



# 无限互联

## Infinite Interconnection

INFINITE  
INTERCONNECTION

无限互联是国内唯一一家**专注于**iPhone和iPad软件开发培训机构，到目前为止为各大公司输送了一大批优秀的iOS高级软件研发人才。随着iOS6系统的发布，我们也在当天陆续发布**国内首套完整**的iOS开发的视频教程，手把手教您写代码，从入门到熟练再到精通。

**高薪就业**是检验一家培训机构质量的唯一标准，我们的学员高薪就业是对我们最好的肯定，也是我们前进的最强烈的动力，我们感谢同学们的努力，感谢你们对我们的支持！我们也将免费为你们提供最好的就业后的技术支持！

亲爱的同学们，你们的**高薪就业**才是我们最大的成功！

<http://www.iphonetrain.com>



# 无限互联

## Infinite Interconnection

INFINITE CONNECTION

### iOS动画

主讲:

<http://www.iphonetrain.com>



## 本节目标

掌握**UIView**的基本用法

掌握**Block**语法使用动画

**CATransition**动画

掌握开源动画库**HMGLTransitions**





- 动画效果是iOS界面的重要特色，使用恰当能给用户视觉上的体验
- iOS上的动画实现有三种方式：
  - 1. UIView动画，这是最基本最常用的动画使用方式，主要改变 UIView的属性达到动画效果。(例如：渐隐/渐显)
  - 2. CATransition动画，主要是用于两个视图切换过渡的动画效果，系统内置了几种效果，我们可以直接使用。
  - 3. CAAnimation动画，这是ios的核心动画，结合绘图可以实现灵活多变的动画效果。



```
//标记动画块开始
[UIView beginAnimations:nil context:nil];
//定义动画加速和减速方式
[UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];
//以秒为单位指定动画时长
[UIView setAnimationDuration:0.5];
//设置代理
[UIView setAnimationDelegate:self];
//动画结束后回调方法
[UIView setAnimationDidStopSelector:
    @selector(animationDidStop:finished:context:)];

/*视图变化动作在这里*/

//标志动画块的结束
[UIView commitAnimations];
```



## ALPHA变换(淡进/淡出)

```
//淡入过程  
aView.alpha = 0.0;  
[UIView beginAnimations:nil context:nil];  
[UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];  
[UIView setAnimationDuration:0.5];  
[UIView setAnimationDelegate:self];  
aView.alpha = 1.0;  
[UIView commitAnimations];
```

```
//淡出过程  
aView.alpha = 1.0;  
[UIView beginAnimations:nil context:nil];  
[UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];  
[UIView setAnimationDuration:0.5];  
[UIView setAnimationDelegate:self];  
aView.alpha = 0.0;  
[UIView commitAnimations];
```



## 位置变换(位移)

```
aView.frame = CGRectMake(20, 20, 50, 50);

//x轴向右移动200位移
[UIView beginAnimations:nil context:nil];
[UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];
[UIView setAnimationDuration:1];
[UIView setAnimationDelegate:self];

aView.frame = CGRectMake(220, 20, 50, 50);

[UIView commitAnimations];
```



## 缩放变换

```
CGAffineTransform transform = myView.transform;
//缩小尺寸
myView.transform = CGAffineTransformScale(transform, 0.01, 0.01);

[UIView beginAnimations:nil context:nil];
[UIView setAnimationDuration:0.6];

//还原为原来的尺寸
myView.transform = CGAffineTransformScale(transform, 1, 1);
// myView.transform = CGAffineTransformIdentity;

[UIView commitAnimations];
```





## 旋转变换

```
//旋转90度
[UIView beginAnimations:nil context:nil];
[UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];
[UIView setAnimationDuration:1];
[UIView setAnimationDelegate:self];

view.transform = CGAffineTransformMakeRotation(3.14/2.0);

[UIView commitAnimations];
```



## 设置代理

```
//旋转90度
[UIView beginAnimations:nil context:nil];
[UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];
[UIView setAnimationDuration:1];
[UIView setAnimationDelegate:self];

[UIView setAnimationDidStopSelector:@selector(animationDidStop)];

view.transform = CGAffineTransformMakeRotation(3.14/2.0);
[UIView commitAnimations];
```

```
- (void)animationDidStop {
    NSLog(@"动画结束");
}
```



## 视图切换的过渡动画

```
[UIView beginAnimations:nil context:nil];  
[UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];  
[UIView setAnimationDuration:0.6];  
//为parentView添加动画效果  
[UIView setAnimationTransition:UIViewAnimationTransitionFlipFromLeft  
        forView:parentView cache:YES];  
  
//在parentView执行动画的时候，调换两个视图的位置，以达到视图切换的效果  
[parentView exchangeSubviewAtIndex:index1 withSubviewAtIndex:index2];  
[UIView commitAnimations];
```

UIViewAnimationTransitionFlipFromLeft	从左向右翻转，用新视图代替旧视图
UIViewAnimationTransitionFlipFromRight	从右向左做翻转，隐藏旧视图，显示新视图
UIViewAnimationTransitionCurlUp	用从下向上翻页的方式显示新视图
UIViewAnimationTransitionCurlDown	新视图从上向下翻页盖住旧视图



## 1. 淡入动画

```
[UIView animateWithDuration:0.5 animations:^(void){ //动画开始执行调用的
block
    self.myView.alpha = 1.0;
} completion:^(BOOL finish) { //动画结束调用的block
    NSLog(@"动画结束");
}];
```

## 2. 视图切换

```
UIViewAnimationOptions options =
UIViewAnimationOptionTransitionCurlUp |
UIViewAnimationOptionCurveEaseInOut;
[UIView transitionWithView:parentView
    duration:3.0f options:options animations:^(
    [parentView exchangeSubviewAtIndex:page1Index
        withSubviewAtIndex:page2Index];
    }
    completion:^(BOOL finished) {
        NSLog(@"finished %d", finished);
    }
];
```



设置代理缺省为nil

```
+ (void)setAnimationDelegate:(id)delegate;
```

设置动画开始的调用方法，缺省NULL.

```
+ (void)setAnimationWillStartSelector:(SEL)selector;
```

设置动画停止的调用方法，缺省NULL

```
+ (void)setAnimationDidStopSelector:(SEL)selector;
```

设置动画持续事件，缺省是0.2秒

```
+ (void)setAnimationDuration:(NSTimeInterval)duration;
```

设置动画的方式，缺省是UIViewAnimationCurveEaseInOut

```
+ (void)setAnimationCurve:(UIViewAnimationCurve)curve;
```

设置动画的重复次数，缺省是0

```
+ (void)setAnimationRepeatCount:(float)repeatCount;
```



## CATRANSITION动画

- CATransition是对UIView进行更低层次的控制。
- CATransition 不同于UIView动画，它是作用于UIView的Layer层
- CATransition 主要用于两个视图切换过渡时的动画

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/738140113120006100>