



UIKit基础（3）

这节课我会学到什么



学会UISwitch , UISlider , UISegementControl的使用

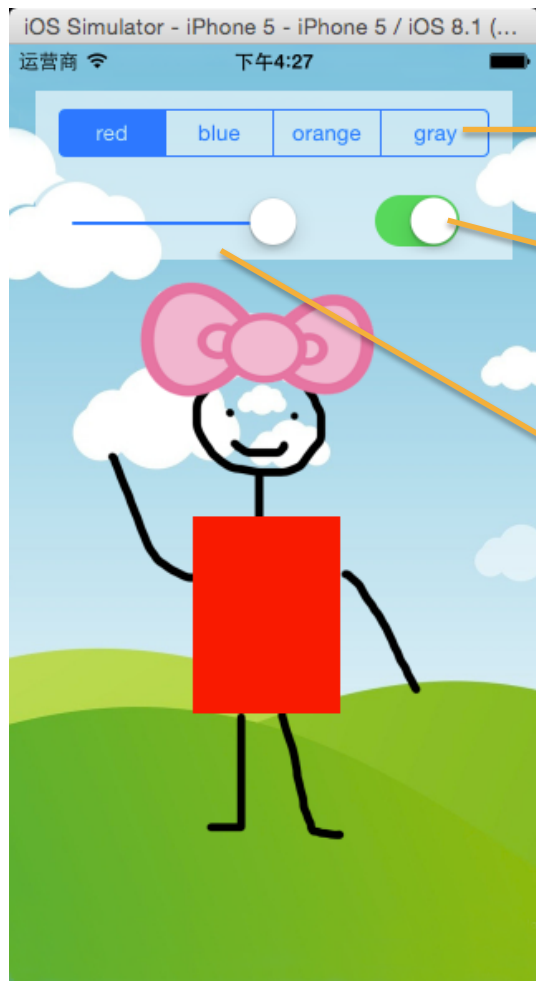
学会UITextView的使用

学会UIToolBar , UIImageView的使用

学会UIProgressView , UIStepper使用



练习：换衣服



分段控件UISegmentControl，这里我们使用它来控制小朋友的衣服颜色

开关UISwitch，这里我们使用它来控制小朋友的性别

滑块UISlider，用来控制衣服的透明度

开关控件UISwitch

开关控件(UISwitch), 它只有两种状态。



// UISwitch大小是固定的所以即使创建的时候不需要赋值宽高(即使赋值了也不起作用)

```
UISwitch *mySwitch = [[UISwitch alloc] init];
```

```
mySwitch.center = CGPointMake(100, 50);
```

```
mySwitch.onTintColor = [UIColor redColor]; // 开关打开时的颜色
```

```
mySwitch.thumbTintColor = [UIColor orangeColor]; // 开关钮的颜色
```

// 添加事件

```
[mySwitch addTarget:self action:@selector(switchValueChanged:)
```

```
forControlEvents:UIControlEventValueChanged];
```

```
[self.view addSubview:mySwitch];
```

// 开关切换事件

```
- (void)switchValueChanged:(UISwitch *)theSwitch {
```

```
    BOOL isOn = theSwitch.isOn;
```

```
    NSLog(@"State: %@", isOn ? @"ON" : @"OFF");
```

```
}
```

滑块控件UISlider

常用在选择一个参数值的场景。



```
// 创建
UISlider *slider = [[UISlider alloc] initWithFrame: CGRectMake(0, 0, 200, 20)];
slider.minimumValue = 0; // 最大值
slider.maximumValue = 100; // 最小值
slider.value = 50; // 当前值
slider.continuous = YES; // 是否连续触发滑动事件
[slider addTarget:self action:@selector(updateValue:)
forControlEvents:UIControlEventValueChanged];
[self.view addSubview:slider];

// 事件处理
-(void)updateValue:(id)sender
{
    NSLog(@"%g", ((UISlider*)sender).value);
}
```

分段控件UISegmentedControl



```
NSArray *segmentedArray = @[@"1",@"2",@"3",@"4"];  
//初始化UISegmentedControl  
UISegmentedControl *segmentedControl = [[UISegmentedControl  
alloc]initWithItems:segmentedArray];  
segmentedControl.frame = CGRectMake(60.0, 10.0, 200.0, 50.0);  
segmentedControl.selectedSegmentIndex = 2;//设置默认选择项索引  
segmentedControl.tintColor = [UIColor redColor];  
segmentedControl.momentary = YES;//设置在点击后是否恢复原样  
[segmentedControl addTarget:self action:@selector(pickOne:)  
forControlEvents:UIControlEventValueChanged]; // 设置事件  
[self.view addSubview:segmentedControl];  
  
- (void)pickOne:(UISegmentedControl *)sender  
{  
    NSLog(@"%d",sender.selectedSegmentIndex);  
}
```



分段设置UISegmentedControl



```
[segmentedControl setTitle:@"two" forSegmentAtIndex:1];//设置指定索引的标题  
[segmentedControl setImage:[UIImage imageNamed:@"colorize.png"] forSegmentAtIndex:  
3];//设置指定索引的图片
```

```
[segmentedControl insertSegmentWithImage:[UIImage imageNamed:@"wizard.png"]  
atIndex:2 animated:NO];//在指定索引插入一个选项并设置图片
```

```
[segmentedControl insertSegmentWithTitle:@"insert" atIndex:3 animated:NO];//在指定  
索引插入一个选项并设置标题
```

```
[segmentedControl removeSegmentAtIndex:0 animated:NO];//移除指定索引的选项
```

```
[segmentedControl setWidth:70.0 forSegmentAtIndex:2];//设置指定索引选项的宽度
```

```
[segmentedControl setContentOffset:CGSizeMake(10.0,10.0) forSegmentAtIndex:1];//设  
置选项中图片等的左上角的位置
```

```
[segmentedControl setEnabled:NO forSegmentAtIndex:3];//设置指定索引选项不可选
```

获取分段控件一些参数



```
//获取指定索引选项的图片 imageForSegmentAtIndex :
UIImageView *imageForSegmentAtIndex = [[UIImageView alloc] initWithImage:[segmentedControl
imageForSegmentAtIndex:1]];

//获取指定索引选项的标题titleForSegmentAtIndex
titleForSegmentAtIndex.text = [segmentedControl titleForSegmentAtIndex:0];

//获取总选项数segmentedControl.numberOfSegments
numberOfSegments.text = [NSString stringWithFormat:@"%d",segmentedControl.numberOfSegments];

//获取指定索引选项的宽度widthForSegmentAtIndex:
widthForSegmentAtIndex.text = [NSString stringWithFormat:@"%f",[segmentedControl widthForSegmentAtIndex:
2]];

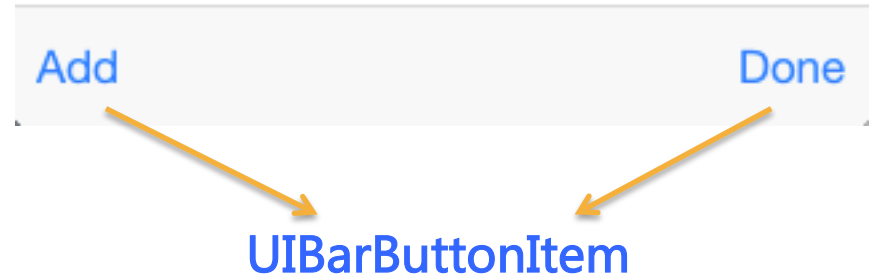
//判断指定索引选项是否可选
BOOL enableFlag = [segmentedControl isEnabledForSegmentAtIndex:3];
```


UISegmentedControl小结



- 初始化
 - initWithItems:
- 管理分段内容
 - setImage:forSegmentAtIndex:
 - imageForSegmentAtIndex:
 - setTitle:forSegmentAtIndex:
 - titleForSegmentAtIndex:
- 管理分段
 - insertSegmentWithImage:atIndex:animated:
 - insertSegmentWithTitle:atIndex:animated:
 - numberOfSegments property
 - removeAllSegments
 - removeSegmentAtIndex:animated:
 - selectedIndex property
- 管理分段的行为和外观
 - momentary property
 - segmentedControlStyle property
 - tintColor property
 - setEnabled:forSegmentAtIndex:
 - isEnabledForSegmentAtIndex:
 - setContentOffset:forSegmentAtIndex:
 - contentOffsetForSegmentAtIndex:
 - setWidth:forSegmentAtIndex:
 - widthForSegmentAtIndex:

- UIBarButtonItem初始化有以下几种方法



```
UIBarButtonItem *buttonImage =
```

```
UIBarButtonItem *buttonText =
```

[illegible]

2.使用系统自带默认样式初始化

```
UIBarButtonItem *myBookmarks =
```

```
[ [ UIBarButtonItem alloc ]initWithBarButtonSystemItem: UIBarButtonSystemItemBookmarks  
target: self  
action: @selector(mySelector:)  
];
```

3.自定义view作为item

```
UIBarButtonItem *customButton = [ [ UIBarButtonItem alloc ]  
initWithCustomView: myView ]; // 这里的myView可以是自定义的任何View
```

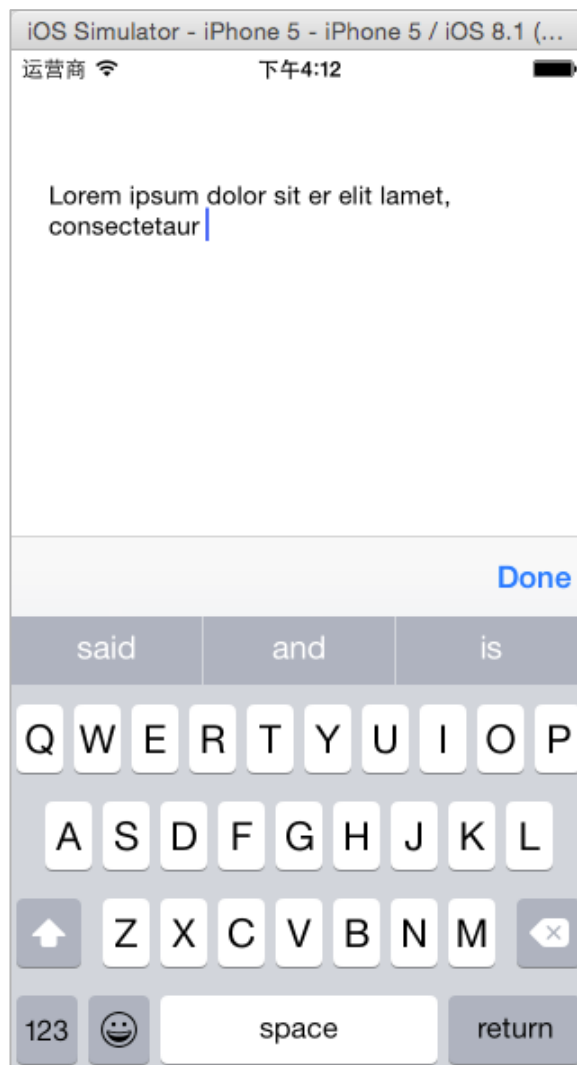
表：部分系统自带UIBarButtonItem样式

UIBarButtonItemDone	
UIBarButtonItemCancel	
UIBarButtonItemSearch	
UIBarButtonItemCamera	
UIBarButtonItemFlexibleSpace	空白，占用空间灵活（弹簧）

然后UIToolBar可以依靠这些已经创建好的Item们进行初始化:

```
UIToolbar *toolBar = [[UIToolbar alloc] initWithFrame:CGRectMake(0, 40, 320, 44)];  
// 通过设置items的属性值让toolBar显示前面创建好的item们  
// 注意items属性是一个由UIBarButtonItem组成的数组  
    toolBar.items = @[flex,item];  
// 将toolBar添加到视图上  
[self.view addSubview:toolBar];
```

多行文本控件UITextView



自定义键盘附件

适用于UITextField和UITextView

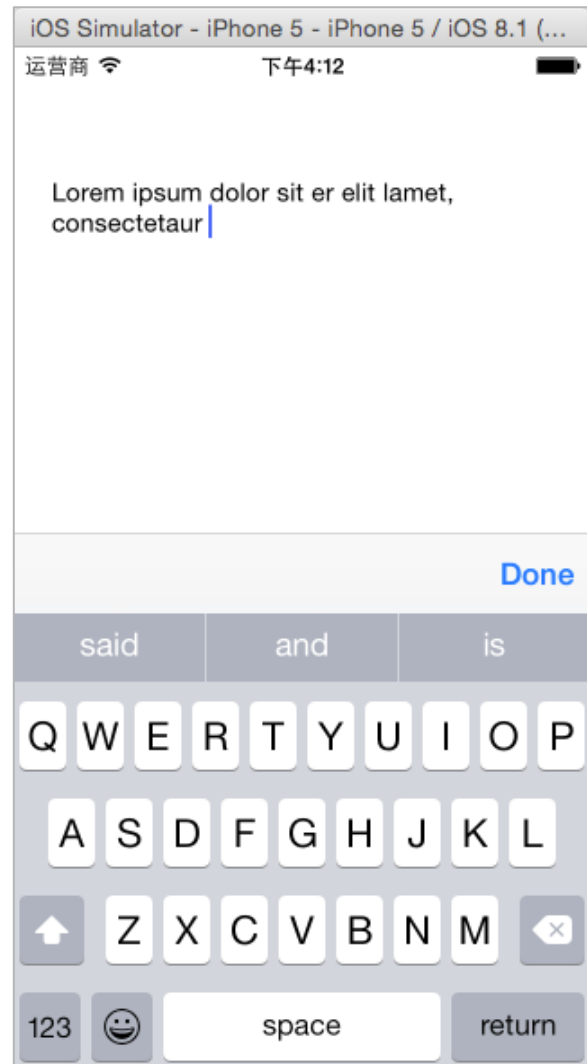
UITextField和UITextView有一个共同的属性叫做，我们可以通过给这个属性赋值定义键盘附件(通常是UIToolbar)，定义的附件可以在UITextField或者UITextView成为第一响应者的时候随着键盘一起弹出。

```
UIBarButtonItem *item = [[UIBarButtonItem  
alloc]initWithBarButtonSystemItem:UIBarButtonSys  
temItemDone target:self  
action:@selector(finishEdit:)];
```

```
UIBarButtonItem *flex = [[UIBarButtonItem  
alloc]initWithBarButtonSystemItem:UIBarButtonSys  
temItemFlexibleSpace target:nil action:nil];
```

```
UIToolbar *toolBar = [[UIToolbar  
alloc]initWithFrame:CGRectMake(0, 40, 320, 44)];  
toolBar.items = @[flex,item];
```

```
self.textView.inputAccessoryView = toolBar;
```



UITextView其他关键属性

- **inputView**(UITextField也有这个属性, 作用是一样的)

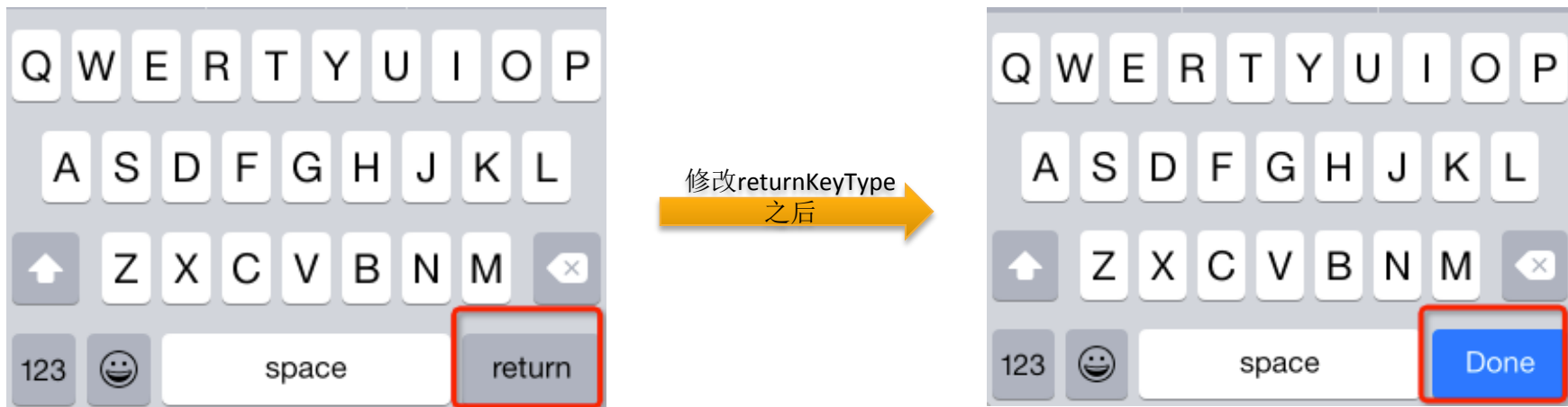
inputView就是显示键盘的view, 如果重写这个View则不再弹出键盘, 而是弹出自己的View. 如果想实现当某一控件变为第一响应者时不弹出键盘而是弹出我们自定义的界面, 那么我们就可以通过修改这个inputView来实现, 比如弹出一个日期选择器, 或者自定义的键盘

- **returnKeyType**

通过修改这个属性我们可以修改弹出键盘中return按钮的显示

- **- endEditing:**

这是一个UIView对UITextView的扩展方法, 可以让View内所有的输入项失去第一响应者 (UITextField同样适用)



练习：简易图片浏览器

(全代码实现)



图像视图UIImageView



UIImage和UIImageView

图像是iOS应用程序的强大工具。UIImage是NSObject的子类，Foundation框架的一部分。UIImage是一个简单的对象，它代表了显示图像所需要的数据。UIImageView是UIView的子类，但它是为在屏幕上绘制UIImage而设计的。

UIImage支持以下的格式：

- Graphic Interchange Format (.gif) ；
- Joint Photographic Experts Group (.jpg , .jpeg) ；
- Portable Network Graphic (.png) ；
- Tagged Image File Format (.tiff , .tif) ；
- Windows Bitmap Format (.bmp , .BMPf) ；
- Windows Icon Format (.ico) ；
- Windows Cursor (.cur) ；
- XWindow Bitmap (.xbm) 。

UIImageView的创建



当创建一个UIImageView，你可以选择使用标准的视图初始化方法 initWithFrame。然而，由于图像的图特性，iOS为我们提供了一个额外的初始化方法，它是 initWithImage。当你使用一个图像来初始化 UIImageView，它会自动把 UIImageView 的高度和宽度设置为 UIImage 的高度和宽度。例如，

```
UIImage *myImage=[UIImage imageNamed:@"sample.png"];
UIImageView *myImageView=[[UIImageView
alloc]initWithImage:myImage];
[self.view addSubview:myImageView];
```

如果使用 initWithFrame 进行 UIImageView 的初始化的话,可以通过 UIImageView 的 image 属性为其赋值图片,如:

```
myImageView.image = [UIImage imageNamed:@"sample.png"];
```

contentMode

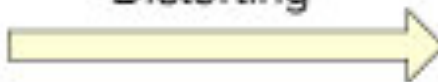
- 这其实是UIView的属性,但是在UIImageView中比较常用,主要用来设置图片的显示方式,如居中,拉伸等.

UIViewContentModeScaleToFill	默认情况下为此方式,会拉伸图片填充整个image,可能导致图片变形
UIViewContentModeScaleAspectFit	保持图片比例不变,而且全部显示在ImageView中,这意味着ImageView会有部分空白
UIViewContentModeScaleAspectFill	图片比例不变,但是是填充整个ImageView的,可能只有部分图片显示出来。
其它	

UIViewContentModeScaleToFill



Distorting



UIViewContentModeScaleAspectFit



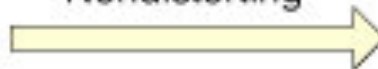
Nondistorting



`UIViewContentModeScaleAspectFill`



Nondistorting



使用UI ImageView播放动画



- UIImageView提供一个属性叫做animationImages,它应该是一个由UIImage组成的数组,然后设置animation一系列属性,如动画时间,动画重复次数等,就可以实现动画效果.

```
NSArray *imagesArray = @[[UIImage imageNamed:@"image1"],  
                          [UIImage imageNamed:@"image2"],  
                          [UIImage imageNamed:@"image3"],  
                          [UIImage imageNamed:@"image4"],  
                          [UIImage imageNamed:@"image5"]];
```

```
UIImageView *animationImageView = [[UIImageView  
alloc] initWithFrame:CGRectMake(0, 0, 131, 125)];  
animationImageView.animationImages = imagesArray;//将序列帧数组赋给  
UIImageView的animationImages属性  
animationImageView.animationDuration = 0.25;//设置动画时间  
animationImageView.animationRepeatCount = 0;//设置动画次数 0 表示无限  
[animationImageView startAnimating];//开始播放动画
```

练习：汤姆猫



UIImageView的内存问题



用UIImage加载图像的方法很多，最常用的是下面两种：

1.用imageNameNamed函数

```
[UIImage imageNamed:ImageName];
```

2.用contentsOfFile的方式加载，例如：

```
[[UIImage alloc] initWithContentsOfFile:[NSBundle mainBundle] pathForResource:  
@"yourImageName" ofType:@"png"]];
```

- 1) 用imageNameNamed的方式加载时，系统会先去检查系统缓存中是否存在该名字的图像，如果存在则直接返回。如果不存在该名字的图像，则系统会把图像Cache到内存，且不容易释放。这种机制适合于那些频繁用到界面贴图类的加载。
- 2) 使用contentsOfFile方式加载，不会经过系统缓存，直接从文件系统中加载并返回，使用后内存直接释放。适用于短时间内需要频繁加载一些一次性的图像的情况。

进度控件UIProgressView



UIProgressView提供了一个接口让你可以显示一个类似进度条的东西，这样就能让用户知道当前操作完成了多少。

```
UIProgressView* progressView = [[UIProgressView alloc ]
                                initWithFrame:CGRectMake(150.0,20.0,130.0,30.0)];

/*
UIProgressViewStyleDefault //标准进度条
UIProgressViewStyleBar     //深灰色进度条，用于工具栏中
*/
progressView.progressViewStyle = UIProgressViewStyleDefault;
progressView.progress = 0.5; // 进度 0.0到1.0之间的浮点数
[self.view addSubview:progressView ];
```

步进控件UIStepper



iOS5中新增的，它可以递进式输入数量。UIStepper继承自UIControl，它主要的事件是UIControlEventValueChanged，每当它的值改变了就会触发这个事件

```
UIStepper *stepper = [[UIStepper alloc] init];
stepper.minimumValue= 2; // 最小可以表示的值
stepper.maximumValue= 5; // 最大可以表示的值
stepper.stepValue= 2; // 每次增加或者减少的值
stepper.value= 3; // 当前值
stepper.center= CGPointMake(160, 240);
[stepper addTarget:self action:@selector(valueChanged:)
forControlEvents:UIControlEventsValueChanged];
[self.view addSubview:stepper];
```

另外有几个控制属性：

- continuous 控制是否持续触发UIControlEventValueChanged事件。默认YES，即当按住时每次值改变都触发一次UIControlEventValueChanged事件，否则只有在释放按钮时触发UIControlEventValueChanged事件。
- autorepeat 控制是否在按住是自动持续递增或递减。默认YES。
- wraps 控制值是否在[minimumValue,maximumValue]区间内循环。默认NO。

作业



1. 用UISlider实现对一个按钮位置的偏移控制，使得按钮可以到达屏幕上任意一个地方（代码实现）
 2. 使用代码实现穿衣服练习
 3. 自定义键盘附件实现类似微信回复的功能
 4. 自定义步进控件
 5. 开放性题：利用所学的控制自己构造一个工程，不要求全部用上，尽可能让你的界面看着顺眼，让你使用的控件能有正确的用途，如果有内容就更好。
-