

swiftkodlamadersleri.com



*Xcode nesneleri*

*Buton Özellikleri ve Kullanımı*



# İçindekiler

- ❖ Bölüm-1 : Buton'un tanıtılması,
- ❖ Bölüm-2 : Buton ekleme ve utilities ile düzenleme,
- ❖ Bölüm-3 : frame ve layer kullanımı,
- ❖ Bölüm-4 : titleLabel kullanımı,
- ❖ Bölüm-5 : set formlarının kullanımı ve State kavramı,
- ❖ Bölüm-6 : isHidden, isEnabled, isHighlighted ve Attributed kullanımı,
- ❖ Bölüm-7 : Buton olayları (events) kullanımı,



Carrier



12:44 PM



Button

Button

Button

Butonum

Buton 5

Button

Label



# Bölüm 1

## Buton'un tanıtılması,

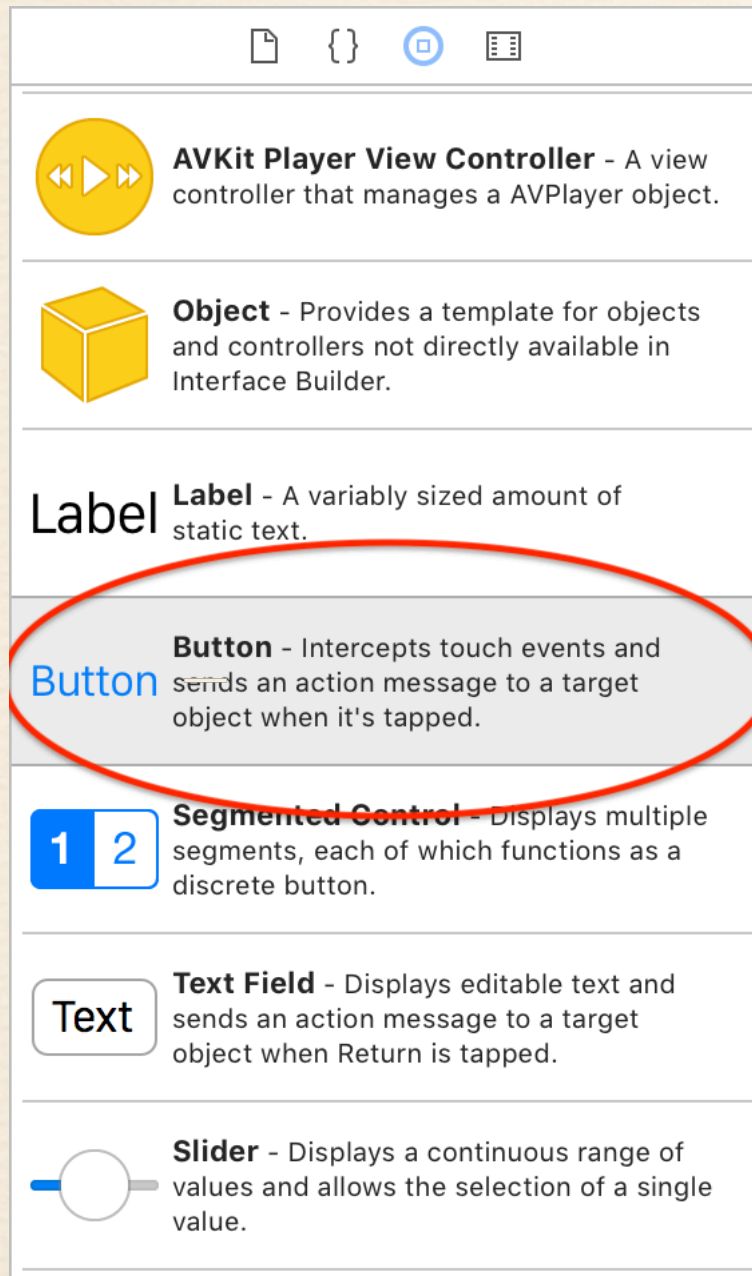


# Buton Nedir?

- ❖ Kullanıcı etkileşimlerine yanıt olarak özel kodunuzu yürüten bir kontrol.
- ❖ Hem Outlet hemde Action özelliği vardır.
- ❖ Outlet ile; özelliklerine ulaşılır. (Renk, Font vb.)
- ❖ Action ile; Butona dokunduğunuzda ona bağlı eylemleri gerçekleştirir.
- ❖ Butonun üzerine bir metin etiketi (başlık), bir resim veya her ikisi birden yerleştirilerek Butonun amacı kullanıcıya bildirilir.
- ❖ Arayüzünüze programsal olarak veya Arayüz Oluşturucu kullanarak düğmeler ekleyebilir ve yapılandırabilirsiniz.



# Nesne kütüphanesi ve Buton çeşitleri



Button  

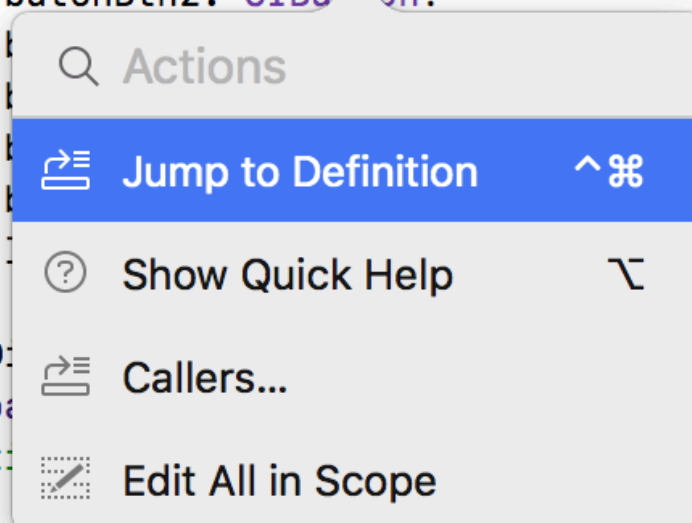


# En önemli başvuru kaynağımız

❖ <https://developer.apple.com/documentation/uikit/uibutton>

ve

```
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var butonBtn1: UIButton!
14     @IBOutlet weak var butonBtn2: UIButton!
15     @IBOutlet weak var
16     @IBOutlet weak var
17     @IBOutlet weak var
18     @IBOutlet weak var
19     @IBOutlet weak var
20
21     override func viewDidLoad() {
22         super.viewDidLoad()
23         // Do any additional setup after loading the view, typically from a nib.
24
25         butonBtn2.frame = CGRect(x: 16, y: 150, width: 343, height: 80)
```





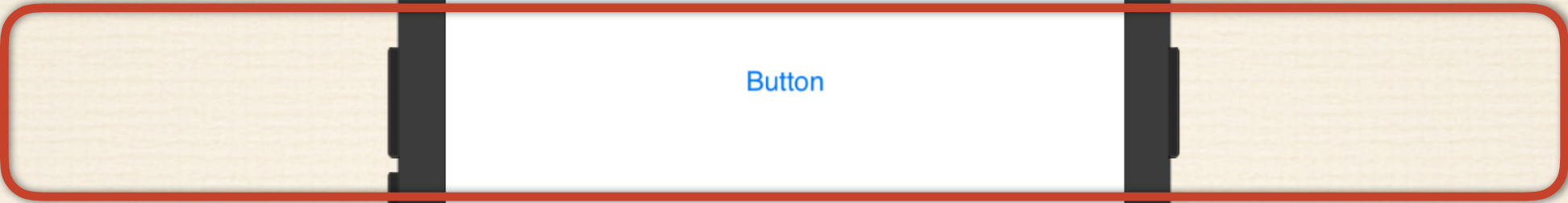
# Bölüm 2

Buton ekleme ve utilities ile düzenleme,

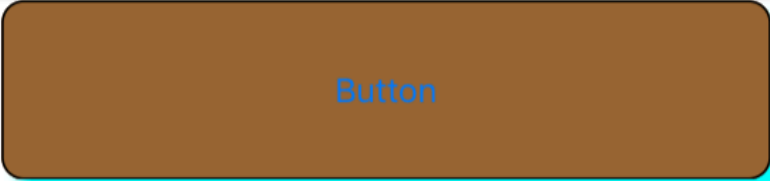


- ❖ Buton ekleyecek ve Connection edeceğiz,
- ❖ Buton'un özelliklerine Utilities bölümünden ulaşacağız ve düzenleyeceğiz,
- ❖ Yapacağımız değişiklikler sabit atama sınıfına girecek,
- ❖ Unutma! Kod satırında yapılacak olan atama Utilities bölümünden önce gelir,
- ❖ Düzenlemeler butonBtn1 üzerinde izlenecektir.

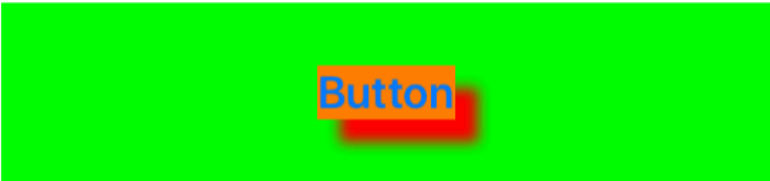




Button



Button

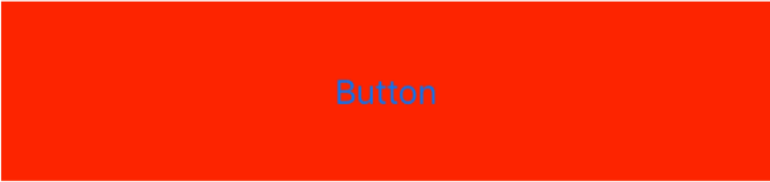


Button



Butonum

Buton 5



Button

Label





# Bölüm 3

frame ve layer kullanımı,

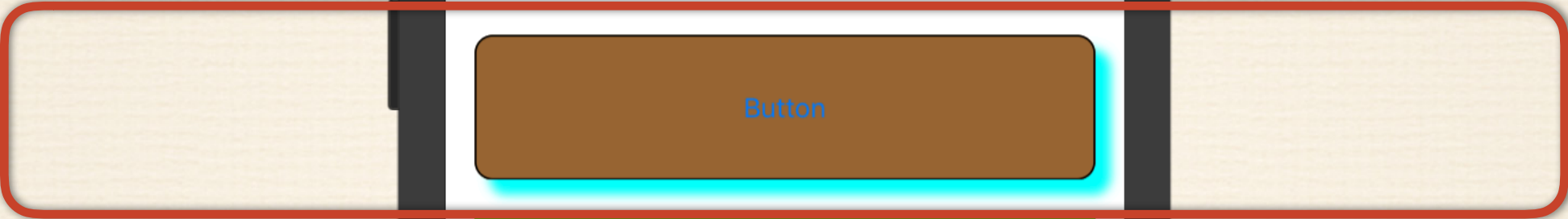


❖ Buton'un;

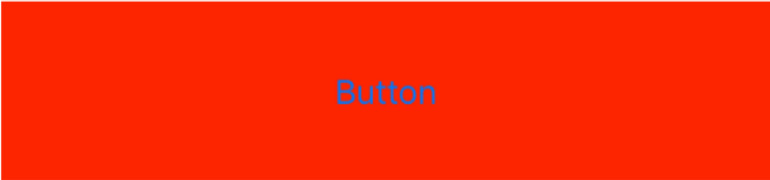
- yerleşiminin ve boyutlarının,
- köşeleri,
- arka plan renginin,
- Gölgesinin düzenlenmesi,

❖ Düzenlemeler butonBtn2 üzerinde izlenecektir.





Buton 5



Label





# Bölüm 4

titleLabel kullanımı,



Carrier



12:44 PM



Button

Button

Button

Butonum

Buton 5

Button

Label



- ❖ Buton başlığının;
  - arka plan renginin
  - font'unun,
  - satır sayısının,
  - konumunun,
  - gölge eklenmesinin,
  - Butonun ve buton başlığının istenen açıda düzenlenmesi,
  
- ❖ Düzenlemeler butonBtn3 üzerinde izlenecektir.



# titleLabel

- ❖ `var titleLabel: UILabel?`
- ❖ Butonun Title değerini çağırıp değiştirebilen bir işlevi vardır. (font, lineBreakMode, numberOfLines, textAlignment, shadowOffset, shadowColor, transform vb.)
- ❖ `buton.titleLabel?.font = UIFont(name: "Farah", size: 20)`





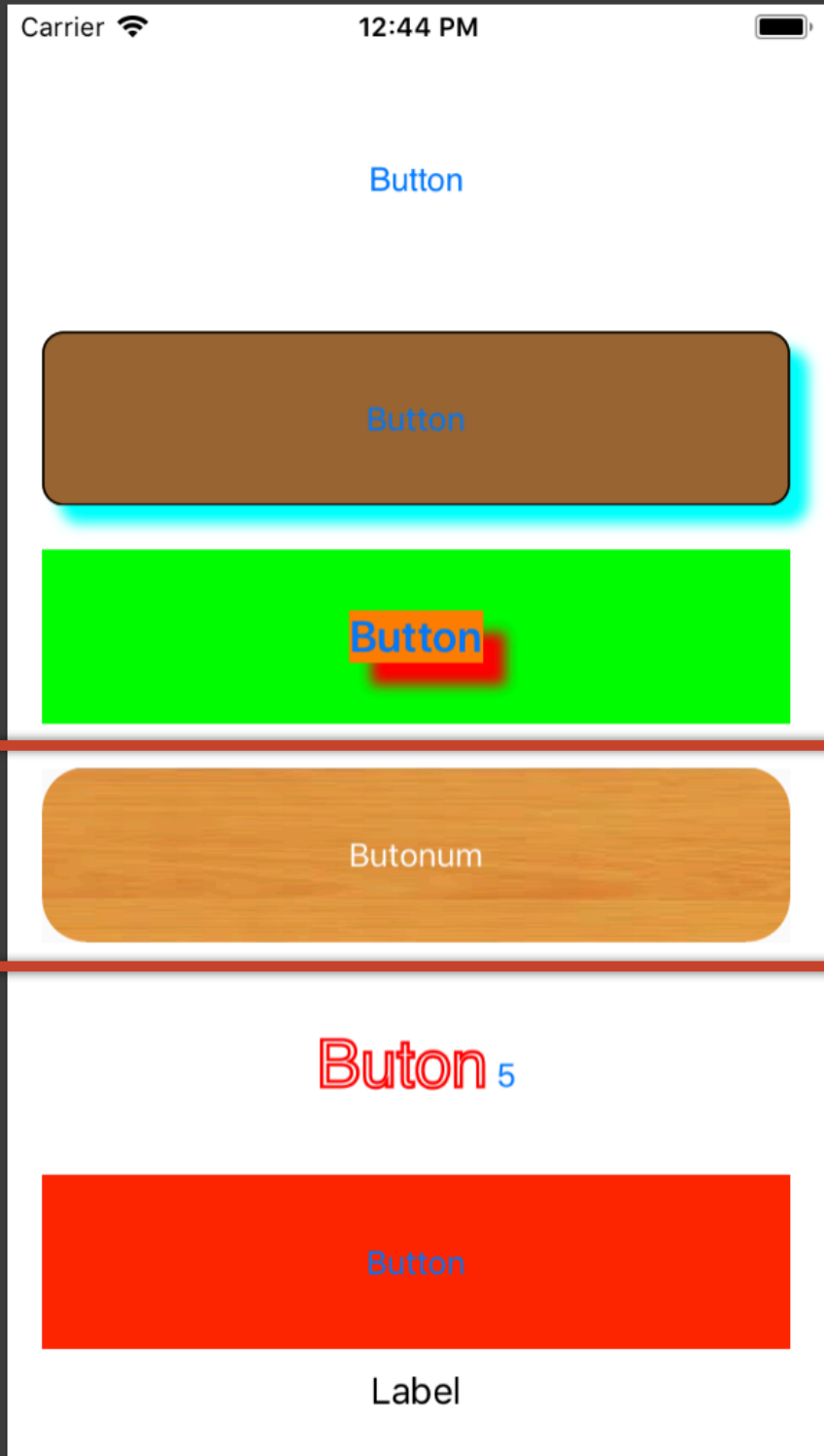
# Bölüm 5

set formlarının kullanımı ve state kavramı,



- ❖ Buton'a State durumuna göre ;
  - Başlık eklenmesi,
  - Arka plan resmi eklenmesi,
  - Başlık rengini düzenleme,
- ❖ Köşelerini düzenleme,
- ❖ Düzenlemeler butonBtn4 üzerinde izlenecektir.







# setTitle

- ❖ setTitle(String?, for: UIControl.State)
- ❖ Belirtilen State için kullanılacak Title'ı ayarlar.
- ❖ `buton.setTitle("Butonum", for: .normal)`
- ❖ `buton.setTitle("Butonuma basıldı", for: .highlighted)`



# State kavramı

- ❖ Arayüzünüze bir düğme eklediğinizde, başlangıçta varsayılan State'tedir (normal), bu, düğmenin etkin olduğu ve kullanıcının bununla etkileşmediği anlamına gelir. Kullanıcı butonla etkileşime girdikçe, durumu diğer değerlere geçer.
- ❖ Örneğin, kullanıcı bir butona dokunduğunda, buton vurgulanan (highlighted) duruma geçer.



# State kavramı

- ❖ **normal** : Bir denetimin normal veya varsayılan durumu,
- ❖ **highlighted** : Bir kontrolün vurgulanması durumu,
- ❖ **disabled** : Bir kontrolün devre dışı durumu,
- ❖ **selected** : Bir kontrolün seçilmiş hali,



# setTitleColor

- ❖ setTitleColor(UIColor?, for: UIControl.State)
- ❖ Belirtilen State için kullanılacak başlığın rengini ayarlar.
- ❖ `buton.setTitleColor(UIColor.green, for: .normal)`



# setBackgroundImage

- ❖ setBackgroundImage(UIImage?, for: UIControl.State)
- ❖ Belirtilen State için kullanılacak arka plan resmini ayarlar.
- ❖ `buton.setBackgroundImage(UIImage(named: "a.jpg"), for: .normal)`



# setImage

- ❖ setImage(UIImage?, for: UIControl.State)
- ❖ Belirtilen State için butonun üzerine eklenecek resmi ayarlar.
- ❖ `buton.setImage(UIImage(named: "home.jpg"), for: .normal)`





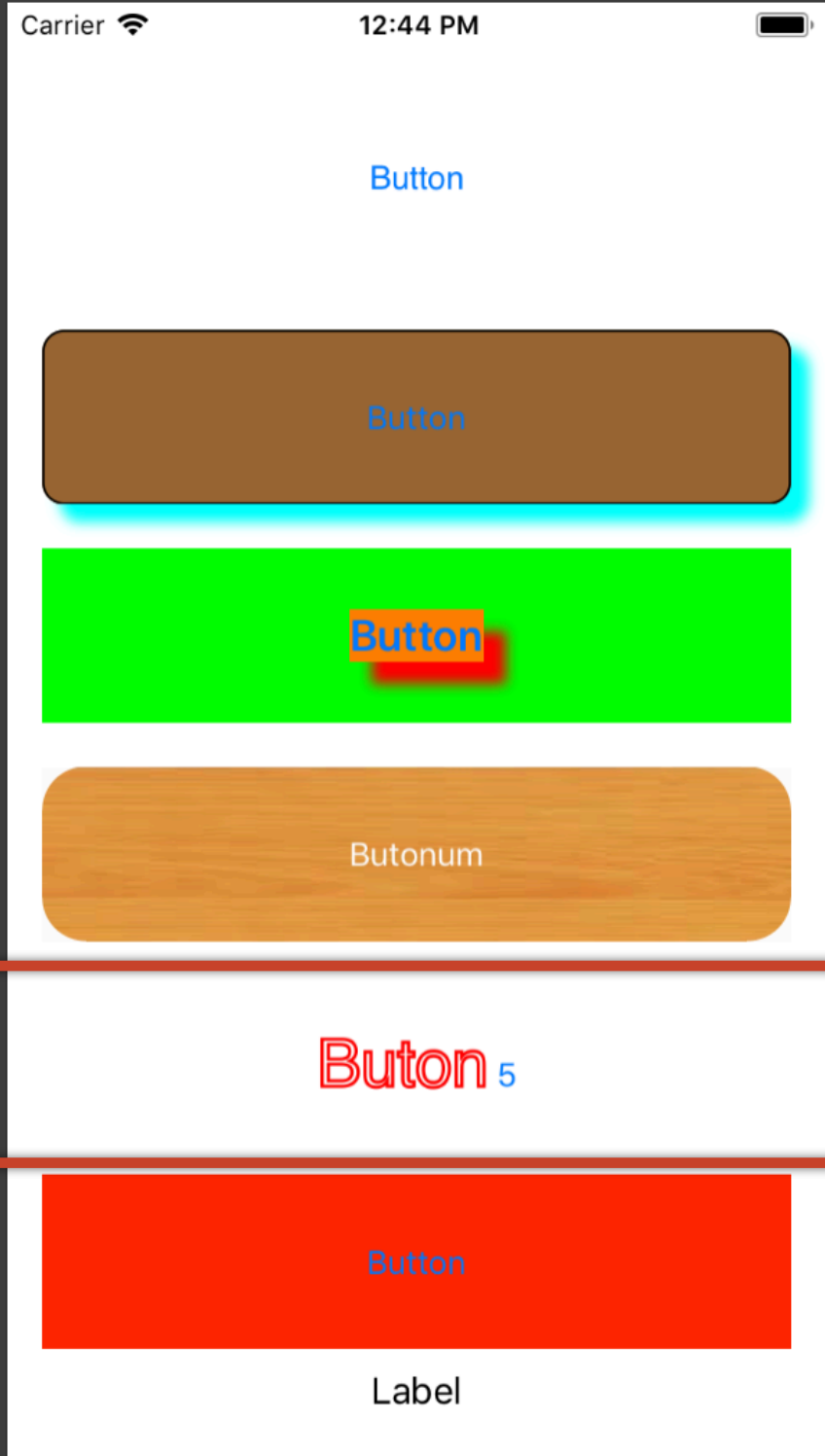
# Bölüm 6

isHidden, isEnabled, isHighlighted ve Attributed  
kullanımı,



- ❖ Buton'un;
  - isHidden
  - isEnabled,
  - isHighlighted özellikleri,
- ❖ Buton başlığının Attributed özelliğini düzenleme,
- ❖ Düzenlemeler butonBtn5 üzerinde izlenecektir.







# Gizleme, Devre dışı bırakma ve Vurgulama

❖ Butonu gizlemek için;

`butonBtn5.isHidden = true`

❖ Butonu devre dışı (kullanım dışı) bırakmak için;

`butonBtn5.isEnabled = false`

❖ Butonu vurgulamak için;

`butonBtn5.isHighlighted = true`



# setAttributedTitle

- ❖ `setAttributedTitle(NSAttributedString?, for: UIControl.State)`
- ❖ Belirtilen State için kullanılacak stil başlıklarını ayarlar.

```
let attributedTextDeneme = NSMutableAttributedString(string: "Buton")
```

```
attributedTextDeneme.addAttribute(NSAttributedStringKey.underlineStyle, value: 1, range: NSRange(location: 0, length: 5))
```

```
buton.setAttributedTitle(attributedTextDeneme, for: .normal)
```





# Bölüm 7

Buton olayları (events) kullanımı,



❖ Buton'un event'lerini tanıma;

- touchCancel,
- touchDown,
- touchDownRepeat,
- touchDragEnter,
- touchDragExit,
- touchDragInside,
- touchDragOutside,
- touchUpInside,
- touchUpOutside,

❖ Düzenlemeler butonBtn6 üzerinde izlenecektir.



Carrier

12:44 PM



Button

Button

Button

Butonum

Buton 5

Button

Label



# Events

## ❖ touchCancel

Kontrol için mevcut dokunuşları iptal eden bir sistem olayı.

## ❖ touchDown

Butona bastığınız anda tetiklenir. Butonu bırakmanıza gerek yok.

## ❖ touchDownRepeat

Butona kısa aralıklarla tekrarlanan bir dokunma olayıdır. Çift tıklamaya denk gelir.



# Events

## ❖ touchDragEnter

Butona basıldığında tetiklenmez. Butona bastıktan sonra parmağınızı butonun sınırları dışına çıkarıp tekrar buton üzerine döndüğünüzde tetiklenir.

## ❖ touchDragExit

Butona basıldığında tetiklenmez. Butona bastıktan sonra parmağınızı butonun sınırları dışına çıkardığınızda tetiklenir.

## ❖ touchDragInside

Butona basıldığında tetiklenmez. Butona bastıktan sonra parmağınızı buton üzerinde gezdirdiğinizde tetiklenir.



# Events

- ❖ **touchDragOutside**

Butona basıldığında tetiklenmez. Butona bastıktan sonra parmağınızı buton sınırlarının dışına çıkardığınızda tetiklenir.

- ❖ **touchUpInside**

Butona basıldığında tetiklenmez. Butona basıldıktan sonra parmağın kaldırılması ile tetiklenir.

- ❖ **touchUpOutside**

Butona basıldığında tetiklenmez. Butona basıldıktan sonra parmağınızı butonun sınırları dışına çıkardıktan sonra kaldırılması ile tetiklenir.

