Animation

강사 주영민

Fast campus

Animation

- UIView Animation
- UIImageView Animation
- UIViewController Animation

UIView Animation

- 특정 시간 동안 View의 속성값을 변화시키는 작업
  예)move, fade, Size Change,repeat 등

Animatable UIView properties

- frame
- bounds
- center
- transform : Modify this property to scale, rotate, or translate the view relative to its center point.
- alpha
- backgroundColor
- contentStretch

UIView Animation Method

```
@available(iOS 4.0, *)
open class func animate(withDuration duration: TimeInterval, delay: TimeInterval,
options: UIViewAnimationOptions = [], animations: @escaping () -> Swift.Void,
completion: ((Bool) -> Swift.Void)? = nil)
```

```
@available(iOS 4.0, *)
open class func animate(withDuration duration: TimeInterval, animations: @escaping
() -> Swift.Void, completion: ((Bool) -> Swift.Void)? = nil) // delay = 0.0,
options = 0
```

```
@available(iOS 4.0, *)
open class func animate(withDuration duration: TimeInterval, animations: @escaping
() -> Swift.Void) // delay = 0.0, options = 0, completion = NULL
```

Animation 속성

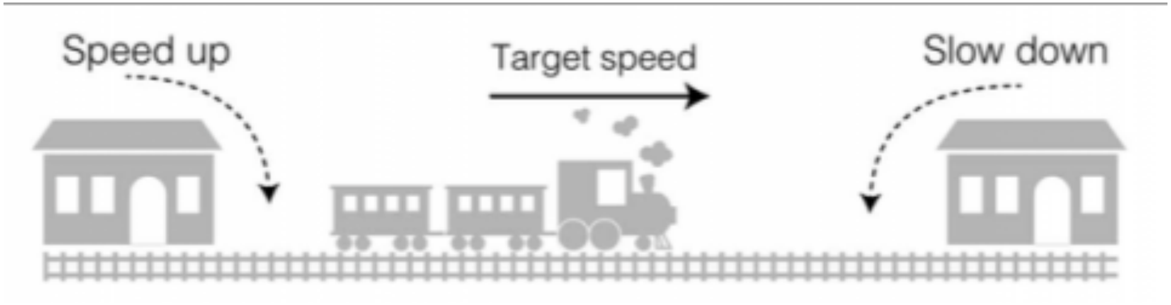
- Duration : Animation 진행 시간
- Delay : 대기 시간
- Options : Animation 옵션
- Animations : 애니메이션 동작 Block 함수
- Completions : 애니메이션 완료 후 동작 Block함수

Options

Loop

```
public static var `repeat`: UIViewAnimationOptions { get }
// repeat animation indefinitely

public static var autoreverse: UIViewAnimationOptions { get }
// if repeat, run animation back and    Bounce

public static var curveEaseInOut: UIViewAnimationOptions { get }
// default

public static var curveEaseIn: UIViewAnimationOptions { get }

public static var curveEaseOut: UIViewAnimationOptions { get }

public static var curveLinear: UIViewAnimationOptions { get }
```

Options - 속도



- Linear: This option applies no acceleration or deceleration to the animation.
- CurveEaseIn: This option applies acceleration to the start of your animation.
- CurveEaseOut: This option applies deceleration to the end of your animation.
- CurveEaseInOut: This option applies acceleration to the start of your animation and applies deceleration to the end of your animation.

Animation 예제

```
UIView.animate(withDuration: 0.5,
               delay: 0,
               options: [.curveEaseIn,.repeat],
               animations: {

                //에니메이션 내용
}) { (completion) in
    //완료후 동장
}
```

Animation 실습

- Auto Layout 애니메이션 적용하기

추가 UIView Animation Method

```
//애니메이션의 Bounce를 줄때 사용
@available(iOS 7.0, *)
open class func animate(withDuration duration: TimeInterval,
                                        delay: TimeInterval,
          usingSpringWithDamping dampingRatio: CGFloat,
               initialSpringVelocity velocity: CGFloat,
                         options: UIViewAnimationOptions = [],
                      animations: @escaping () -> Swift.Void,
                      completion:((Bool) -> Swift.Void)? = nil)
```

Spring Animation Method 속성

- **dampingRatio** : The damping ratio for the spring animation as it approaches its quiescent state.
  To smoothly decelerate the animation without oscillation, use a value of 1. Employ a damping ratio closer to zero to increase oscillation.

- **velocity** : The initial spring velocity. For smooth start to the animation, match this value to the view's velocity as it was prior to attachment.
  A value of 1 corresponds to the total animation distance traversed in one second. For example, if the total animation distance is 200 points and you want the start of the animation to match a view velocity of 100 pt/s, use a value of 0.5.

Animation 실습

- 통통튀기는 버튼 만들기