

# Proje: Işığ ı Ayarlama

Bu zamana kadar Swift ile alakalı birçok şey öğrendiniz. Şimdi ise bu öğrendiklerinizle bir proje oluşturacaksınız.

Bu projenin sonunda, bir düğmeye basıldığında ekranı siyahtan beyaza değiştiren bir uygulama oluşturacaksınız.

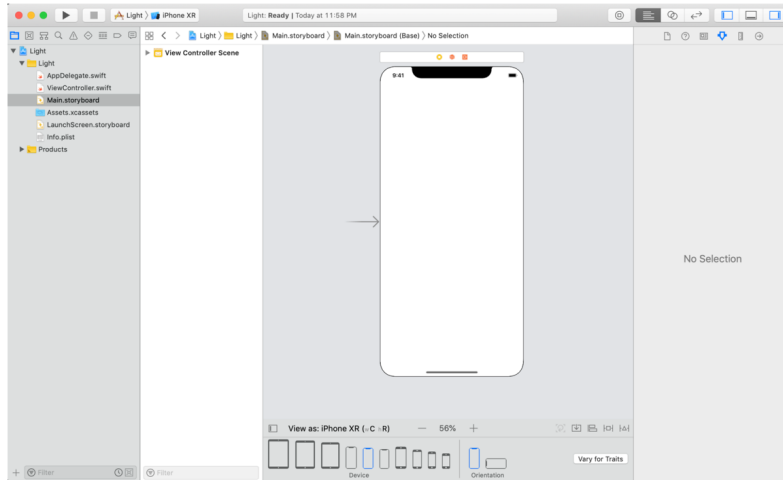
Uygulamayı başarıyla oluşturmak için Xcode dökümanlarını kullanmanız ve döngüler, kontrol akışları ve aksiyonlar oluşturmanız gerekir.

Eğer bir noktada zorlanıyorsanız, cesaretiniz kırılmasın. Kodlamaya devam edin!

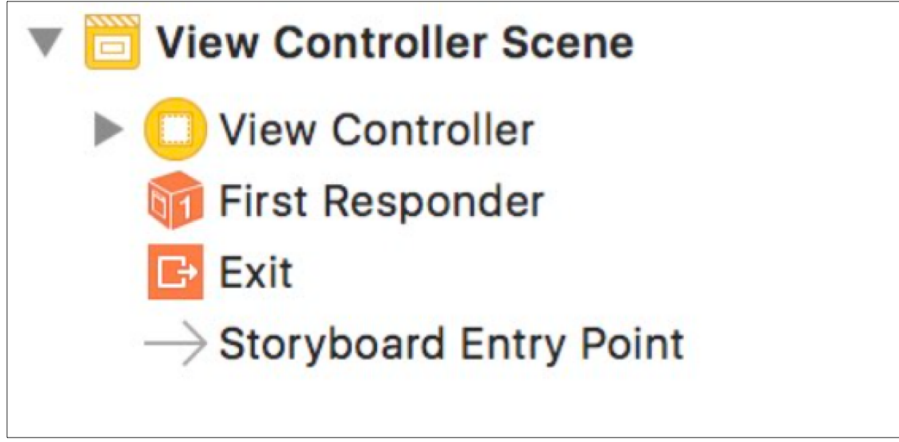
## Birinci Bölüm — Buton ve Aksiyon Oluşturma

1- İlk önce "Single View App" şablonunu kullanarak yeni bir Xcode projesi oluşturun. Projeye "Işığ ı" adını verin. Projeyi oluşturup çalıştırdığınızda, ekranınızda hiçbir şey bulunmadığını fark edeceksinizdir. Bunu yakında değiştireceksiniz. 😊

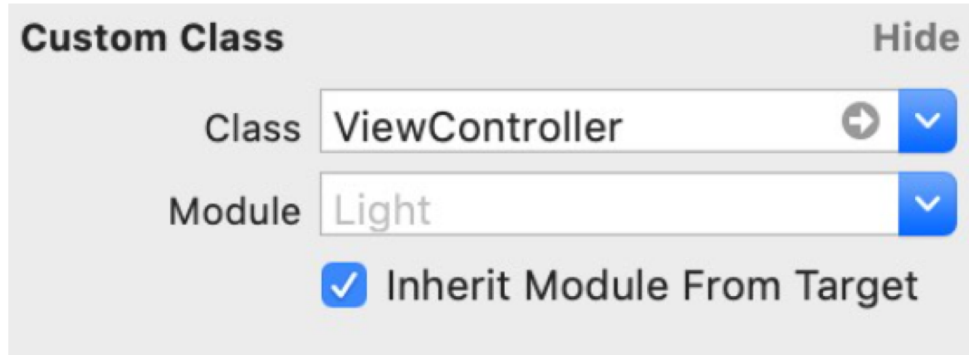
2- Storyboard ekranınızı "Interface Builder" ortamında açmak için projenizdeki `Main.storyboard` dosyasını açın.



3- Karşınıza çıkan ilk ekran bir `ViewController` elemanıdır. `ViewController` elemanı, "Single View App" şablonunun bir parçasıdır. `UIViewController` kütüphanesinin alt sınıfıdır. `Main.storyboard` dosyasında tanımlanan tüm elemanları görmek için "Show Document Outline" adındaki butona tıklayın.



4- Belgenin ana hattında `View Controller` elemanını seçin, ardından bunun `ViewController` sınıfına bağlı olduğunu doğrulamak için araçlar alanının üstündeki "Identity Inspector" butonuna basın.

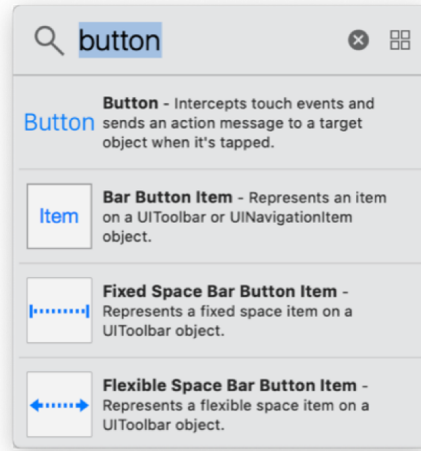


- 5- Kanvasın boyutunu ayarlamak için kanvasın altındaki "View as" düğmesini kullanın. Kanvas boyutunu "iPhone XR" olarak ayarlayın ve sol üstteki menüden simulatörü "iPhone 8" olarak seçin.
- 6- Bu proje için kendi cihazınızı kullanıyorsanız, bilgisayarınızdaki ekran boyutu telefonunuzun ekran boyutuyla aynı olmayabilir. Kanvasın boyutunu cihazınıza uyacak şekilde ayarlamak için kanvasın altındaki "View as" butonunu kullanın, ancak dikey yönde bırakın.

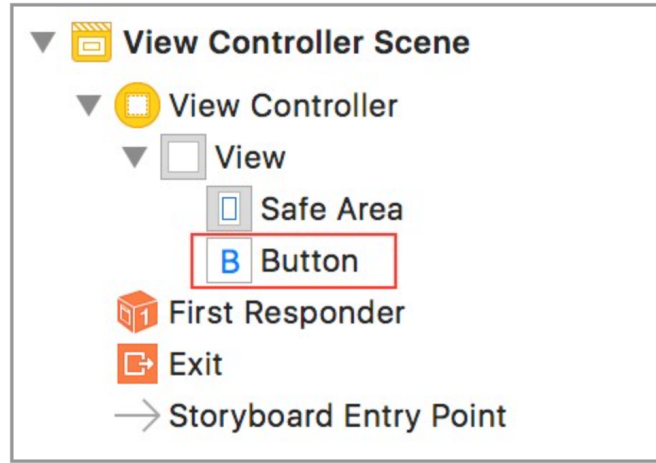


7- `ViewController` elemanının içerisindeki `view` elemanını seçin. Ardından elemanların özelliklerini gösteren "Attribute Inspector" butonuna basın.

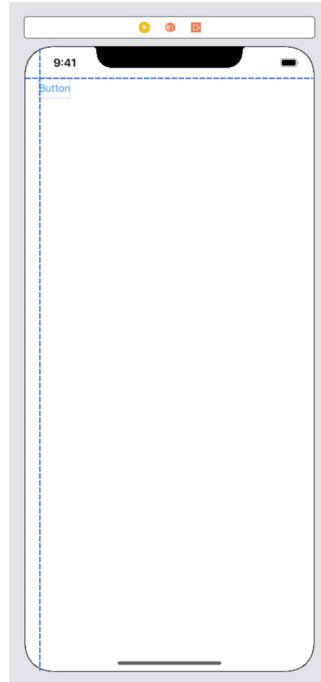
8- Ardından, ışığın açılıp kapanmasını temsil etmek için arka plan rengini değiştirecek bir buton ekleyelim. Bunun için sağ üstteki "+" butonuna basın. Böylece, ekleyebileceğiniz objelerin bir listesine ulaşacaksınız. Listede ilerleyin veya kitaplık alanının üst kısmındaki arama filtresini kullanın.



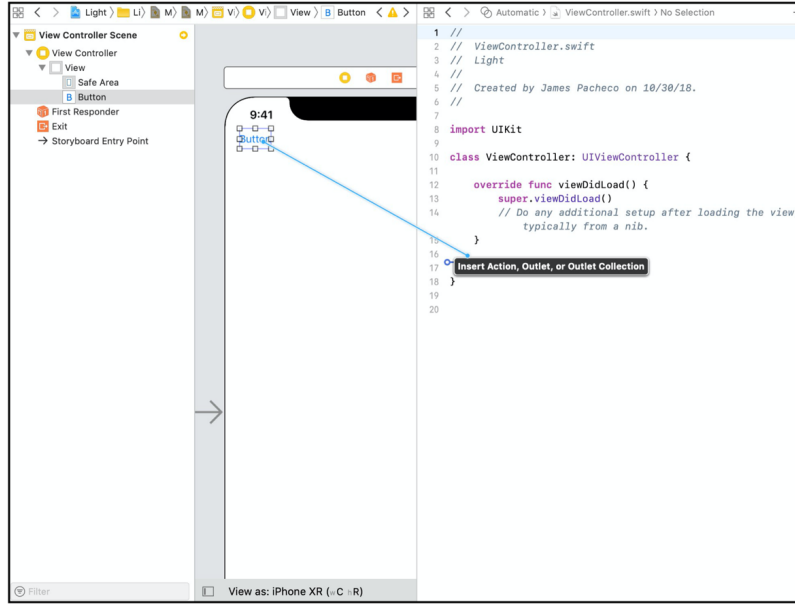
9- Kütüphanede `Button` nesnesini bulun ve `ViewController` elemanının üstüne sürükleyin ve bırakın.



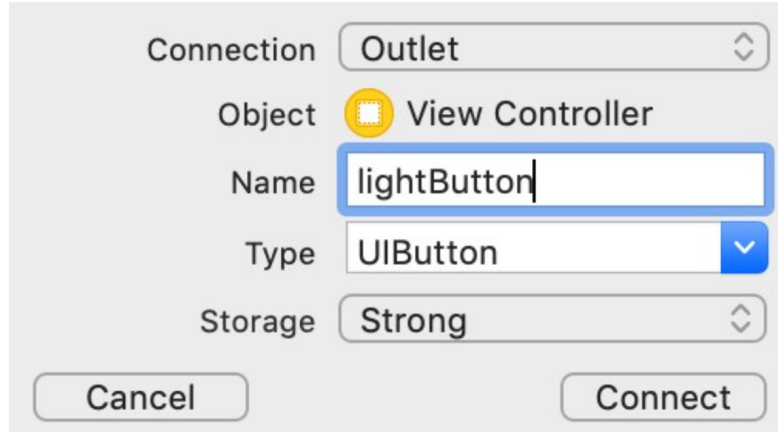
10- Düğmeyi `UIViewController` elemanının sol üst köşesine getirin.



11- Butonu kodlamanız gerekiyor. Xcode çalışma alanını iki bölüme ayırmak için "Assistant Editor" düğmesine tıklayın.



**12-** Şimdi butona çift tıklayarak veya **control + click** yaparak kod yazdığınız alana sürükleyin. İmleci bıraktığınızda, bir eleman tanımlamanız istenir. Bağlantıyı "Action" olarak değiştirin, ardından adını "buttonPressed" olarak ayarlayın. "Connect" butonuna tıklattığınızda, Xcode, düğmeye her dokunulduğunda çağrılacak yeni bir metod oluşturur.



**13-** Uygulamayı çalıştırın. Düğmeye bastığınızda hiçbir şey olmuyor, çünkü `buttonPressed(_ :)` fonksiyonun içerisinde herhangi bir kod yoktur. Fonksiyonun çağrıldığını doğrulamak için, yöntem tanımının içine bir breakpoint yerleştirebilirsiniz. Şimdi butona bastığınızda breakpoint tetiklenir.

```
13 @IBAction func buttonPressed(_ sender: Any) {  
14 }
```

**14-** Fonksiyonun düzgün çalıştığını doğruladığınıza göre, arka plan rengini değiştirmek için bazı kodlar ekleyebilirsiniz. Devam etmeden önce brakpoint elemanını kaldırın.

## İkinci Bölüm — Arka Planı Değiştirme

**15-** Işık yaniyorsa (yani arka plan rengi beyazsa) ışık kapatılmalıdır. Aksi takdirde, ışık kapalıysa (arka plan siyahsa), ışığın açılması gerekir.

**16-** `ViewController` sınıfının üst kısmına yakın bir yerde, ekran beyaz bir arka planla başladığı için `lightOn` adlı bir değişken oluşturun ve başlangıç değerini `true` olarak ayarlayın.

```
var lightOn = true
```

**17-** Düğmeye her dokunulduğunda, değişken, `true` değerinden `false` değerine veya `false` değerinden `true` değerine değişmelidir. Bunun için bir fonksiyon kullanacağız: `toggle()`. Bu fonksiyonu `buttonPressed(_ :)` fonksiyonunun içerisinde kullanın.

```
@IBAction func buttonPressed(_ sender: Any) {
    lightOn.toggle()
}
```

**18-** Arka plan rengini nasıl değiştirileceğini belirlemek için bir if yapısı kullanın. Işık kapalıysa, arka plan rengi siyah, açıksa beyaz olmalıdır.

```
@IBAction func buttonPressed(_ sender: Any) {
    lightOn.toggle()
    if lightOn {
        view.backgroundColor = .white
    } else {
        view.backgroundColor = .black
    }
}
```

**19-** Uygulamanızı çalıştırın; her butona bastığınızda arka plan rengi değişmelidir.

**20-** Uygulamanız büyüdükçe, kodunuzun anlaşılır olması önemli hale gelir. If yapısını doğrudan `buttonPressed(_ :)` fonksiyonunun içine koymak yerine, sürekli olarak kullanabilmek için `updateUI()` adında yeni bir fonksiyon tanımlayın. If yapısını içine taşıyın, sonra `buttonPressed(_ :)` içinde `updateUI()` yöntemini çağırın.

```
@IBAction func buttonPressed(_ sender: AnyObject) {
    lightOn = !lightOn
    updateUI()
}

func updateUI() {
    if lightOn {
        view.backgroundColor = .white
    } else {
        view.backgroundColor = .black
    }
}
```

Bu şimdilik küçük bir değişiklik gibi gözükse de gelecekte daha karmaşık programlarda hataları daha rahat ayırt edebileceksinizdir.

## Üçüncü Bölüm — Butonun Yazısını Değiştirme

Butonunuzu arkaplan rengi değiştirilebilecek şekilde kodladınız. Ancak butonunun yazısı ışığı açıp kapadıkça değişmiyor. Şu an kod sayfanızda buton ile alakalı yazdığınız kodların hiçbiri çalışmaz çünkü butonunuzu daha tanımlamadınız.

**21-** Bunu yapabilmek için butonunuzu aynı yukarıda yaptığımız gibi kodlama sayfasına bağlamalıyız.

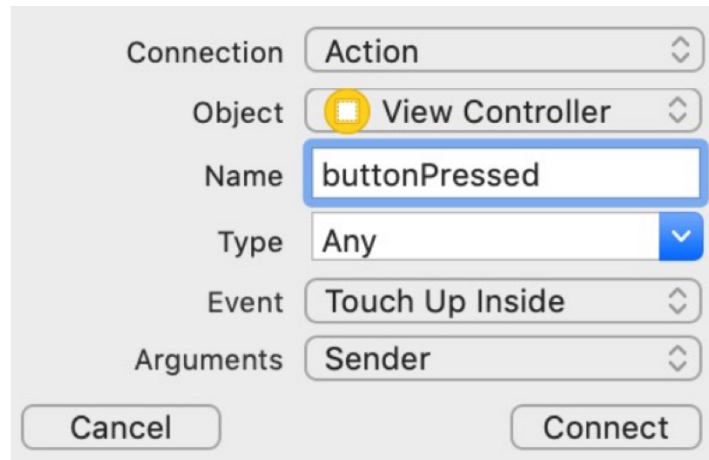
**Control + click** yaparak kodlama sayfasına doğru sürükleyin.

Oluşturduğunuz elemanın detaylarını soldan sağa inceleyebilirsiniz.

- Daire - Dairenin içi dolu olduğunda elemanı oluşturduğunuz anlamına gelir.
- `@IBOutlet` - Bu bir terimdir. Xcode'daki elemanlar bu terim altında toplanır.
- `var lightButton` - Butonunuz için oluşturduğunuz değişkeni tanımlar.
- `UIButton!` - Tanımladığınız değişken bir `UIButton!` türündendir. Ünlem işareti, bilgisayar, eleman bağına ulaşamazsa programın çalışmayacağını belirtir.

**22-** Bunu yapabilmek için butonunuzu aynı yukarıda yaptığımız gibi kodlama sayfasına bağlamalıyız.

**Control + click** yaparak kodlama sayfasına doğru sürükleyin.



Elemanı tanımladığınıza göre butonun özelliklerini kodunuzda değiştirebilirsiniz. Işık kapanıp açıldıkça butonun başlığını değiştirmekle başlayalım.

**23-** `lightOn.toggle()` satırına bir breakpoint ekleyin. Programınızı çalıştırdığınızda, arkaplan değişmeden önce kodunuz duracaktır.

**24-** İlk satırı çalıştırmak için "Step Over" butonuna basın. Şimdi uygulamanız `updateUI()` fonksiyonunda durmuştur. `updateUI()` fonksiyonunun ilk satırını çalıştırmak için "Step Into" butonuna basın. `updateUI()` bir fonksiyon olduğundan "Step Over" yerine o fonksiyonu adım adım çalıştırabilmek için "Step Into" butonunu kullanmak gerekir.

```
15  @IBAction func buttonPressed(_ sender: Any) {  
16      lightOn.toggle()  
17      updateUI()  
18  }
```

Thread 1: breakpoint 1.1

**25-** `updateUI()` fonksiyonunun içindekileri inceleyin ve yaptıklarını bir cümleye dökmeye çalışın. Swift rahatça anlaşılabilen bir kodlama dili olduğundan "Eğer ışıklar açıksa arkaplan rengini beyaz yap. Eğer ışıklar açık değilse arkaplan rengini siyah yap." cümlesine benzer bir cümle oluşturacaksınız.

Şimdi butonunuzun başlığının hangi durumda ne olması gerektiğini düşünün. Işıklar kapalı ise açık, ışıklar açık ise de kapalı olmalıdır.

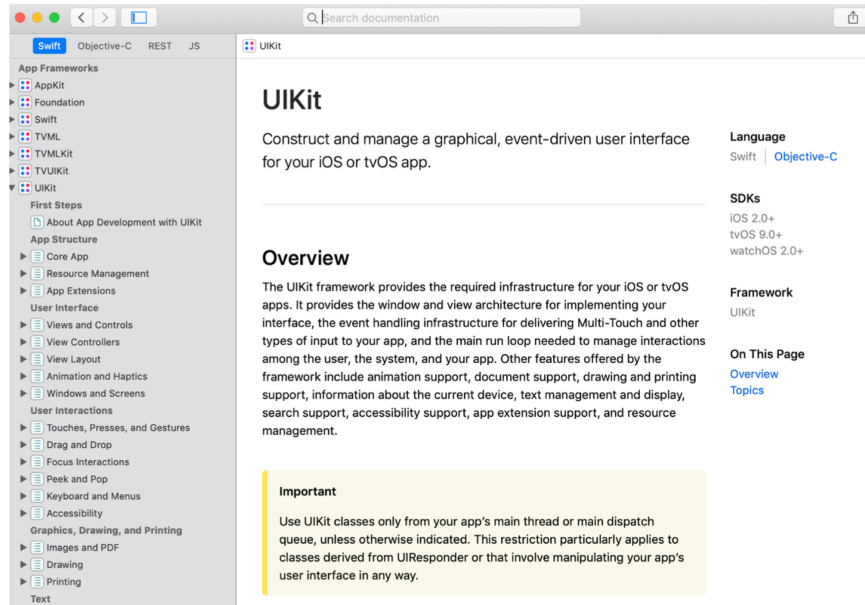
**26-** Geriye kalan tek şey, bu bulmacanın parçalarını birleştirmektir. Eğer ışıklar açıkken butonumuzun adını "Off" yapmak istiyorsak `updateUI()` fonksiyonunun içindeki if yapısında ışıkların açık olduğu bölüme butonumuzun adını "Off" olarak değiştirmeniz gerekir.



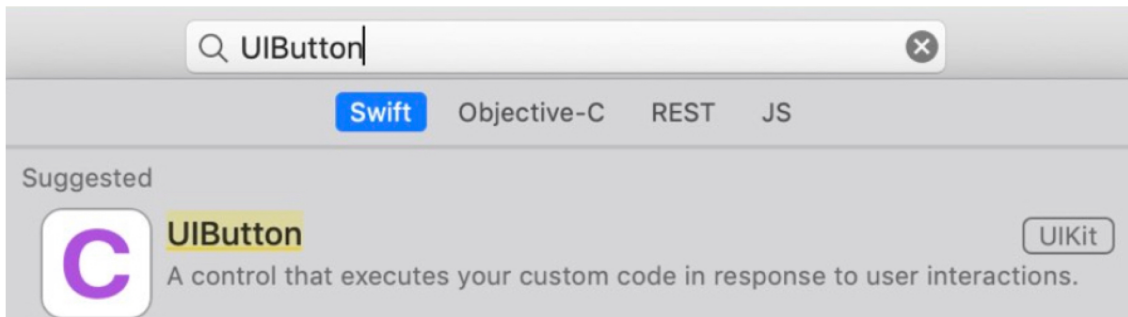
```
func updateUI() {  
    if lightOn {  
        view.backgroundColor = .white  
    } else {  
        view.backgroundColor = .black  
    }  
}
```

**27-** Butonunun adını ayarlamayı unuttuysanız Xcode'un "Documentation" özelliğinden yararlanabilirsiniz. Bir kodun nasıl yazıldığını unutursanız bu özellik size yardım edecektir. Yukarıdaki seçeneklerinizden "Window > Developer" bölgesine gidin ve karşınıza bir "Documentation" ekranı çıkacaktır. "Documentation" ekranında herhangi bir elemanın ismini aratıp bu eleman hakkında bilgi edinebileceksinizdir.

**28-** "Documentation" ekranında arama çubuğuna `UIButton` yazın. "Configuring the Button" başlığını bulun. Butonunuzun ismini değiştiren `func setTitle(String?, for: UIControl.State)` fonksiyonu ve tanımını bulacaksınız.



**29-** Bu fonksiyonu butonun ismini değiştirmek için kullanın. Kodunuz şu şekilde görülmelidir:



**30-** Programınızı çalıştırın. Uygulamanız ilk açıldığında butonun isminin "Button" olarak kaldığını ancak sonra değiştiğini fark edeceksinizdir. Sizce bu sorun neden kaynaklanıyor?

```
func updateUI() {
    if lightOn {
        view.backgroundColor = .white
        lightButton.setTitle("Off", for: .normal)
    } else {
        view.backgroundColor = .black
        lightButton.setTitle("On", for: .normal)
    }
}
```

**31-** Uygulamanızın ilk çalıştırdığı kodların `viewDidLoad()` fonksiyonun altında olduğunu unutmayın. Şuan o fonksiyonun altında hiç bir kod bulunmadığını fark edeceksinizdir. Bu yüzden uygulamanız ilk çalıştığında butonunuzda "Button" yazmaktadır.

**32-** Bunu düzeltebilmek için `updateUI()` fonksiyonunu `viewDidLoad()` fonksiyonunun içinde çağırın.

**33-** Programınızı tekrar çalıştırın. Başlangıçta butonunuzda "Off" yazdığını göreceksinizdir.

```
1 import UIKit
2
3 class ViewController: UIViewController {
4
5     var lightOn = true
6     @IBOutlet var lightButton: UIButton!
7
8     override func viewDidLoad() {
9         super.viewDidLoad()
10        updateUI()
11    }
12
13    @IBAction func buttonPressed(_ sender: Any) {
14        lightOn.toggle()
15        updateUI()
16    }
17
18    func updateUI() {
19        if lightOn {
20            view.backgroundColor = .white
21            lightButton.setTitle("Off", for: .normal)
22        } else {
23            view.backgroundColor = .black
24            lightButton.setTitle("On", for: .normal)
25        }
26    }
27 }
28
```

## Dördüncü Bölüm — Uygulamanızı Kullanıcıya Uygun Tasarlamak

Artık çalışan bir uygulamanız var! Ancak kullanıcı bu uygulamanın tasarımına bakınca beğenir mi? Hangi özelliklerini değiştirebilirsiniz?

Butondan bahsetmek gerekirse üstündeki yazı hiç güzel gözüküyor. Belki fontu veya rengi değiştirilebilir. Ancak kullanıcının bu butonun üstünde yazı görmesi gerekiyor mu?

Butonun konumu uygun mu? Sadece küçük bir alana basabilme olacağı garip gelmiyor mu?

**34-** Butonun başlığını tamamen silip butonumuzu ekranın tamamını kaplayacak boyutta ayarlamak mantıklı bir karar olabilir. Böylece, kullanıcı uygulamanın herhangi bir yerine bastığında ışıkları açıp kapatabilecektir.

**35-** Butonun başlığını silmek için önceden `updateUI()` fonksiyonunuzun içine kodladığınız `setTitle()` onksiyonlarını silebilirsiniz.

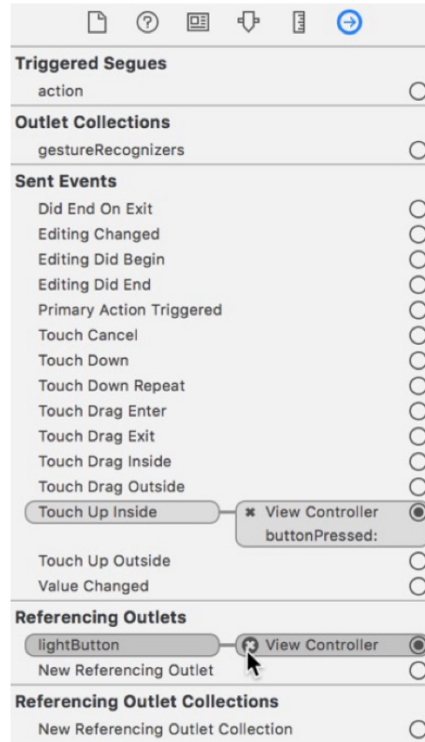


Bu tasarımda butonun özelliklerinin değişmesi gerekmez. Böylece daha güzel bir tasarım yaratmışken, aynı zamanda daha basit ve anlaşılır bir kodumuz olur.

**36-** Artık önceden tanımladığımız `IBOutlet` fonksiyonuna ihtiyacımız olmadığından kodu silebiliriz. Bunu yapabilmek için "Interface Builder" kısmından butonumuzu seçip "Connections Inspector" kısmına ulaşmalıyız. Bu kısımda buton elemanınıza bağlı olan tüm bağlantıları görebileceksinizdir. `lightButton` diye tanımladığımız `IBOutlet` fonksiyonunu silelim.

```
func updateUI() {
    if lightOn {
        view.backgroundColor = .white
    } else {
        view.backgroundColor = .black
    }
}
```

37- Artık buton elemanınıza herhangi bir obje bağlı olmadığı için, `IBOutlet` satırının solunda yer alan dairenin içi boş olacaktır. Şimdi `lightButton` değerini silebilirsiniz.



38- If-else yapınızda her iki durumda da aynı değişiklikler (arkaplanın rengini değiştirme) yapıldığı için bu satırı tek bir satıra indirgeyebilirsiniz. Böylece kodunuz daha anlaşılır ve kısa gözükecektir.

○ **@IBOutlet var lightButton: UIButton!**

39- Uygulamanızı çalıştırın ve yazdığınız kodların çalıştığına emin olun.

```
func updateUI() {
    view.backgroundColor = lightOn ? .white : .black
}
```