```
//此appDelegate与OC的AppDelegate类似，也是我们应用程序的入口对象
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?


    func application(application: UIApplication,
didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?)
-> Bool {
        // Override point for customization after application
launch.
        //1.打印一个helloworld
//        //print类似nslog
//        print("hello world");
//
        //2.把windows设置其他的背景颜色
        //2.1创建一个颜色对象
        let color=UIColor.greenColor();
        self.window=UIWindow(frame:UIScreen.mainScreen().bounds);
        //把window对象的背景颜色改为其他的颜色
        self.window?.backgroundColor=color;
        //3.创建controller的根视图控制器
        //a.用纯代码方式创建视图控制器
       // let vc=UIViewController();
        let firstVC=FirstViewController();
        self.window?.rootViewController=firstVC;
        //b.用storyboard故事板方式创建视图控制器
        //oc的写法
        //UIStoryboard* mainStoryboard = [UIStoryboard
storyboardWithName:@"MainStoryboard_iPhone" bundle:nil];
       // MainViewController *mainController = [mainStoryboard
instantiateViewControllerWithIdentifier:@"MainViewController"];
        //swift写法
//        let vc=UIStoryboard(name: "Main",bundle:
nil).instantiateViewControllerWithIdentifier("SecondVC") as
UIViewController
//
//       self.window?.rootViewController=vc

        self.window?.makeKeyAndVisible();


        return true
    }


//这是一个界面文件，需要继承与UIViewController
//import 表示导入Foundation库，类似于 #import
import UIKit
//swift不支持宏定义,所以用let或func替代
//uilabel标签值
```

```
let Lable_Tag = 1001
//颜色值
func RGBA (r:CGFloat, g:CGFloat, b:CGFloat, a:CGFloat)->UIColor {
    return UIColor(red: r/255.0, green: g/255.0, blue: b/255.0,
alpha: a) }

class FirstViewController: UIViewController {

    var _clickCount = 0;//clickCount并没有声明为int

    //定义一个函数 viewDidLoad函数是父类中的一个函数
    override func viewDidLoad() {
        //1.调用父类的viewDidload()方法
        //super表示父类
        super.viewDidLoad()

        // Do any additional setup after loading the view.
        NSLog("FirstViewController ViewDidLoad");
        //在界面上加一个UILabel
        //CGRect相当于之前的CGRectMake
        let rect=CGRect(x:0,y:0,width: 200,height: 50);
        //创建一个UILabel
        let myLabel=UILabel(frame:rect);
        myLabel.text="码农教育";
        myLabel.tag=Lable_Tag;
        self.view.addSubview(myLabel);

        //创建一个UIButton
        let myButton=UIButton(frame:CGRect(x:0,y: 100,width:
200,height:50));
        myButton.backgroundColor=UIColor.greenColor();
        //给myButton设置一个文字
        myButton.setTitle("点击我", forState: .Normal);
        //setTitle第一个不需要根标签
        //给myButton添加点击事件
        myButton.addTarget(self, action: #selector(clickMe(_:)),
forControlEvents:.TouchUpInside);
        //单击myButton按钮会调用响应事件
        self.view.addSubview(myButton);

    }
     //定义一个点击事件
    func clickMe(sender:UIButton){

        _clickCount+=1;
        print("点击我\(_clickCount)");
        //as 作为类型转换 !
        /* a 强制取值表达式（Forced-Value Expression)
        强制取值表达式用来获取某个目标表达式的值（该目标表达式的值必须不是
nil ）。它的形式如下：
```

```
        expression!

        如果该表达式的值不是nil，则返回对应的值。 否则，抛出运行时错误
 (runtime error) 。
         b.expression?

         后缀'?' 返回目标表达式的值，把它做为可选的参数传递给后续的表达式

         如果某个后缀表达式包含了可选链表达式，那么它的执行过程就比较特殊： 首
先先判断该表达式的值，如果是 nil，整个后缀表达式都返回 nil，如果该可选链的值不
是nil，则正常返回该后缀表达式的值（依次执行它的各个子表达式）。在这两种情况下，
该后缀表达式仍然是一个optional type
        */

        let myLable=self.view .viewWithTag(Lable_Tag) as! UILabel;
        myLable.text="点击我\(_clickCount)";


        //对话框
        let alert=UIAlertController(title: "标题",message: "消
息",preferredStyle:  UIAlertControllerStyle.Alert)

        alert.addAction(UIAlertAction(title: "取消", style:
UIAlertActionStyle.Default, handler:{ (UIAlertAction) in
            print("no nothing")
        }) )// do not handle cancel, just dismiss

        alert.addAction(UIAlertAction(title: "确定", style:
UIAlertActionStyle.Default, handler: { (UIAlertAction) in
            print("no nothing")
        }))// do not handle cancel, just dismiss


        self.presentViewController(alert, animated: true,
completion: nil)

    }
    //定义一个点击事件
    func clickMe1(sender:UIButton){
        let vVc:SecondViewController=SecondViewController()
        self.presentViewController(vVc, animated: true, completion:
nil)

    }
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


    /*
```

```
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do
a little preparation before navigation
    override func prepareForSegue(segue: UIStoryboardSegue, sender:
AnyObject?) {
        // Get the new view controller using
segue.destinationViewController.
        // Pass the selected object to the new view controller.
    }
    */

import UIKit

class SecondViewController: UIViewController {
    //代码中的视图控件与storyboard建立连接
    @IBOutlet weak var _Btn:UIButton!
    @IBOutlet weak var _ImgView:UIImageView!

    var _flag:Bool=false

    override func viewDidLoad() {
        super.viewDidLoad()

        // Do any additional setup after loading the view.
        _Btn.setTitle("改变UIImageView背景颜色", forState:
UIControlState.Normal)
        _ImgView.backgroundColor=UIColor.blueColor();




    }
    @IBAction func BtnClick(sender:UIButton){

        if (!_flag){
            _ImgView.backgroundColor=UIColor.redColor();
        }else{
            _ImgView.backgroundColor=UIColor.greenColor();
        }
        _flag = !_flag


    }
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


    /*
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do
```

```
a little preparation before navigation
    override func prepareForSegue(segue: UIStoryboardSegue, sender:
AnyObject?) {
        // Get the new view controller using
segue.destinationViewController.
        // Pass the selected object to the new view controller.
    }
    */

}
```