

Flutter 自定义 Dialog

主讲教师：（大地）

合作网站：www.itying.com （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html>

一、Flutter 自定义 Dialog.....	1
二、Flutter 定时器.....	3
三、Flutter 定时器结合 Dialog.....	4

一、Flutter 自定义 Dialog

自定义 Dialog 对象，需要继承 Dialog 类，尽管 Dialog 提供了 child 参数可以用来写视图界面，但是往往会达不到我们想要的效果，因为默认的 Dialog 背景框是满屏的。如果我们想完全定义界面，就需要重写 build 函数。下面我们通过两个案例给大家演示一下 Dialog 的使用。





```
import 'package:flutter/material.dart';

class LoadingDialog extends Dialog {
  final String text;

  LoadingDialog(this.text);
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return new Material(
      //创建透明层
      type: MaterialType.transparency, //透明类型
      child: new Center(
        child: Container(
          width: 300,
          height: 200,
          color: Colors.white,
          child: Column(
            children: <Widget>[
              Padding(
                padding: EdgeInsets.all(10),
                child: Stack(
                  children: <Widget>[
```

```
Align(  
  alignment: Alignment.center,  
  child: Text("关于我们"),  
)  
Align(  
  alignment: Alignment.centerRight,  
  child: InkWell(  
    child: Icon(Icons.close),  
    onTap: () {  
      Navigator.pop(context);  
    },  
  ),  
)  
],  
)  
,  
Divider(),  
Column(  
  children: <Widget>[  
    Container(  
      height: 40,  
      child: Text(this.text),  
    ),  
  ],  
)  
,  
)  
,  
)  
,  
)  
);  
}
```

二、Flutter 定时器

```
import 'dart:async';

_showTimer(context) {
  var timer;
  timer = Timer.periodic(
    Duration(milliseconds: 1500), (t) {
      print('执行');
      Navigator.pop(context);
      t.cancel();
    });
}
```

三、Flutter 定时器结合 Dialog

```
import 'dart:async';

import 'package:flutter/material.dart';

class LoadingDialog extends Dialog {
  final String text;

  LoadingDialog(this.text);

  _showTimer(context) {
    var timer;
    timer = Timer.periodic(
      Duration(milliseconds: 1500), (t) {
        print('执行');
        Navigator.pop(context);
        t.cancel();
      });
  }

  @override
  Widget build(BuildContext context) {

    _showTimer(context);

    // TODO: implement build
  }
}
```

```
return new Material(  
  //创建透明层  
  type: MaterialType.transparency, //透明类型  
  child: new Center(  
    child: Container(  
      width: 300,  
      height: 200,  
      color: Colors.white,  
      child: Column(  
        children: <Widget>[  
          Padding(  
            padding: EdgeInsets.all(10),  
            child: Stack(  
              children: <Widget>[  
                Align(  
                  alignment: Alignment.center,  
                  child: Text("关于我们"),  
                ),  
                Align(  
                  alignment: Alignment.centerRight,  
                  child: InkWell(  
                    child: Icon(Icons.close),  
                    onTap: () {  
                      Navigator.pop(context);  
                    },  
                  ),  
                ),  
              ],  
            ),  
          ],  
        ),  
        Divider(),  
        Column(  
          children: <Widget>[  
            Container(  
              height: 40,  
              child: Text(this.text),  
            ),  
          ],  
        ),  
      ],  
    ),  
  ),  
),
```

```
);  
}  
}
```

四、Flutter Vscode 调试 Flutter 项目

1、Vscode 中打开 flutter 项目进行开发

2、运行 Flutter 项目

```
flutter run
```

r 键： 点击后热加载，也就算是重新加载吧。

p 键： 显示网格，这个可以很好的掌握布局情况，工作中很有用。

o 键： 切换 android 和 ios 的预览模式。

q 键： 退出调试预览模式。

3、Vscode 默认连不上第三方模拟器解决方案

cd 到对应夜神模拟器 D:\Program Files\Nox\bin 目录 然后运行

```
nox_adb.exe connect 127.0.0.1:62001
```