UISwitch控件

**ViewController.h**

@interface ViewController : UIViewController

{

//定义一个开关控件

//可以进行状态的改变

//开；关 两种状态可以切换

//所有UIKit框架库中的控件均以UI开头

//苹果官方的控件都定义在UIKit框架库中

UISwitch\* \_mySwitch;

}

@property (retain,nonatomic)UISwitch\* mySwitch;

@end

**ViewController.m**

@implementation ViewController

//同步属性和成员变量

@synthesize mySwitch = \_mySwitch;

- (void)viewDidLoad {

[super viewDidLoad];

// Do any additional setup after loading the view, typically from a nib.

//创建一个开关对象

//继承于UIView

\_mySwitch = [[UISwitch alloc]init];

//苹果官方的控件的位置设置

//位置X，Y的值可以改变

//宽度和高度值无法改变

\_mySwitch.frame = CGRectMake(100, 100, 80, 40);

//开关状态设置属性

//YES 开启状态

//NO 关闭状态

\_mySwitch.on = YES;

//也可以使用set函数

[\_mySwitch setOn:YES];

//设置开关状态

//P1 状态设置

//P2 是否开启动画效果

[\_mySwitch setOn:YES animated:YES];

[self.view addSubview:\_mySwitch];

//设置开启状态的风格颜色

[\_mySwitch setOnTintColor:[UIColor orangeColor]];

//设置安按钮的风格颜色

[\_mySwitch setThumbTintColor:[UIColor blueColor]];

//设置整体风格颜色

[\_mySwitch setTintColor:[UIColor purpleColor]];

//ps 按钮关闭时的白色其实是透明状态，显示的是下一层view的颜色；

//向开关控件添加事件函数

//P1 函数实现对象

//P2 函数对象

//P3 事件响应时的事件类型UIControlEventValueChange：状态发生变化时触发函数

[\_mySwitch addTarget:self action:@selector(swChange:) forControlEvents:UIControlEventValueChanged];

}

-(void)swChange:(UISwitch\*)sw

{

if(sw.on == YES){

NSLog(@"开关被打开");

}

else

{

NSLog(@"开关被关闭");

}

NSLog(@"开关状态变化");

}

- (void)didReceiveMemoryWarning {

[super didReceiveMemoryWarning];

// Dispose of any resources that can be recreated.

}

@end