

IOS 中的导航

➤ IOS 导航中常用的类：

1、UINavigationController：导航栏 bar；

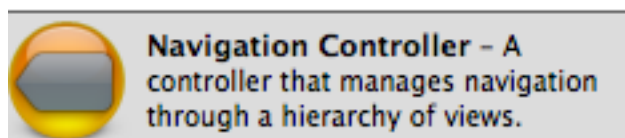
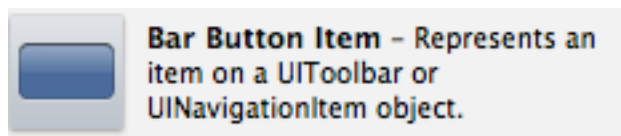
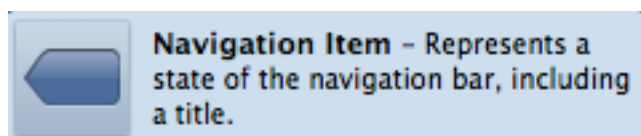
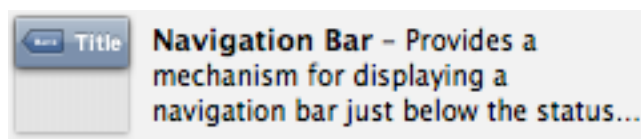
2、UINavigationController：导航栏集合；

所谓 UINavigationController 就可以理解为 Navigation Bar 中的内容，通过编辑 UINavigationController，我们可以使得在 Navigation Bar 中显示想要的东西，比如设置标题、添加按钮、返回按钮等；

3、UIBarButtonItem：导航按钮，需要添加到 UINavigationController 导航栏集合中；

4、UINavigationController：带有导航栏的视图控制器类；

对应控件库中的相关控件：



➤ Demo

1、使用代码生成以及简单的导航栏，并设置两个导航按钮（“左边”，“右边”）和标题；

```

//获取主屏幕的长宽
CGRect mainRect = [[UIScreen mainScreen]bounds];

//创建导航栏bar对象
UINavigationController *bar = [[UINavigationController alloc] initWithFrame:CGRectMakeMake
(0,0, mainRect.size.width, 50)];

//创建导航栏集合（存放标题和按钮）
UINavigationController *item = [[UINavigationController alloc] initWithTitle:@"这是代码
生成的"];

//创建左右两个按钮
UIBarButtonItem *left = [[UIBarButtonItem alloc]
initWithBarButtonSystemItem:UIBarButtonSystemItemAdd target:nil
action:nil];
UIBarButtonItem *right = [[UIBarButtonItem alloc]
initWithBarButtonSystemItem:UIBarButtonSystemItemDone target:nil
action:nil];

//将左右按钮添加到导航栏集合中
[item setLeftBarButtonItem:left];
[item setRightBarButtonItem:right];

//将导航栏集合添加到导航bar上
[bar pushViewController:item animated:YES];

//将整个导航bar添加到主视图上
[self.view addSubview:bar];

```

2、 使用控件库中的控件实现上述功能；

➤ UINavigationController 导航原理：

UINavigationController 是带有导航栏的控制器，常用此类作为根视图实现多个视图的切换。切换原理 UINavigationController 采用栈（stack）的数据结构管理所有的视图控制器，点击进入下一个视图时采用 push 方法压栈，返回时采用 pop 方法出栈获取上一个视图。

➤ Demo：三层视图的简单切换

关键代码：

- （1） 首先添加 UINavigationController，并将其作为根视图，方法是在 AppDelegate.h 中声明属性：

```

//定义UINavigationController类型的视图作为根视图
@property (retain, nonatomic) UINavigationController
*navController;

```

在AppDelegate.m中创建该导航栏视图，方法为：

```

- (BOOL)application:(UIApplication *)application
  didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[[UIWindow alloc] initWithFrame:[UIScreen mainScreen]
        bounds]] autorelease];
    // Override point for customization after application launch.

    //创建内容视图
    self.viewController = [[[XQXViewController alloc]
        initWithNibName:@"XQXViewController" bundle:nil] autorelease];

    //把内容视图添加到导航栏视图中
    self.navigationController = [[UINavigationController alloc]
        initWithRootViewController:self.viewController];

    [self.navigationController autorelease];

    //将导航栏视图添加到根视图中
    [self.window addSubview:self.navigationController.view];

    [self.window makeKeyAndVisible];
    return YES;
}

```

(2) 如何跳转：pushViewController 方法实现页面跳转，跳转的同时采用压栈的方式把目标视图压入导航栏栈中。

```
[self.navigationController pushViewController:controller animated:YES];
```

(3) 如何返回：采用上述方式进行视图切换时，导航栏左边自动出现上一个视图的标题按钮，点击可以返回上一个视图（出栈）；



如图中的返回按钮“第一个页面”，上一个视图设置了标题，该按钮显示标题信息，如果没设置，则显示“返回”

