

Flutter 基础Widget —— 菜单按钮

本节讲 Flutter 菜单按钮的使用。

菜单按钮的 Widget 是：[PopupMenuButton](https://docs.flutter.io/flutter/material/PopupMenuButton-class.html)
(<https://docs.flutter.io/flutter/material/PopupMenuButton-class.html>)

PopupMenuButton

PopupMenuButton 是一个按钮，当你点击按钮的时候，就会弹出类似 Android 选项菜单的框。

代码所在位置

flutter_widget_demo/lib/menu/PopupMenuButtonWidget.dart

PopupMenuButton 的快速上手

PopupMenuButton 是点击后会弹出菜单选项的按钮，其 child 参数是用来显示按钮的内容，itemBuilder 参数是用来显示弹出的菜单，onSelected 参数用于处理菜单点击事件：

```

PopupMenuButton<MenuItem>(
  child: Text('更多'),
  onPressed: (MenuItem result) {
    print('click '+result.toString());
  },
  itemBuilder: (BuildContext context) =>
<PopupMenuEntry<MenuItem>>[
  const PopupMenuItem<MenuItem>(
    value: MenuItem.menuA,
    child: Text('menu A'),
  ),
  const PopupMenuItem<MenuItem>(
    value: MenuItem.menuB,
    child: Text('menu B'),
  ),
  const PopupMenuItem<MenuItem>(
    value: MenuItem.menuC,
    child: Text('menu C'),
  ),
  const PopupMenuItem<MenuItem>(
    value: MenuItem.menuD,
    child: Text('menu D'),
  ),
],
)

```

PopupMenuButton 在一个页面使用的完整 Demo 如下：

```

import 'package:flutter/material.dart';

void main() => runApp(PopupMenuButtonWidget());

class PopupMenuButtonWidget extends

```

```
StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: "Flutter Demo",
      theme: ThemeData(
        primaryColor: Colors.blue,
      ),
      home: Scaffold(
        appBar: AppBar(title: Text("Flutter UI基础Widget -- 按钮")),
        body: PopupMenuButton<MenuItem>(
          child: Text('更多'),
          onSelect: (MenuItem result) {
            print('click '+result.toString());
          },
          itemBuilder: (BuildContext context)
=> <PopupMenuEntry<MenuItem>>[
            const PopupMenuItem<MenuItem>(
              value: MenuItem.menuA,
              child: Text('menu A'),
            ),
            const PopupMenuItem<MenuItem>(
              value: MenuItem.menuB,
              child: Text('menu B'),
            ),
            const PopupMenuItem<MenuItem>(
              value: MenuItem.menuC,
              child: Text('menu C'),
            ),
            const PopupMenuItem<MenuItem>(
              value: MenuItem.menuD,
              child: Text('menu D'),
            ),
          ],
        ),
      ),
    );
  }
}
```

```

),
],
)),
);
}
}

enum MenuItem { menuA, menuB, menuC, menuD }

```

运行后有一个更多的按钮，点击这个按钮，会展开菜单：



菜单选择

如何知道选中了哪个菜单呢？

在 `onSelected` 里会返回选中的菜单：

```
onSelected: (MenuItem result) {  
  print('click '+result.toString());  
},
```

PopupMenuButton 的构造函数及参数说明

PopupMenuButton 的构造函数为：

```
class PopupMenuButton<T> extends StatefulWidget {  
  const PopupMenuButton({  
    Key key,  
    @required this.itemBuilder,  
    this.initialValue,  
    this.onSelected,  
    this.onCanceled,  
    this.tooltip,  
    this.elevation = 8.0,  
    this.padding = const EdgeInsets.all(8.0),  
    this.child,  
    this.icon,  
    this.offset = Offset.zero,  
  }) : assert(itemBuilder != null),  
        assert(offset != null),  
        assert(!(child != null && icon != null)),  
        // fails if passed both parameters  
        super(key: key);  
  ...  
}
```

参数名字

参数类型

意义

必
选
or
可

key	Key	Widget 的标识	选 可 选
itemBuilder	PopupMenuBuilder	要显示的菜单	必 选
initialValue	T	初始化的菜单项 如果有，在菜单打开时会突出显示。	可 选
onSelected	PopupMenuItemSelected	当用户从此按钮创建的弹出菜单中选择一个值时调用。	可 选
onCanceled	PopupMenuItemSelected	当用户在不选择项目的情况下关闭弹出菜单时调用。	可 选
tooltip	String	当长按时显示的文本，用于盲人辅助模式下	可 选
elevation	double	Button 相对于其父级放置的z坐标，这可以控制 menu 下的阴影大小 默认值为8.0，该值必须 ≥ 0	可 选
padding	EdgeInsetsGeometry	menu 的内边距	可 选
child	Widget	这个按钮里显示的内容	可 选
icon	Icon	这个按钮里显示的 Icon Widget，前面在 图片 框和Icon 一节讲过如何使用 Icon Widget	可 选
offset	Offset	menu 相对按钮的偏移量	可 选