

Animações

UIView Animations

UIView Animations

- Alto nível
- Mais comum
- Nem todas propriedades são animáveis

https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewPG_iPhoneOS/AnimatingViews/AnimatingViews.html

UIView.animateWithDuration

```
UIView.animateWithDuration(1.0, delay: 0.0, options: .CurveEaseOut, animations: {  
    self.favoriteButton.alpha = 0.0  
}) { finished in  
  
}
```

Transitions

```
@IBAction func likeButtonPressed(sender: UIButton) {  
  
    let favorited = !sender.selected  
    UIView.transitionWithView(sender, duration: 0.4, options: .TransitionCrossDissolve, animations: {  
        sender.selected = favorited  
    }, completion: nil)  
  
    // ...  
}
```

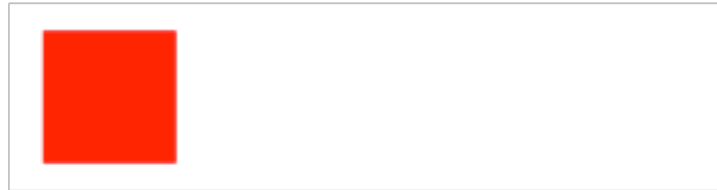
Core Animation

- Mais baixo nível
- Mais customizável
- **CABasicAnimation**

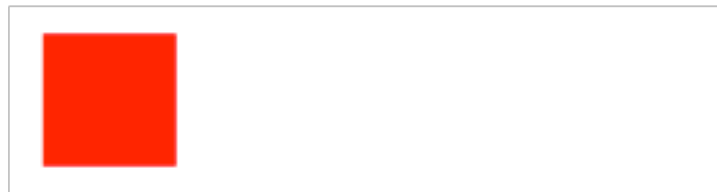
https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/CoreAnimation_guide/CoreAnimationBasics/CoreAnimationBasics.html

Curvas de animação

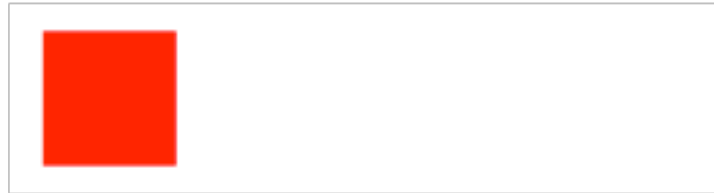
- Ease in (`kCAMediaTimingFunctionEaseIn`)



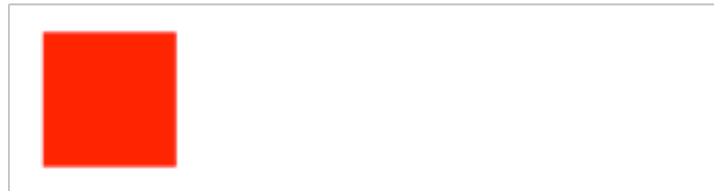
- Ease out (`kCAMediaTimingFunctionEaseOut`):



- Ease in ease out
(`kCAMediaTimingFunctionEaseInEaseOut`):



- Default (`kCAMediaTimingFunctionDefault`):



<http://www.objc.io/issues/12-animations/animations-explained/>


```
let pulseAnimation = CABasicAnimation(keyPath: "transform.scale")

pulseAnimation.duration = 0.4
pulseAnimation.fromValue = 1
pulseAnimation.toValue = favorited ? 1.2 : 0.8

pulseAnimation.autoreverses = true
pulseAnimation.repeatCount = 1

let function = kCAMediaTimingFunctionEaseInEaseOut
pulseAnimation.timingFunction = CAMediaTimingFunction(name: function)

sender.layer.addAnimation(pulseAnimation, forKey: nil)
```