

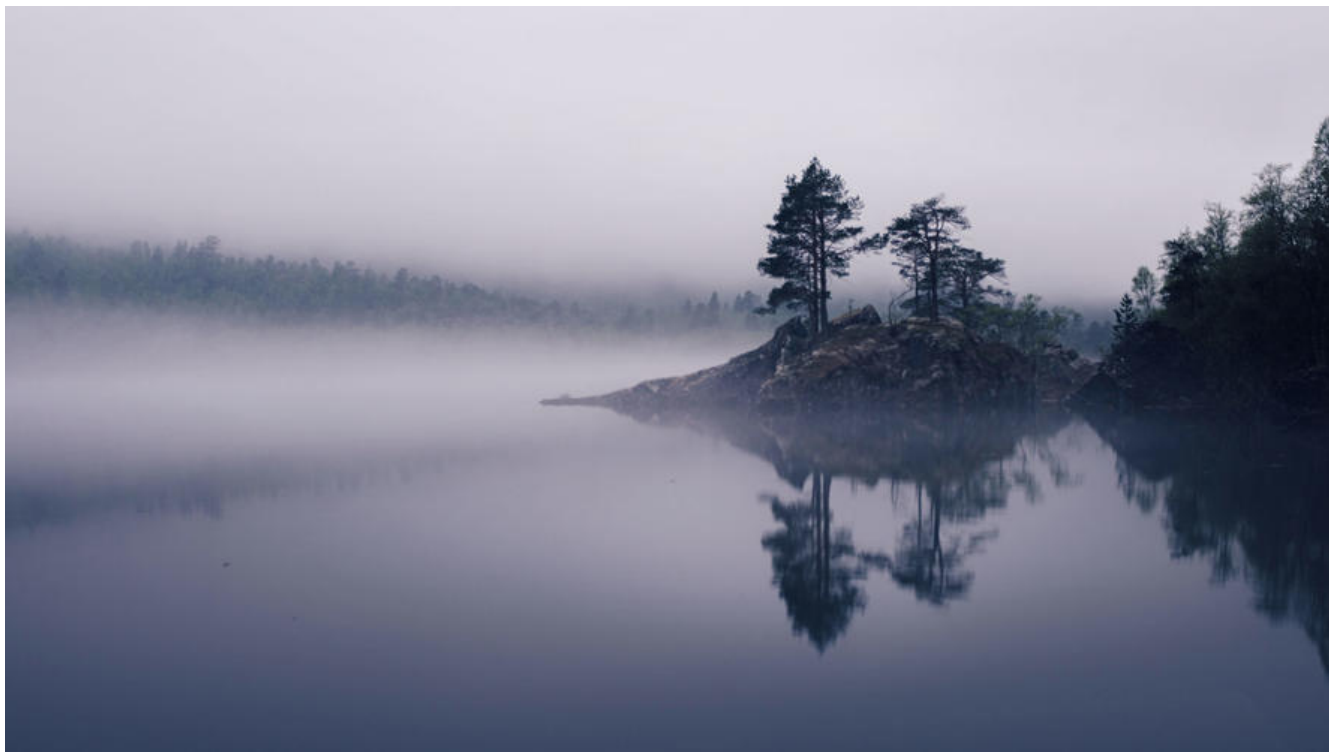
一起来学 Flutter 之文本组件、容器组件（一）

作者: [santa9527](#)

原文链接: <https://ld246.com/article/1568387056030>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



写在前面：

最近新接触到Flutter混合开发，开始学习强大的Flutter，这是第一篇Flutter教程。关于Flutter，在这里就不多介绍了，网上有很多，不过关于Flutter的安装，本人确实踩了不少坑，差点就从安装到放弃。后面有时间我可以单独出一篇教程讲述Flutter的安装过程。

学习Flutter之前，一定要告诉自己：Flutter一切皆组件。本节先认识一下Flutter的基本组件：文本组和容器组件。

1.文本组件 (Text Widget)

首先，我们构建一个Flutter基本方法：

```
import 'package:flutter/material.dart';
void main () => runApp(TextApp());

class TextApp extends StatelessWidget{
  @override
  Widget build(BuildContext context ){
    return MaterialApp(
      title:'Text widget',
      home:Scaffold(
        body:Center(
          child:Text('你好 Santa')
        ),
      ),
    );
  }
}
```

这里的Text()就是一个文本组件，熟悉Dart语言的都知道 new对象时，new关键字是可以省略的，所以在后面的学习中，所有对象或者方法的新都会被我省略。

关于文本组件的属性，常用的主要有以下几个：

TextAlign对齐属性

常用属性值：

- center：文本居中对齐
- right：左对齐
- left：右对齐
- start：从开始位置对齐
- end：从结束位置对齐

具体的对齐效果，大家可以自行试验，下面奉上实例代码

```
child:Text('你好 Santa',  
  textAlign: TextAlign.center)
```

maxLines行数显示属性

```
child:Text('你好 Santa',  
  textAlign: TextAlign.center,  
  maxLines: 2,)
```

这里maxLines值是多少，文本就最多显示多少行。

overflow属性

当文本较多，而又无法全部显示时，就用到了overflow属性

常用属性值：

- clip：直接去掉多余的。
- ellipsis：多余的用...表示。
- fade：多余的用渐变表示，但是效果太丑，个人不太喜欢。

```
child:Text('你好 Santa',  
  textAlign: TextAlign.center,  
  maxLines: 2,  
  overflow: TextOverflow.ellipsis,)
```

style属性

这里的style属性就类似于CSS中的style属性，主要是一些字体的美化功能，属性较多，大家可以自行试验,但是需要注意一点的是，这里的style对应的不再是值，而是需要新给一个textStyle组件，在组件给定各种属性，这也是Flutter的一个特点吧，就是组件里面套组件：

```
child:Text('你好 Santa',  
  textAlign: TextAlign.center,  
  maxLines: 2,  
  overflow: TextOverflow.ellipsis,
```

```
        style: TextStyle(
          fontSize:25.0,
          color:Color.fromARGB(255, 255, 255, 255),
          decoration:TextDecoration.underline,
          decorationStyle:TextDecorationStyle.solid,
        ),)
```

以上就是Flutter文本组件的一些常用属性和常规用法。

2.容器组件

容器组件就有点类似与HTML中的 `<div>` 标签和 `` 标签，可以用来分割界面，将其他组件放入定容器中：

首先，看下容器组件的基本写法：

```
import 'package:flutter/material.dart';
void main () => runApp(containerApp());

class containerApp extends StatelessWidget{
  @override
  Widget build(BuildContext context ){
    return MaterialApp(
      title:'Text widget',
      home:Scaffold(
        body:Center(
          //在页面中放入一个容器组件，再向容器组件中放入一个文本组件
          child:Container(
            child:Text('你好 Santa'),
          ),
        ),
      ),
    );
  }
}
```

Alignment属性

注意，该属性主要针对容器组件的子组件，而不是容器组件本身的布局属性也就是只对child下的组件效。

- bottomCenter:下部居中对齐。
- botomLeft: 下部左对齐。
- bottomRight: 下部右对齐。
- center: 纵横双向居中对齐。
- centerLeft: 纵向居中横向居左对齐。
- centerRight: 纵向居中横向居右对齐。
- topLeft: 顶部左侧对齐。
- topCenter: 顶部居中对齐。
- topRight: 顶部居左对齐。

```
child:Container(  
  child:Text('你好 Santa'),  
  alignment: Alignment.center,)
```

同时，还可以设置容器的大小，颜色等属性

```
child:Container(  
  child:Text('你好 Santa'),  
  alignment: Alignment.center,  
  width:300.0,  
  height:300.0,  
  color: Colors.lightBlue,)
```

另外，容器还具有边距属性，padding和margin，与CSS中的定义一样，padding是容器和child子件之间的距离，而margin则是容器和其外部组件之间的距离。

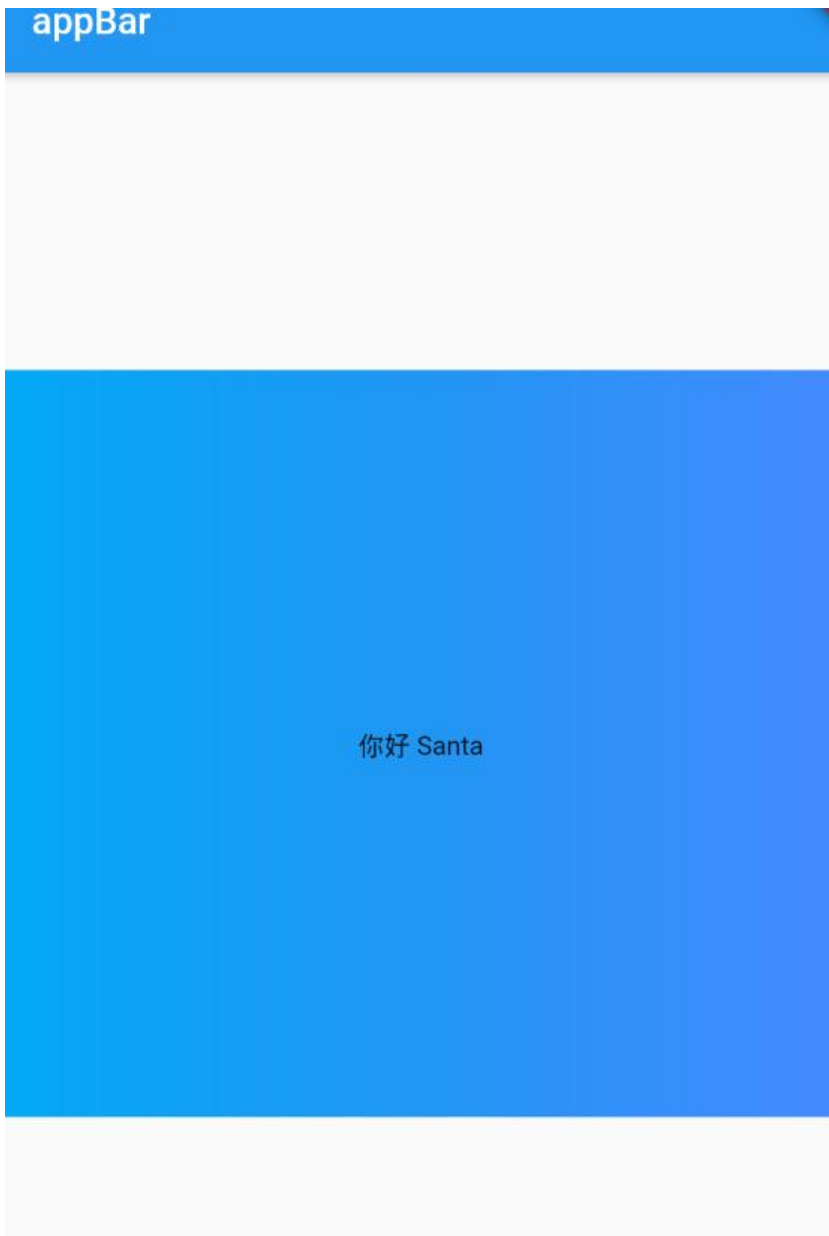
```
child:Container(  
  child:Text('你好 Santa'),  
  alignment: Alignment.center,  
  width:300.0,  
  height:300.0,  
  color: Colors.lightBlue,  
  //需要注意一点的是，两个属性值需要new一个新的常量来对应属性值  
  //Flutter中，组件的属性值通常是double类型，而不是int，所以10应该写成10.0  
  padding:const EdgeInsets.fromLTRB(20.0,20.0,10.0,10.0),  
  margin: const EdgeInsets.all(10.0),)
```

decoration属性

decoration是 container 的修饰器，主要的功能是设置背景和边框。

比如我们现在要给容器加一个渐变的背景色，则：

```
child: Container(  
  child: Text("你好 Santa"),  
  width:500.0,  
  height:400.0,  
  decoration:new BoxDecoration(  
    gradient:const LinearGradient(  
      colors:[Colors.lightBlue,Colors.blue,Colors.blueAccent]  
    )  
  ),  
),
```



但是需要注意的是，此处的decoration属性会colors属性冲突，二者取其一。

border边框属性

可以通过border属性，为容器设置边框属性

```
border:Border.all(width:2.0,color:Colors.red)
```

以上就是Flutter的文本组件和容器组件，第一次写Flutter技术文章，如有错误，请指正，谢谢！