步进器和分栏控件

**ViewController.h**

@interface ViewController : UIViewController

{

//定义步进器对象

//按照一定的数字来调整摸个数据

UIStepper\* \_stepper;

//分栏控制器定义

UISegmentedControl\* \_segControl;

}

@property (retain,nonatomic) UIStepper\* stepper;

@property (retain,nonatomic) UISegmentedControl\* segControl;

@end

**ViewController.m**

@implementation ViewController

@synthesize stepper = \_stepper;

@synthesize segControl = \_segControl;

- (void)viewDidLoad {

[super viewDidLoad];

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

//创建步进器对象

\_stepper = [[UIStepper alloc]init];

//设置位置，高度不能变更

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

//设置步进器的最小值

\_stepper.minimumValue = 0;

//设置步进器的最大值

\_stepper.maximumValue = 100;

//设置步进器的当前值，默认为0

\_stepper.value = 10;

//设置步进器步值，每次向前或者向后步进的步伐值

\_stepper.stepValue = 1;

//是否可以重复响应事件操作 yes情况下，可以长按持续响应；

\_stepper.autorepeat = YES;

//是否将步进结果通过事件函数响应出来 即按住时不调用事件函数，在抬起时调用

\_stepper.continuous = YES;

//添加事件函数

//1：函数实现体

//2：函数体

//3：事件值改变状态

[\_stepper addTarget:self action:@selector(stepChange) forControlEvents:UIControlEventValueChanged];

[self.view addSubview:\_stepper];

//创建分栏控件

\_segControl = [[UISegmentedControl alloc]init];

//设置控件位置，宽度可变，高度不变

\_segControl.frame = CGRectMake(10, 200, 300, 40);

//添加一个按钮元素

[\_segControl insertSegmentWithTitle:@"吃翔" atIndex:0 animated:YES];

//参数一：按钮选择文字

//P2：按钮的索引位置

//P3：是否有插入动画效果

//index设置为0 为最左边，如果2个index值相同，则靠后设置的在左边

//每个选项系统自动平均分配宽度

[\_segControl insertSegmentWithTitle:@"喝尿" atIndex:1 animated:YES];

[\_segControl insertSegmentWithTitle:@"噶屎" atIndex:2 animated:NO];

//当前默认按钮索引设置

\_segControl.selectedSegmentIndex = 0;

[\_segControl addTarget:self action:@selector(segChange)forControlEvents:UIControlEventValueChanged];

[self.view addSubview:\_segControl];

}

-(void)segChange

{

NSLog(@"%ld",\_segControl.selectedSegmentIndex);

}

-(void)stepChange

{

NSLog(@"step press =-= %f",\_stepper.value);

}

- (void)didReceiveMemoryWarning {

[super didReceiveMemoryWarning];

// Dispose of any resources that can be recreated.

}

@end