

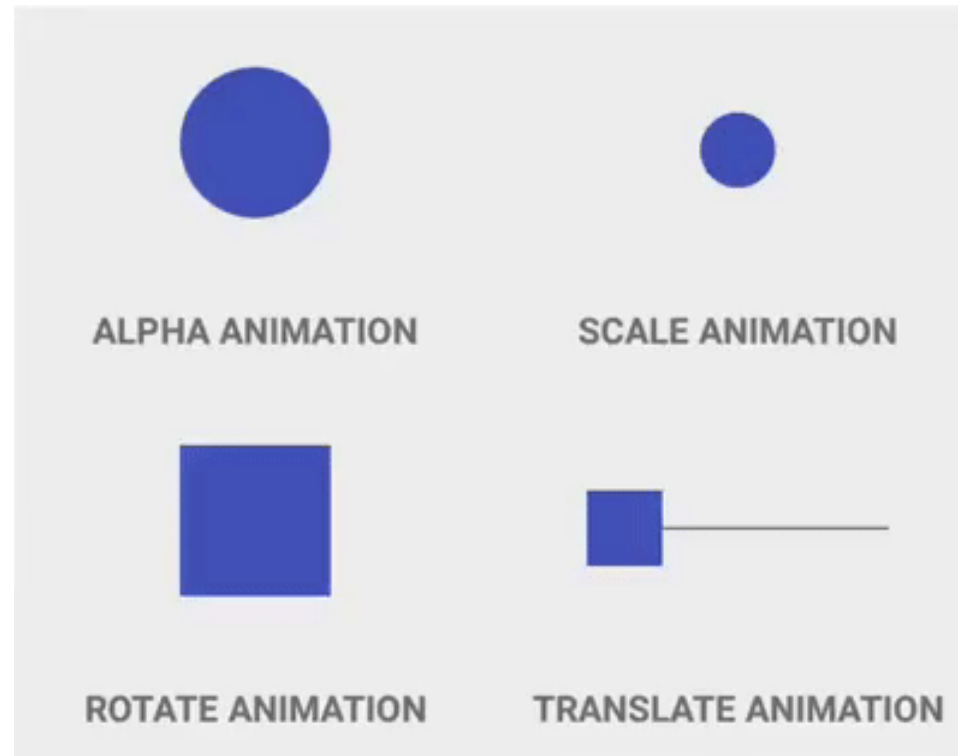
IOS Animasyon İşlemleri

Kasım ADALAN

Elektronik ve Haberleşme Mühendisi

Senior Android and IOS Developer

Animasyon Çeşitleri

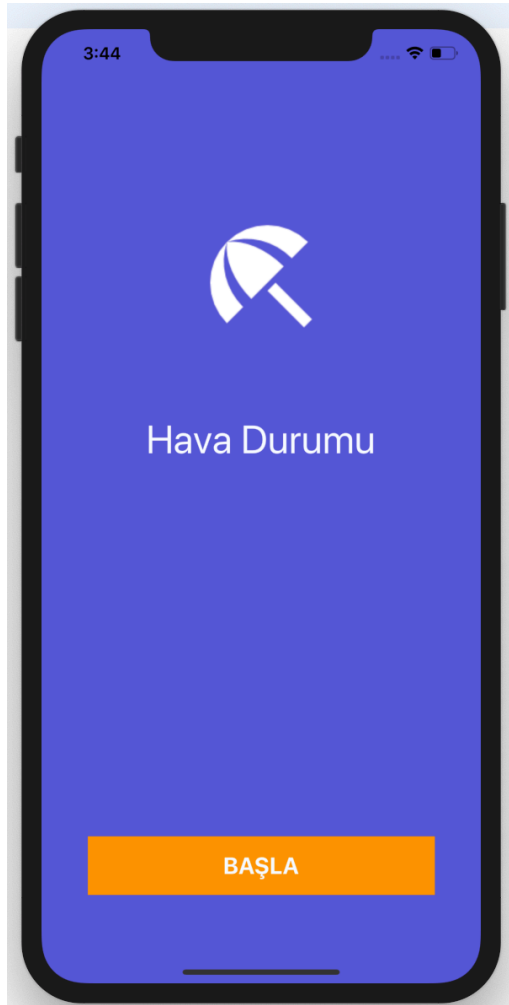


Animasyon kullanımı

```
//withDuration : Animasyonun sn miktarı  
//delay        : Animasyona başlamadan önce verilen geçikme  
//options      : Bu alanda özellikler belirtilir.  
//repeat       : Animasyonun tekrarlı olmasını sağlar  
//autoReverse  : Animasyonu tersten yaptırır.
```

```
UIView.animate(withDuration: 5  
    ,delay: 1  
    ,options: [.repeat,.autoreverse]  
    ,animations: {  
  
        //Animasyon olmasını istediğimiz kod buraya yazılır.  
  
    }, completion: nil)
```

Animasyonları Test Etmek için
Uygulama Tasarımı Hazırla

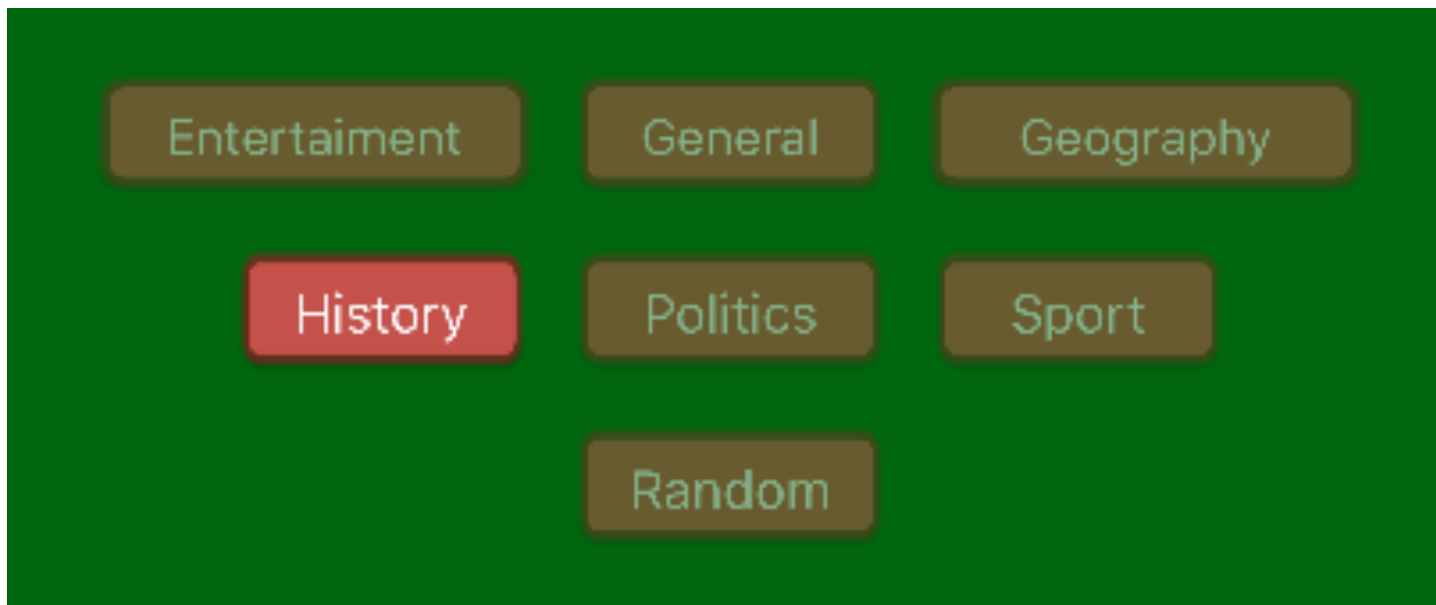


```
9  import UIKit
10
11  class ViewController: UIViewController {
12
13      @IBOutlet weak var button: UIButton!
14      @IBOutlet weak var yazi: UILabel!
15      @IBOutlet weak var resim: UIImageView!
16
17      override func viewDidLoad() {
18          super.viewDidLoad()
19      }
20
21      @IBAction func buttonBasla(_ sender: Any) {
22
23      }
24
25  }
```

Alpha Animasyonu

Alpha

- Görsel nesnelerin transparan hale dönüşümünü sağlar , 0 ile 1 arasında bir değişim olur. 0 : tamamen görünmez , 1 : tam görünür hale gelir.



Alpha

```
//Görsel nesnenin saydamlığını değiştirir.  
//0 : görünmez  
//1 : görünürdür
```

```
self.resim.alpha = 0
```

```
UIView.animate(withDuration: 5, animations: {
```

```
    self.resim.alpha = 1
```

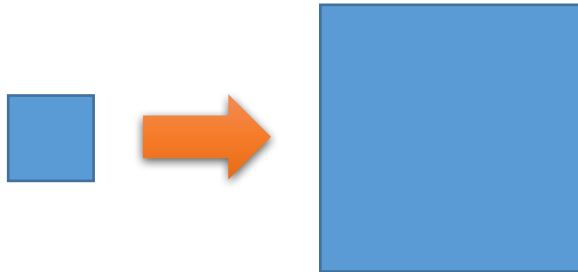
```
}, completion: nil)
```



Scale Animasyonu

Scale

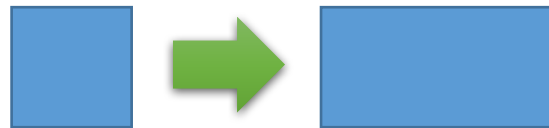
- Görsel nesnelerin boyutunu değiştirme işlemleri için kullanılır.



Örnek

```
UIView.animate(withDuration: 1, animations: {  
    self.resim.transform = CGAffineTransform(scaleX: 2, y: 1)  
}, completion: nil)
```

X yönünde normal haline göre 2 kat büyüdü



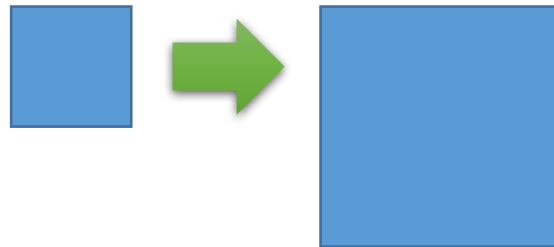
Büyüme merkezi



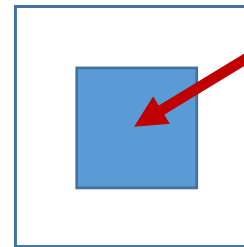
Örnek

```
UIView.animate(withDuration: 1, animations: {  
    self.resim.transform = CGAffineTransform(scaleX: 2, y: 2)  
}, completion: nil)
```

X ve Y yönünde normal haline göre 2 kat büyüdü



Büyüme merkezi



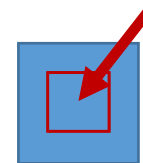
Örnek

```
UIView.animate(withDuration: 1, animations: {  
    self.resim.transform = CGAffineTransform(scaleX: 0.5, y: 0.5)  
}, completion: nil)
```

X ve Y yönünde normal haline göre 2 kat küçüldü



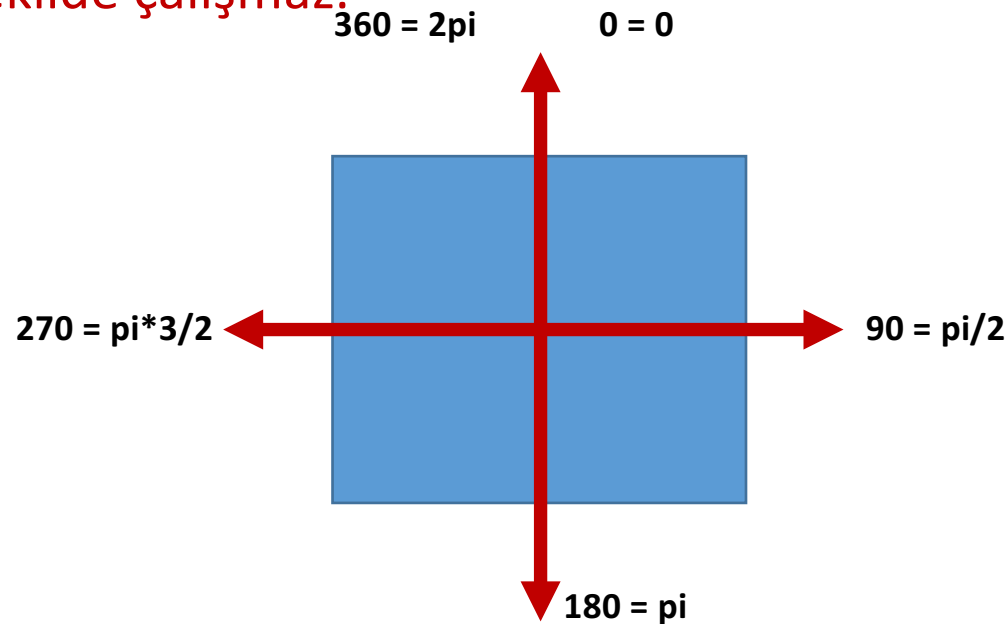
Küçülme merkezi



Rotate Animasyonu

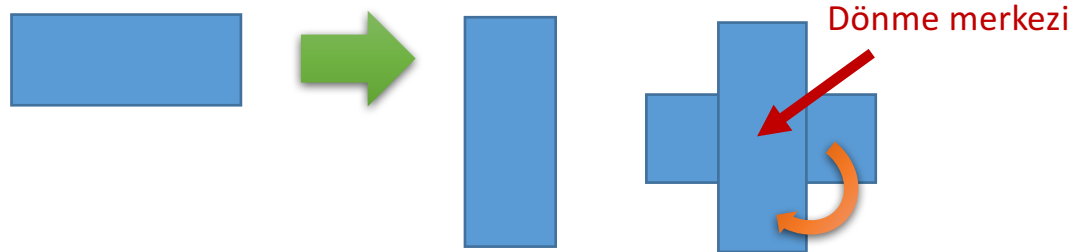
Rotate

- Görsel nesnenin döndürülme işlemidir.
- Radyan cinsinden dönme işlemi sağlanır.
- Belirtilen açıya gider. Belirtilen açı kadar döner gibi görünür ama o şekilde çalışmaz.



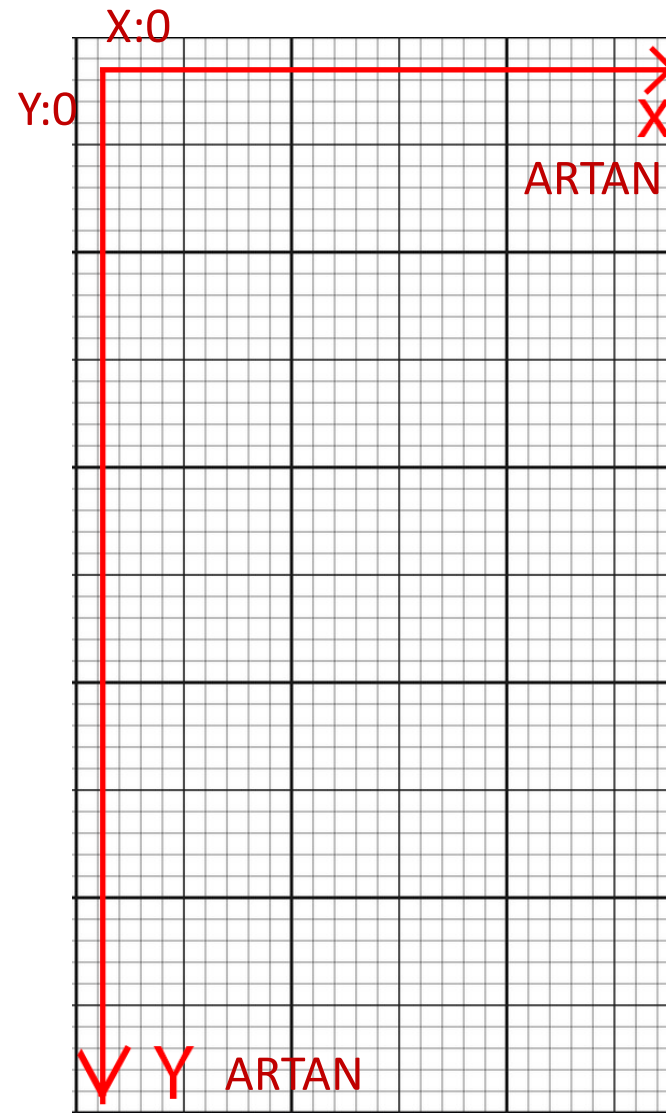
Örnek

```
//radyan dönüşümü derece * .pi/180  
  
UIView.animate(withDuration: 1, animations: {  
    self.yazi.transform = CGAffineTransform(rotationAngle: 90 * .pi/180 )  
}, completion: nil)
```



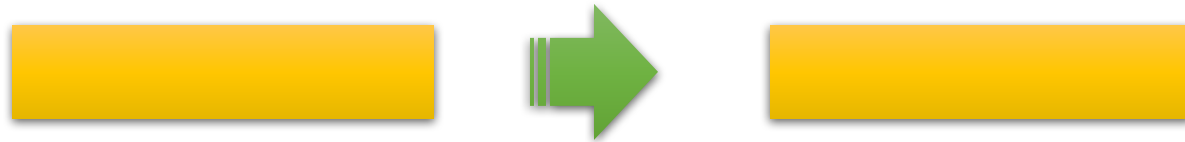
Translation Animasyonu

Ekran Koordinatı



Translation

- Görsel nesnelere hareket vermek için veya yer değişimi vermek için kullanılır.
- İlerleme miktarlarıdır.



Örnek

```
UIView.animate(withDuration: 1, animations: {  
    self.yazi.transform = CGAffineTransform(translationX: 50, y: 0)  
}, completion: nil)
```

x yönünde ileri hareket



Örnek

```
UIView.animate(withDuration: 1, animations: {  
    self.yazi.transform = CGAffineTransform(translationX: -50, y: 0)  
}, completion: nil)
```

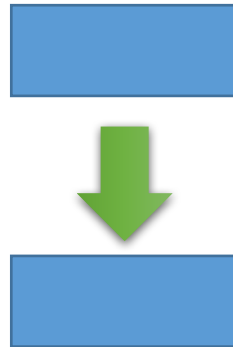
x yönünde geri hareket



Örnek

```
UIView.animate(withDuration: 1, animations: {  
    self.yazi.transform = CGAffineTransform(translationX: 0, y: 50)  
}, completion: nil)
```

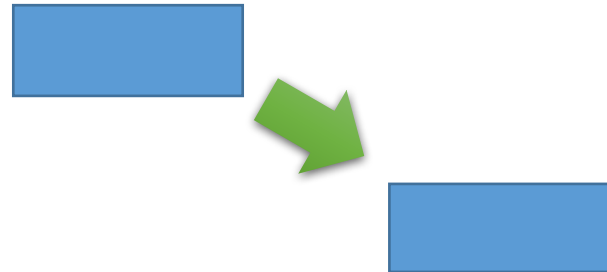
y yönünde ileri hareket



Örnek

```
UIView.animate(withDuration: 1, animations: {  
    self.yazi.transform = CGAffineTransform(translationX: 50, y: 50)  
}, completion: nil)
```

y yönünde ileri hareket



Diğer Özellikler

Gecikme Oluşturma

//Görsel nesneye delay ile animasyona başlamadan önce gecikme eklenebilir.

```
UIView.animate(withDuration: 2
    ,delay: 1
    ,animations: {

        self.yazi.transform = CGAffineTransform(translationX: 0, y: 100)

    }, completion: nil)
```

İvme Oluşturma

- Görsel nesneye gerçek hayatta olduğu gibi hızlanma ve yavaşlama etkisi verilebilir. **options** parametresi ile bu özellik verilir.
- **.curveLinear**: varsayılan hali ile çalışır ivme etkisi olmaz.
- **.curveEaseIn**: Animasyonun başında ivme uygulanır.
- **.curveEaseOut**: Animasyonun sonunda ivme uygulanır.
- **.curveEaseInOut**: Animasyonun başında ve sonunda ivme uygulanır.

//Görsel nesneye gerçek hayatta olduğu gibi hızlanma ve yavaşlama etkisi verilebilir.
//Not : Geçikme parametresi olmadan options özelliği eklenemez.

```
UIView.animate(withDuration: 2
    ,delay: 0
    ,options: [.curveEaseIn]
    ,animations: {

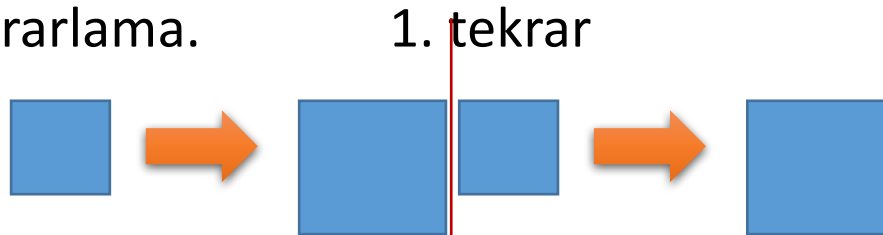
        self.yazi.transform = CGAffineTransform(translationX: 0, y: 100)

    }, completion: nil)
```

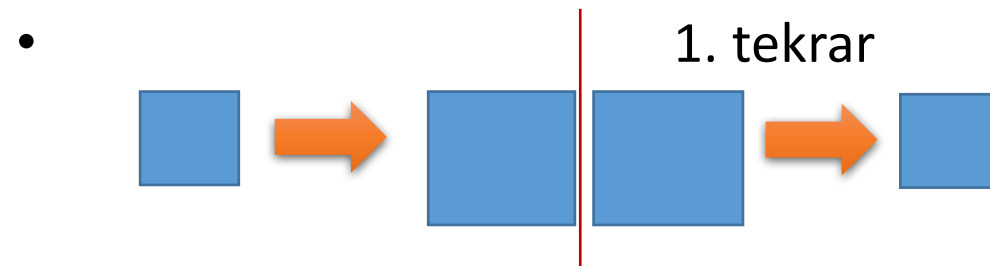
Tekrar Oluşturma

- Animasyonların tekrarlama şeklini belirler. **options** parametresi ile bu özellik verilir.

- repeat**: yapılan işlemin son haline geldikten sonra başa dönerek işlemi tekrarlama.



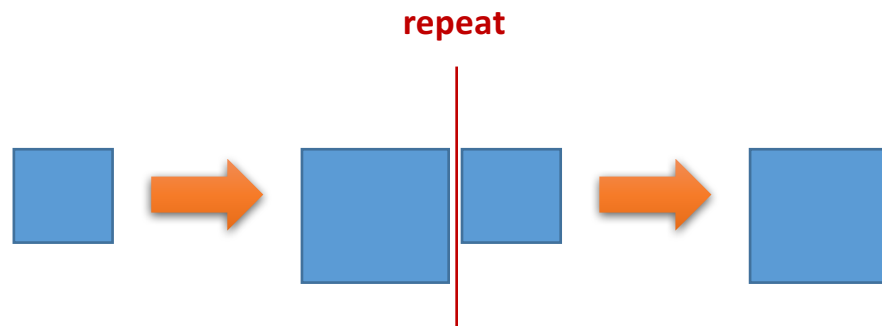
- autoreverse** : yapılan işlemin son haline geldikten sonra geriye doğru işlem yapma şeklinde tekrarlama



Örnek

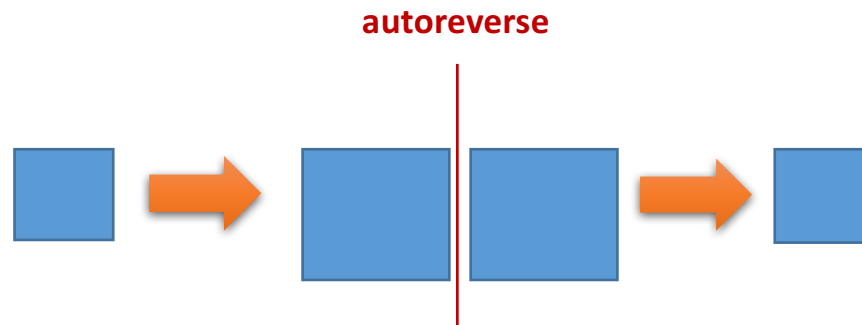
```
//Görsel nesnelere tekrar etme özelliği verilebilir.  
//options alanında bu özellikler belirtilir.  
//repeat : Tekrarlı olmasını sağlar , autoReverse : animasyonu tersten yaptırır.
```

```
UIView.animate(withDuration: 1  
    ,delay: 0  
    ,options: [.repeat]  
    ,animations: {  
  
        self.resim.transform = CGAffineTransform(translationX: 0, y: 50)  
  
    }, completion: nil)
```



Örnek

```
//Görsel nesnelere tekrar etme özelliği verilebilir.  
//options alanında bu özellikler belirtilir.  
//repeat : Tekrarlı olmasını sağlar , autoReverse : animasyonu tersten yaptırır.  
  
UIView.animate(withDuration: 1  
    ,delay: 0  
    ,options: [.repeat,.autoreverse]  
    ,animations: {  
  
        self.resim.transform = CGAffineTransform(translationX: 0, y: 50)  
  
    }, completion: nil)
```



Çoklu Animasyon

Çoklu Animasyon

- Aynı anda birden fazla animasyonu görsel nesneye yaptırılabilir.
- Transform animasyonları concatenating metodu ile birleştirilir.

```
//Aynı anda birden animasyon yaptırılabilir.  
//Transform animasyonları concatenating metodu ile birleştirilir.  
  
UIView.animate(withDuration: 1, animations: {  
  
    let hareket = CGAffineTransform(translationX: 0, y: 100)  
    let kucul = CGAffineTransform(scaleX: 0.5, y: 0.5)  
    let don = CGAffineTransform(rotationAngle: 90 * .pi/180 )  
  
    let x = hareket.concatenating(kucul)  
  
    self.yazi.transform = don.concatenating(x)  
  
    //Alpha değeri concatenating gerektirmez  
    self.yazi.alpha = 0  
  
}, completion: nil)
```

Ardışık Animasyon

Ardışık Animasyon

- Bir animasyon bittikten sonra diğer animasyonu başlatabiliriz.
- Animasyonun bittiğini completion özelliği ile dinleyebiliriz.

```
UIView.animate(withDuration: 1, animations: {  
  
    self.resim.alpha = 0  
  
}, completion: { (true) in  
  
    print("Animasyon bitti")  
  
})
```

Örnek

- Sırayla bütün görsel nesnelerin ekrandan kaybolması ve aynı anda tekrar görünür olması

```
UIView.animate(withDuration: 1, animations: {

    self.resim.alpha = 0

}, completion: { (true) in

    UIView.animate(withDuration: 1, animations: {

        self.yazi.alpha = 0

    }, completion: { (true) in

        UIView.animate(withDuration: 1, animations: {

            self.button.alpha = 0

        }, completion: { (true) in

            UIView.animate(withDuration: 1, animations: {

                self.resim.alpha = 1
                self.yazi.alpha = 1
                self.button.alpha = 1

            }, completion: nil)

        })

    })

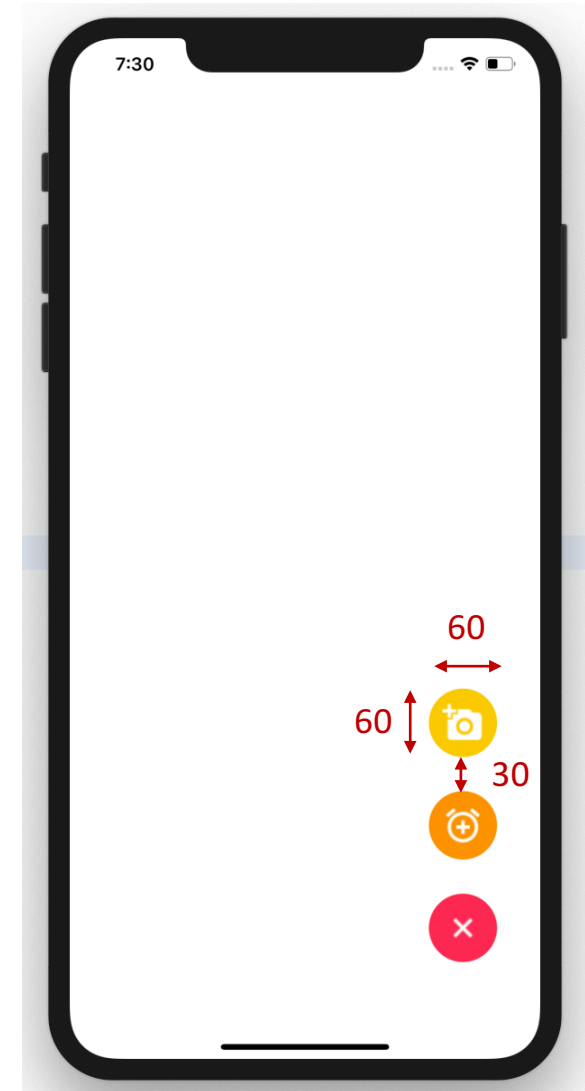
})
```

Animasyonlu FAB Button Uygulaması

Tasarım

```
class ViewController: UIViewController {  
  
    @IBOutlet weak var fabMain: UIButton!  
    @IBOutlet weak var fabBirinci: UIButton!  
    @IBOutlet weak var fabIkinci: UIButton!  
  
    var fabDurum = false  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Buttonları oval yapma  
        fabMain.layer.cornerRadius = fabMain.frame.size.width/2  
        fabBirinci.layer.cornerRadius = fabBirinci.frame.size.width/2  
        fabIkinci.layer.cornerRadius = fabIkinci.frame.size.width/2  
    }  
  
    @IBAction func fabMainTikla(_ sender: Any) {  
        print("Fab main tıklandı")  
    }  
  
    @IBAction func fabBirinciTikla(_ sender: Any) {  
        print("Fab birinci tıklandı")  
    }  
  
    @IBAction func fabIkinciTikla(_ sender: Any) {  
        print("Fab ikinci tıklandı")  
    }  
}
```

Buttonlar Oval Yapılır.



Tam Kod

```
import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var fabMain: UIButton!
    @IBOutlet weak var fabBirinci: UIButton!
    @IBOutlet weak var fabIkinci: UIButton!

    var fabDurum = false

    override func viewDidLoad() {
        super.viewDidLoad()

        //Buttonları oval yapma

        fabMain.layer.cornerRadius = fabMain.frame.size.width/2
        fabBirinci.layer.cornerRadius = fabBirinci.frame.size.width/2
        fabIkinci.layer.cornerRadius = fabIkinci.frame.size.width/2

        //Birinci ve ikinci fab açıldığında görünmez olsunlar
        //ve boyut olarak küçük olsunlar açılırken animasyonlu büyüsünler.
        //Aksi halde direk büyük halleri görünür.
        fabBirinci.alpha = 0
        fabIkinci.alpha = 0
        self.fabBirinci.transform = CGAffineTransform(scaleX: 0.1, y: 0.1)
        self.fabIkinci.transform = CGAffineTransform(scaleX: 0.1, y: 0.1)
    }
}
```

```
@IBAction func fabMainTikla(_ sender: Any) {
    print("Fab main tıklandı")

    UIView.animate(withDuration: 0.5, animations: {

        if self.fabDurum {

            self.fabBirinci.alpha = 0
            self.fabIkinci.alpha = 0

            self.fabBirinci.transform = CGAffineTransform(scaleX: 0.1, y: 0.1)
            self.fabIkinci.transform = CGAffineTransform(scaleX: 0.1, y: 0.1)
            //Açı olarak eski haline gelir.
            self.fabMain.transform = CGAffineTransform(rotationAngle: 0 * .pi/180 )

            self.fabDurum = false

        }else{

            self.fabBirinci.alpha = 1
            self.fabIkinci.alpha = 1

            self.fabBirinci.transform = CGAffineTransform(scaleX: 1.0, y: 1.0)
            self.fabIkinci.transform = CGAffineTransform(scaleX: 1.0, y: 1.0)
            //45. dereceye gider.
            self.fabMain.transform = CGAffineTransform(rotationAngle: 45 * .pi/180 )

            self.fabDurum = true

        }

    }, completion: nil)
}
```

Teşekkürler...



kasım-adalan



kasimadalan@gmail.com



kasimadalan