

一、简介

流式布局是指页面元素的宽度可以按照屏幕分辨率进行适配调整，但整体布局不变在Flutter中的流式布局是Wrap。流式布局会将超出屏幕范围的Widget自动换行

二、Wrap

Wrap需要设置主轴方向，默认为水平方向

1、构造方法

```
1 class Wrap extends MultiChildRenderObjectWidget {
2   Wrap({
3     Key key,
4     this.direction = Axis.horizontal,
5     this.alignment = WrapAlignment.start,
6     this.spacing = 0.0,
7     this.runAlignment = WrapAlignment.start,
8     this.runSpacing = 0.0,
9     this.crossAxisAlignment = WrapCrossAlignment.start,
10    this.textDirection,
11    this.verticalDirection = VerticalDirection.down,
12    List<Widget> children = const <Widget>[],
13  }) : super(key: key, children: children);
14    ...
15 }
```

参数名字	参数类型	意义	必选 or 可选
key	Key	Widget 的标识	可选
direction	Axis	主轴的方向 默认是 Axis.horizontal	可选
alignment	WrapAlignment	子Widget 在主轴上的对齐方式，默认值为 WrapAlignment.start WrapAlignment 的值和 MainAxisAlignment 的一样	可选
runAlignment	WrapAlignment	Wrap 会自动换成或换列，runAlignment 就是每行或每列的对齐方式，如果主轴为水平方向，就是每行，如果主轴为竖直方向，就是每列，默认值为 WrapAlignment.start WrapAlignment 的值和 MainAxisAlignment 的一样	可选
spacing	double	每行或每列之间的间距	可选

runSpacing	double	默认是0.0	可选
crossAxisAlignment	WrapCrossAlignment	子Widget 在交叉轴上的对齐方式, WrapCrossAlignment.start WrapCrossAlignment 的值和 MainAxisAlignment 的一样	可选
textDirection	TextDirection	表示 子Widget 在主轴方向上的布局顺序	可选
verticalDirection	VerticalDirection	表示 子Widget 在交叉轴方向上的布局顺序	可选
children	List< Widget>	Wrap布局 里排列的内容	可选

2、使用

```

1 import 'package:flutter/material.dart';
2
3 main() => runApp(new WrapWidget());
4
5 class WrapWidget extends StatelessWidget {
6   @override
7   Widget build(BuildContext context) {
8     return new MaterialApp(
9       title: 'Test',
10      home: new Scaffold(
11        appBar: new AppBar(title: new Text('Flutter 布局Widget -- 流式布局')),
12        body: Wrap(
13          direction: Axis.horizontal,
14          spacing: 8.0, // 主轴 方向间距
15          runSpacing: 12.0, // 交叉轴 方向间距
16          alignment: WrapAlignment.center,
17          runAlignment: WrapAlignment.start,
18          children: <Widget>[
19            new Chip(
20              avatar: new CircleAvatar(
21                backgroundColor: Colors.blue, child: Text('A')),
22              label: new Text('AAAAAAA'),
23            ),
24            new Chip(
25              avatar: new CircleAvatar(
26                backgroundColor: Colors.blue, child: Text('M')),
27              label: new Text('BBBBBB'),
28            ),
29            new Chip(

```

```
30  avatar: new CircleAvatar(  
31    backgroundColor: Colors.blue, child: Text('H')),  
32    label: new Text('CCCCCCCC'),  
33  ),  
34  new Chip(  
35    avatar: new CircleAvatar(  
36      backgroundColor: Colors.blue, child: Text('J')),  
37      label: new Text('DDDDDDDD'),  
38    ),  
39    new Chip(  
40      avatar: new CircleAvatar(  
41        backgroundColor: Colors.blue, child: Text('J')),  
42        label: new Text('EEEEEEEE'),  
43      ),  
44      new Chip(  
45        avatar: new CircleAvatar(  
46          backgroundColor: Colors.blue, child: Text('J')),  
47          label: new Text('FFFFFFFFFFFFFFFF'),  
48        ),  
49      ],  
50    ));  
51  }  
52 }
```

12:16



DPUG

Flutter 布局Widget -- 流式布局

A

AAAAAAAAA

M

BBBBBBB

H

CCCCCCCCC

J

DDDDDDDD

J

EEEEEEEE

J

FFFFFFFFFFFFFFFF