TextSpan构造函数如下：

```
class TextSpan extends DiagnosticableTree {
 const TextSpan({
 this.style,
 this.text,
 this.children,
 this.recognizer,
 });
 ...
}
```

参数名字	参数类型	意义	必选 or 可选
style	TextStyle	文本样式	可选
text	String	要显示的文字	可选
children	List<TextSpan>	子 TextSpan	可选
recognizer	GestureRecognizer	一个手势识别器，它将接收到达此文本范围的事件。	可选

三、使用自定义字体

在Flutter中我们可以配置自定义的字体，主要分为两部分：

- 首先在 pubspec.yaml 中声明它们，以确保它们会打包到应用程序中。

```
flutter:
 fonts:
 - family: Raleway
 fonts:
 - asset: assets/fonts/Raleway-Regular.ttf
 - asset: assets/fonts/Raleway-Medium.ttf
 weight: 500
 - asset: assets/fonts/Raleway-SemiBold.ttf
 weight: 600
  - family: AbrilFatface
  fonts:
  - asset: assets/fonts/abrilfatface/AbrilFatface-Regular.ttf
```

- 然后通过 TextStyle 属性使用字体。

```
// 声明文本样式
const textStyle = const TextStyle(
 fontFamily: 'Raleway',
);

// 使用文本样式
var buttonText = const Text(
 "Use the font for this text",
 style: textStyle,
 );
```