Flutter 基础布局Widgets之Padding、Offstage详解


若数 LV3
2019年04月29日 21:23 · 阅读 2915

关注

Padding概述

Padding作为一种向内扩充，即用来产生间距的组件，使用的方式也很简单就是是设置内边距属性，产生内边距的空白区域，使用的成本比Container要低一些，可以替代的话推荐使用Padding

Padding构造函数

kotlin 复制代码

```
const Padding({
    Key key,
    @required this.padding,
    Widget child,
  }
```

- padding 定义 `padding` 的为 `EdgeInsetsGeometry` ， 一般使用 `EdgeInsets`
- child 即需要扩充的组件

Padding示例 `EdgeInsets` 多种使用方式

php 复制代码

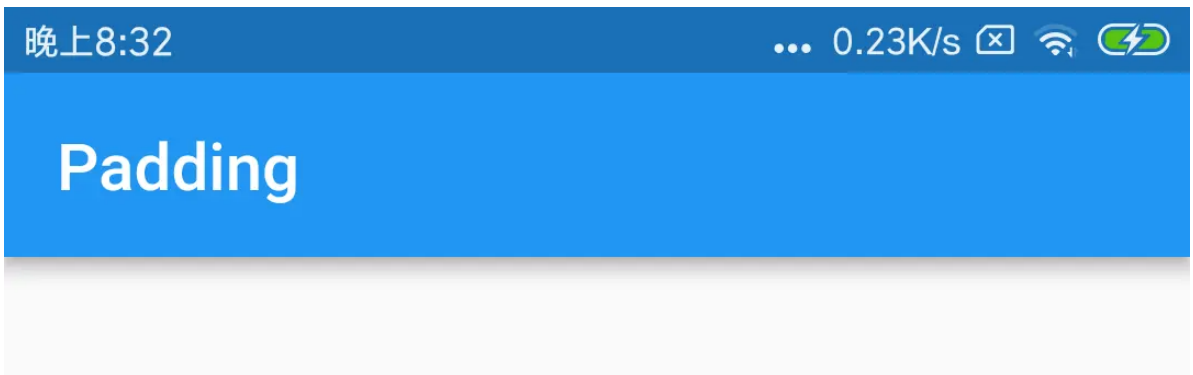
```
// padding  给子节点补白，即内填充操作

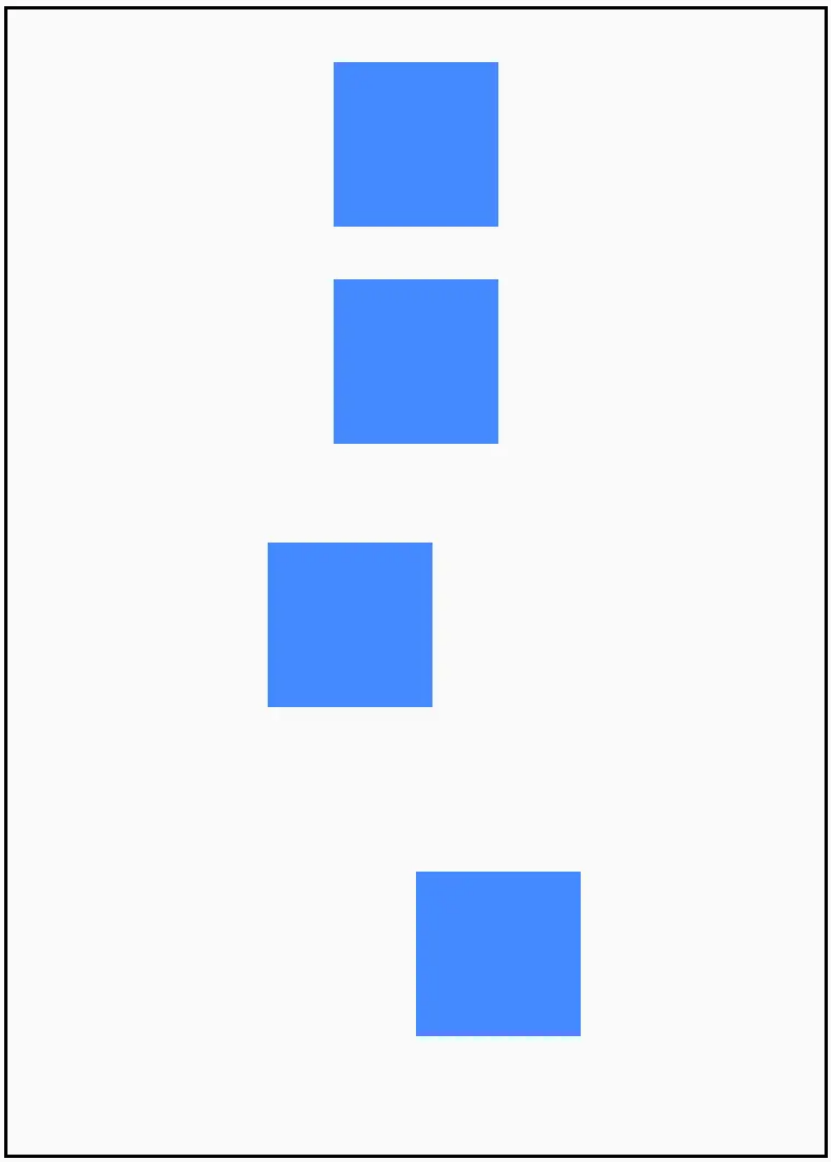
import 'package:flutter/material.dart';

class PaddingLearn extends StatelessWidget {
  @override
  final boxContainer = Container(
    color: Colors.blueAccent,
    height: 50,
    width: 50,
  );
  Widget build(BuildContext context) {
    return new Scaffold(
        appBar: AppBar(title: Text('Padding')),
        body: new Center(
```

```
            child: Container(
              decoration: BoxDecoration(
                border: Border.all(),
              ),
              width: 250,
              height: 350,
              // 构建一个盒子，在盒子里测试padding的属性
              child: new Column(
                children: <Widget>[
                  Padding(
                    // 上下左右全部统一补白
                    padding: EdgeInsets.all(16),
                    child: boxContainer,
                  ),
                  Padding(
                      // 垂直和水平方向补白
                      padding: EdgeInsets.symmetric(vertical: 0, horizontal: 30),
                      child: boxContainer
                  ),
                  Padding(
                      // 上、下、左、右 EdgeInsets.fromLTRB(double left, double top, double righ
                      padding: EdgeInsets.fromLTRB(0, 30, 40, 50),
                      child: boxContainer
                  ),
                  Padding(
                      // 仅仅填充一个方向 比如left
                      padding: EdgeInsets.only(left: 50),
                      child: boxContainer
                  )
                ]
              ),
          )
        )
      );
    }
}
```

Padding示例效果：





php 复制代码

```
// Offstage

import 'package:flutter/material.dart';

class OffstageLearn extends StatefulWidget {
  @override
```

```
    OffstageLearnState createState() => OffstageLearnState();

}

class OffstageLearnState extends State<OffstageLearn> {
  bool _offstage = false;
  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: AppBar(
        title: Text('Offstage'),
      ),
      body: Column(children: <Widget>[
        // 视觉上隐藏小部件  _offstage 为真即隐藏
        Offstage(
          offstage: _offstage,
          child: Container(
            decoration: BoxDecoration(
              gradient: LinearGradient(colors: [Colors.blue, Colors.purple])
            ),
            width: 300,
            height: 400,
          ),
        ),
        IconButton(
          icon: Icon(Icons.accessible),
          onPressed: (){
            setState(() {
             _offstage = !_offstage;
            });
          },
        ),
      ],)
    );
  }
}
```

Offstage概述

Offstage的功能即在视觉上隐藏Widget，设定参数为 `_offstage` ，为 `true` 即隐藏，反之则相反如果是 `true` 的，则将子元素放置在树中，但是不绘制任何内容，不让子元素用于hit测试，也不占用父元素中的任何空间。如果为 `false` ，则将子元素作为正常值包含在树中。

Offstage构造函数

kotlin 复制代码

```
``` const Offstage({ Key key, this.offstage = true, Widget child })```
```

- _offstage 即控制是否隐藏子元素

Offstage对比示例

_offstage=false 时：

php 复制代码

```
// Offstage

import 'package:flutter/material.dart';

class OffstageLearn extends StatefulWidget {
  @override
  OffstageLearnState createState() => OffstageLearnState();

}

class OffstageLearnState extends State<OffstageLearn> {
  bool _offstage = true;
  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: AppBar(
        title: Text('Offstage'),
      ),
      body: Column(children: <Widget>[
        // 视觉上隐藏小部件  _offstage 为真即隐藏
        Offstage(
          offstage: _offstage,
          child: Container(
            decoration: BoxDecoration(
              gradient: LinearGradient(colors: [Colors.blue, Colors.purple])
            ),
            width: 300,
            height: 400,
          ),
        ),
        IconButton(
          icon: Icon(Icons.accessible),
          onPressed: (){
            setState(() {
             _offstage = !_offstage;
            });
          },
        ),
```
```

```
          ],)
        );
      }
    }
```

效果如下：

_offstage=true 时：

php 复制代码

```
// Offstage

import 'package:flutter/material.dart';

class OffstageLearn extends StatefulWidget {
  @override
  OffstageLearnState createState() => OffstageLearnState();

}

class OffstageLearnState extends State<OffstageLearn> {
  bool _offstage = true;
  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: AppBar(
        title: Text('Offstage'),
      ),
      body: Column(children: <Widget>[
        // 视觉上隐藏小部件  _offstage 为真即隐藏
        Offstage(
          offstage: _offstage,
          child: Container(
            decoration: BoxDecoration(
              gradient: LinearGradient(colors: [Colors.blue, Colors.purple])
            ),
            width: 300,
            height: 400,
          ),
        ),
        IconButton(
          icon: Icon(Icons.accessible),
          onPressed: (){
            setState(() {
             _offstage = !_offstage;
            });
          },
```

```
            ),
          ],)
        );
      }
    }
```

效果如下：


@稀土掘金技术社区