TEB

iOS Programlama Eğitimi
7. Gün

Failable Initializers

```
struct Animal {
    let species: String
    init?(species: String) {
        if species.isEmpty { return nil }
        self.species = species
    }
}
```

Failable Initializers

```
let someCreature = Animal(species: "Giraffe")
// someCreature is of type Animal?, not Animal

if let giraffe = someCreature {
    println("A \(giraffe.species) was initialized")
}
// prints "A Giraffe was initialized"
```

Failable Initializers

```
let anonymousCreature = Animal(species: "")
// anonymousCreature is Animal?, not Animal

if anonymousCreature == nil {
    println("Creature could not be initialized")
}
// prints "Creature could not be initialized"
```

Failable Initializers

- Swift'te init metodları geriye birşey dönmez
- Sadece burada init işleminin başarısız olduğunu belirtmek için nil döner
- Aynı parametre tipi ve ismi ile hem normal hem de failable Init tanımlanamaz.

Failable Initializers

```
enum TemperatureUnit {
    case Kelvin, Celsius, Fahrenheit
    init?(symbol: Character) {
        switch symbol {
        case "K":
            self = .Kelvin
        case "C":
            self = .Celsius
        case "F":
            self = .Fahrenheit
        default:
            return nil
        }
    }
}
```

Views

- “View” temelde, dörtgen bir alanı ifade eder.
- İlgili alana çizim yapar ve kullanıcı aksiyonlarını karşılar.
- Her bir view’in, tek bir superView’i vardır.
- Ancak birçok subview’i olabilir.
- Subview’lerin sırası önemlidir.
- Son eklenen en üstte görünür.

Window

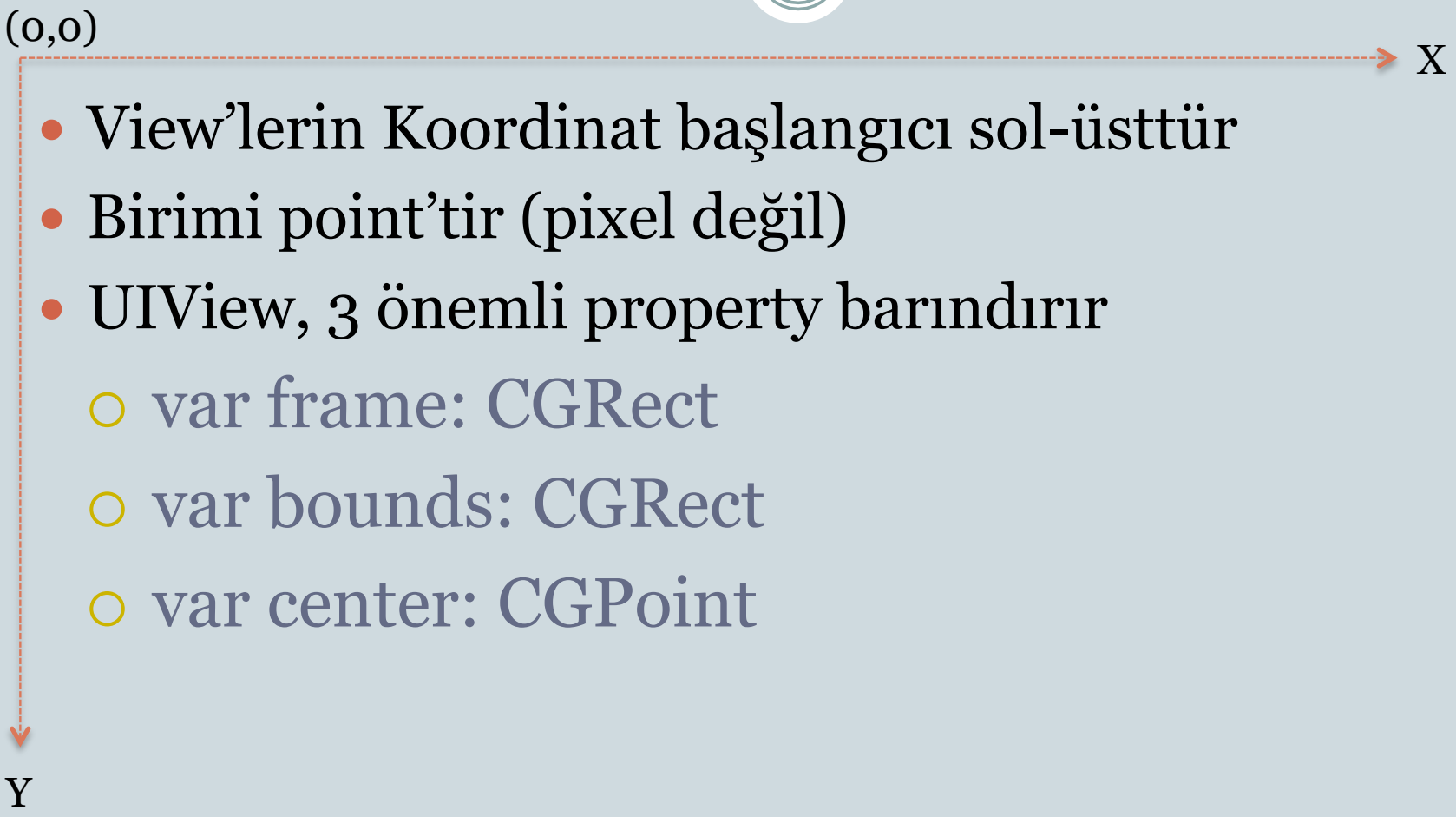
```
class UIWindow : UIView {
    ...
}
```

- View hiyerarşisinin en tepesindeki view
- Uygulamalarda tek bir window olur
- Storyboard'lardan sonra pek kullanılmıyor

View Coordinates

- CGFloat
  - Grafik kütüphanesinde kullanılan float tipi
- CGPoint
  - **x** ve **y** adında CGFloat değerleri tutan bir struct
- CGSize
  - **width** ve **height** adında CGFloat değerleri tutan bir struct
- CGRect
  - **origin** adında bir CGPoint
  - **size** adında bir CGSize tutan bir struct

View Coordinates

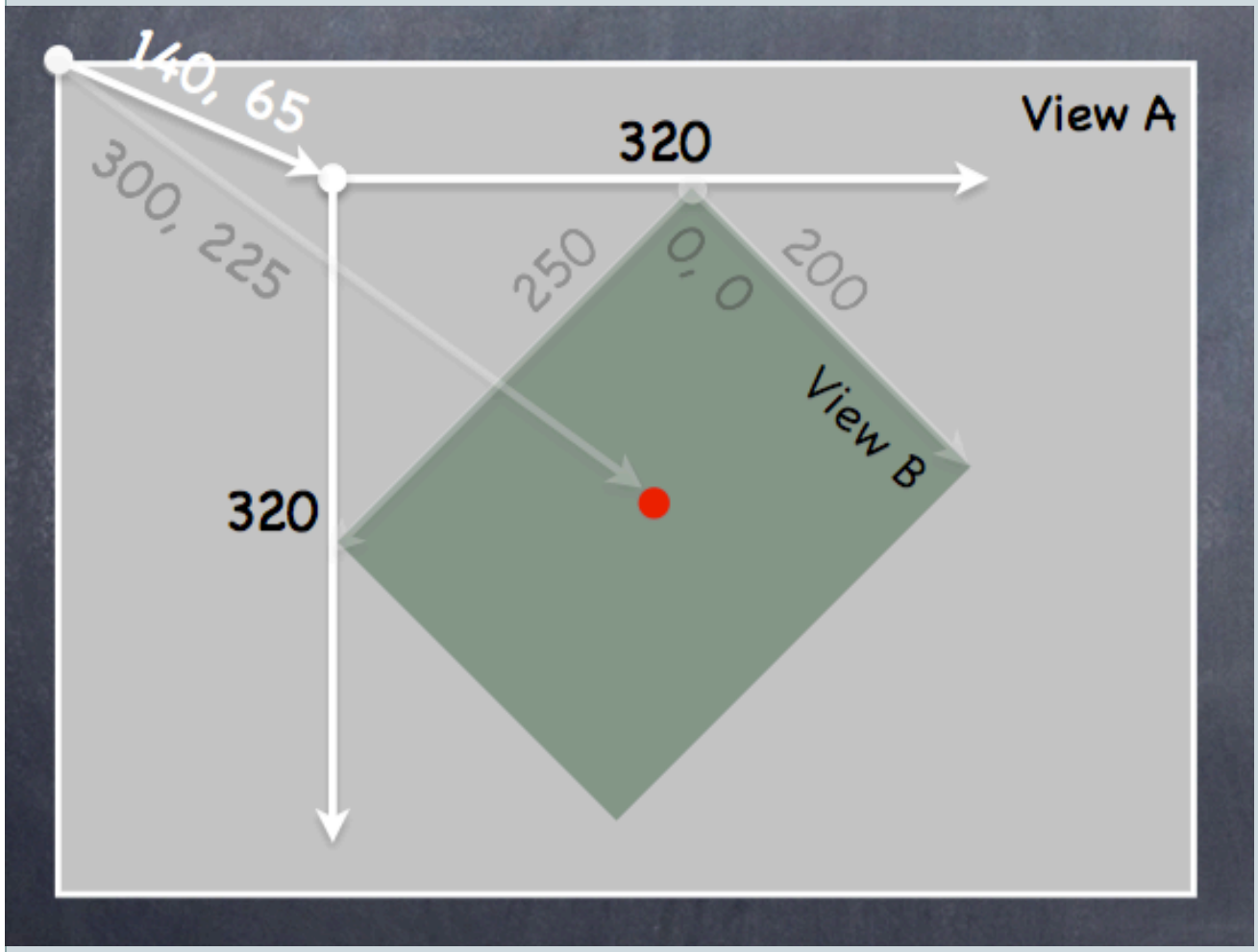


- View'lerin Koordinat başlangıcı sol-üsttür
- Birimi point'tir (pixel değil)
- UIView, 3 önemli property barındırır
  - var frame: CGRect
  - var bounds: CGRect
  - var center: CGPoint

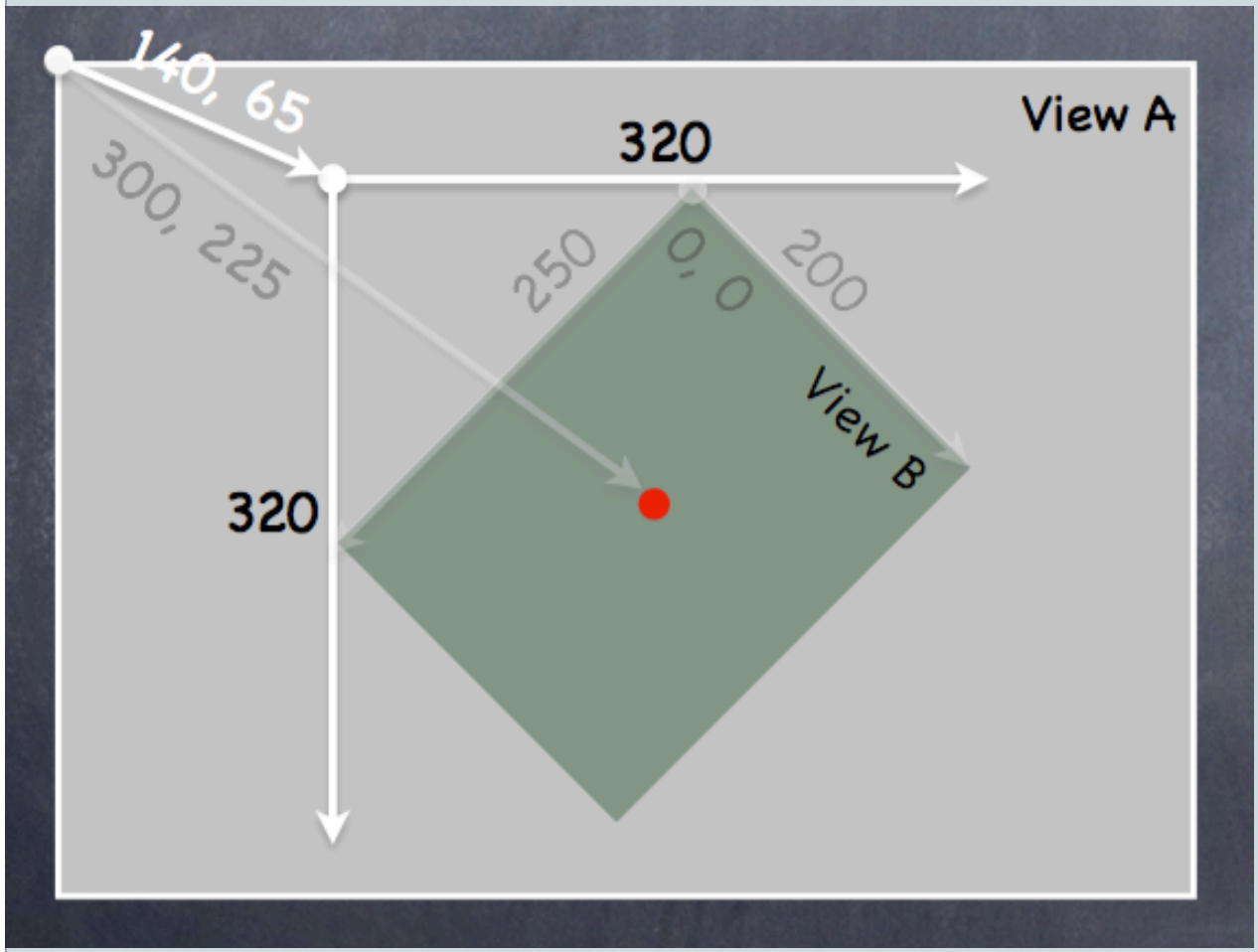
View Coordinates

- bounds
  - Bir view'in kendisine göre koordinat durumu
- frame
  - Bir view'in superView'ine göre olan koordinat durumu
- center
  - Bir view'in superView'e göre merkez noktası

View Coordinates

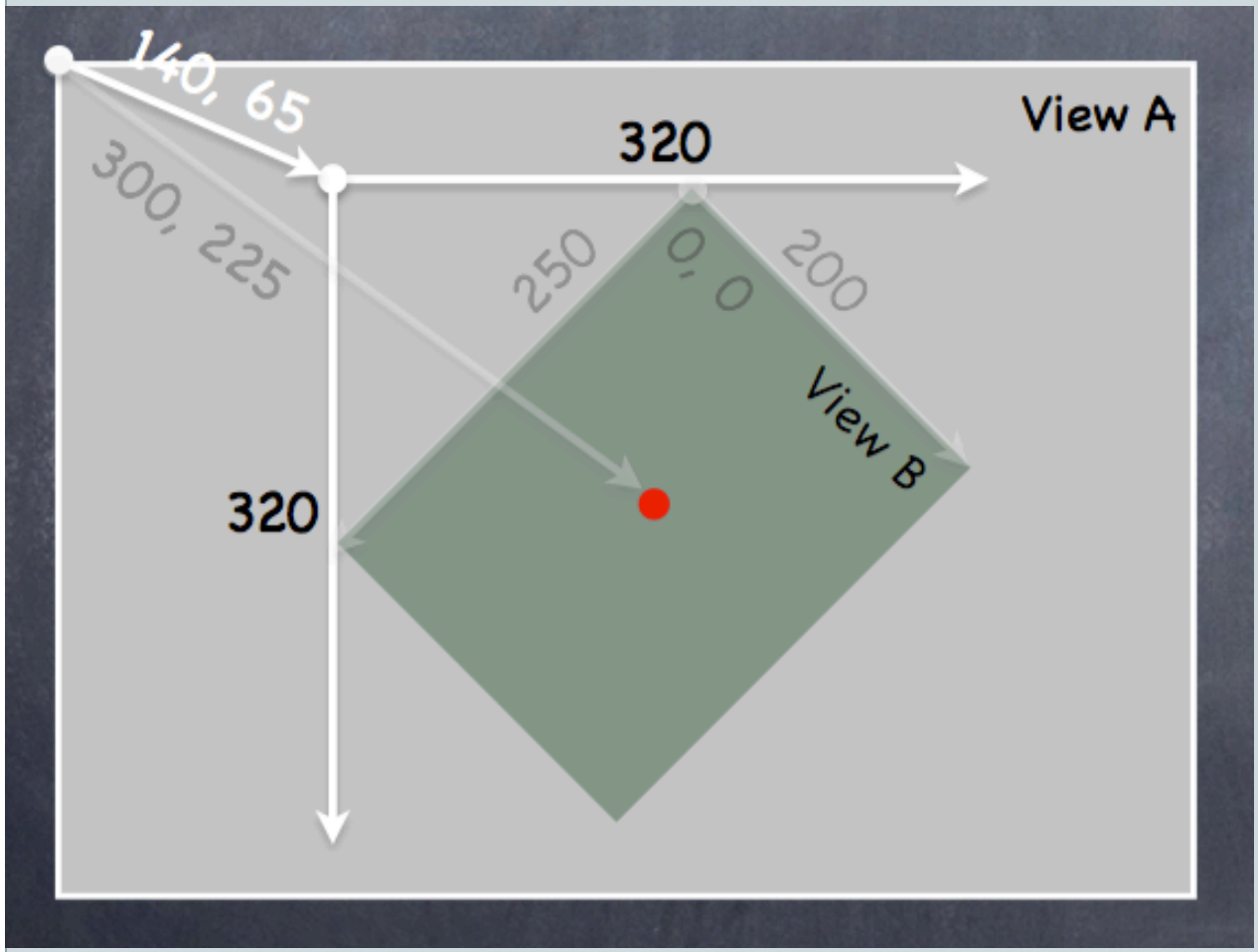


View Coordinates



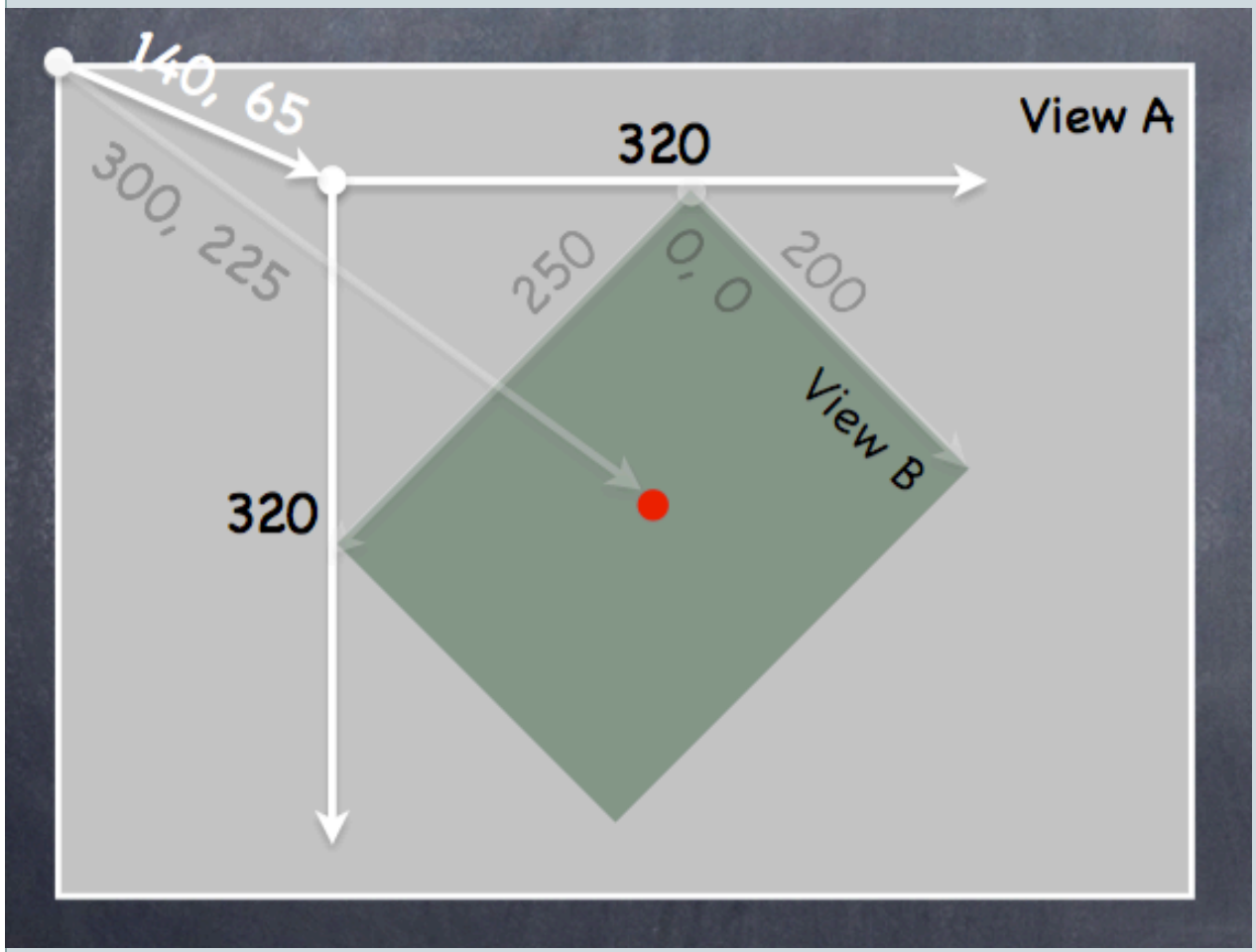
B'nin **bounds** değeri ?

View Coordinates



B'nin **bounds** değeri **((0,0) , (200,250))**

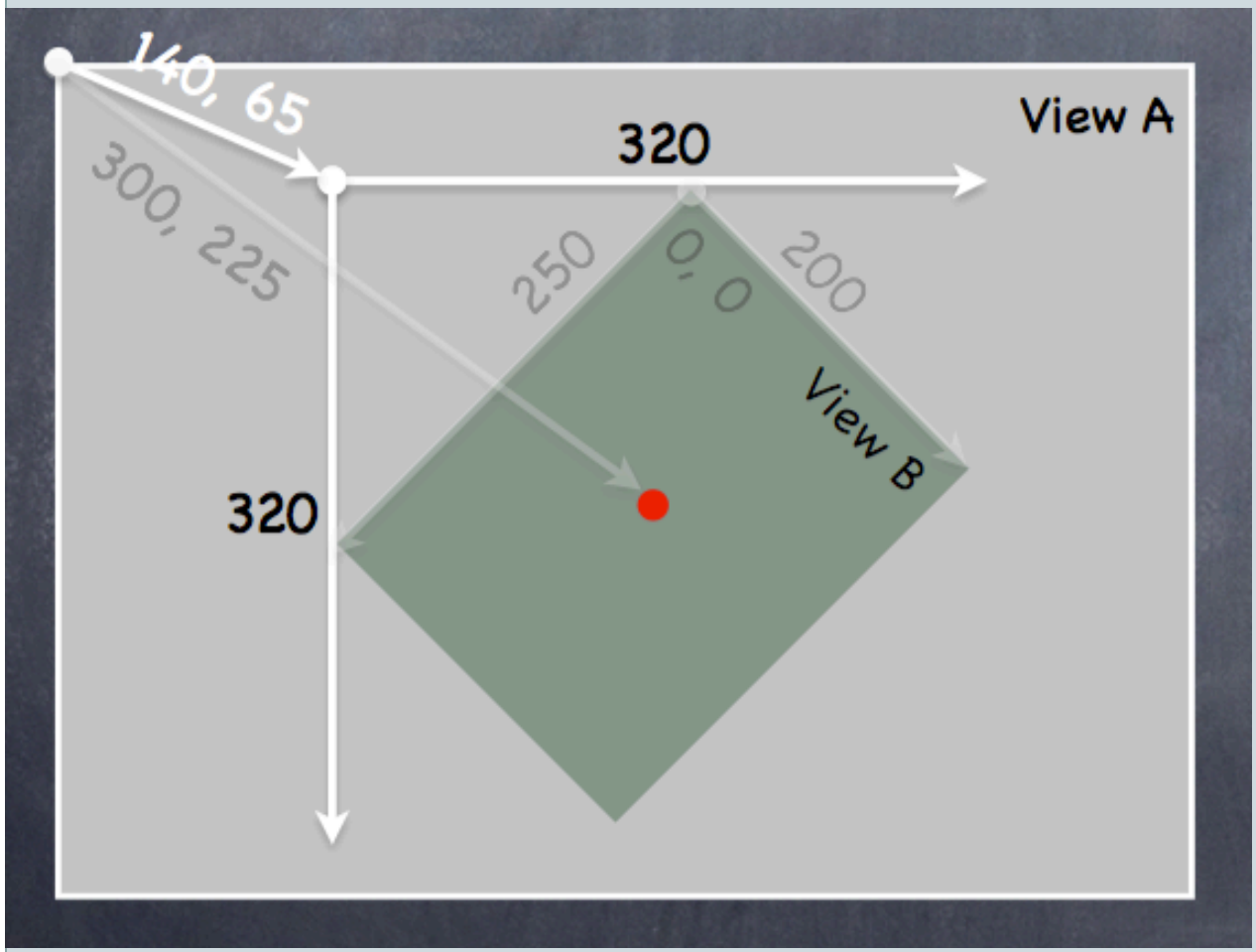
View Coordinates



B'nin **bounds** değeri **((0,0) , (200,250))**

B'nin **frame** değeri ?

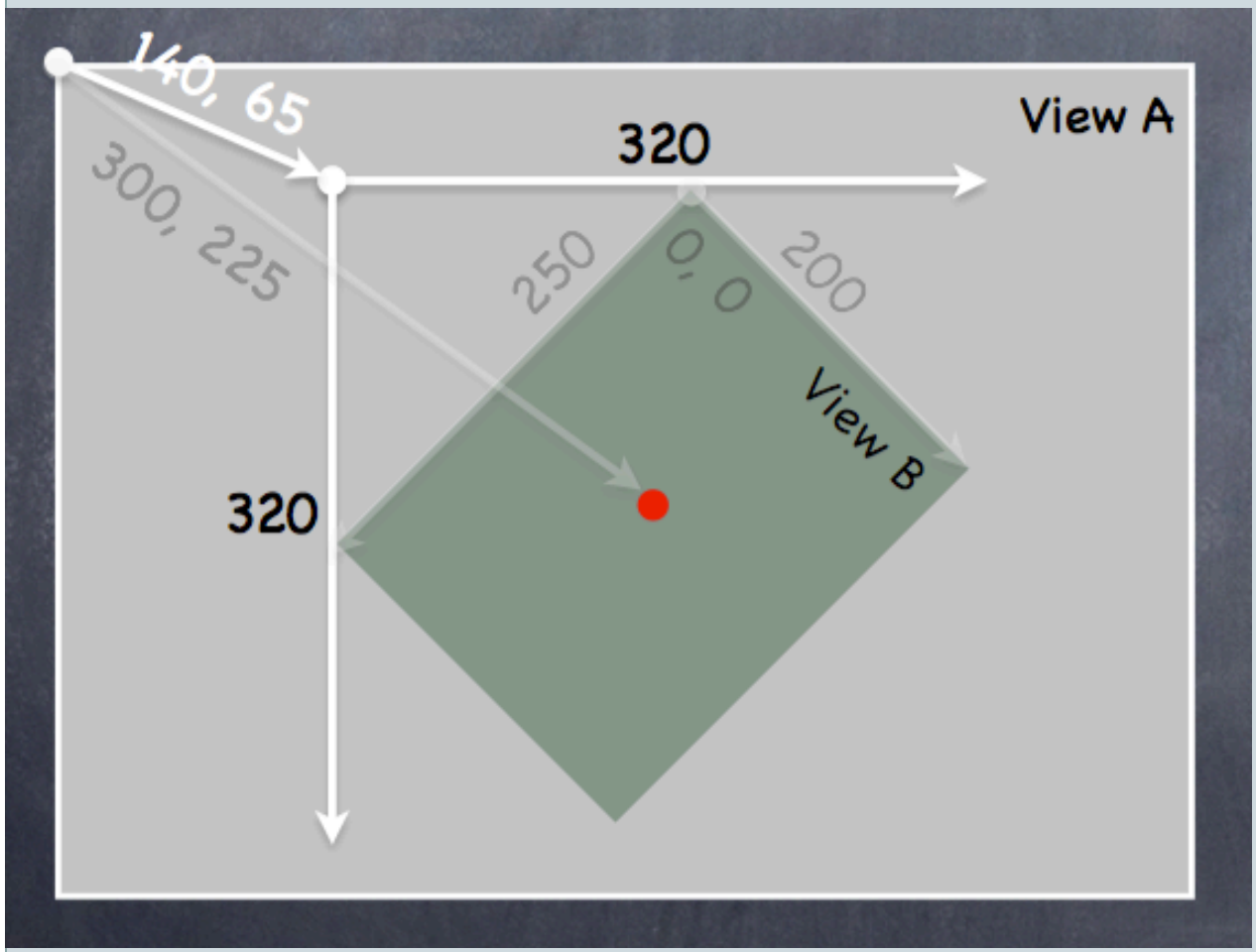
View Coordinates



B'nin **bounds** değeri
((0,0) , (200,250))

B'nin **frame** değeri
((140,65),(320,320))

View Coordinates

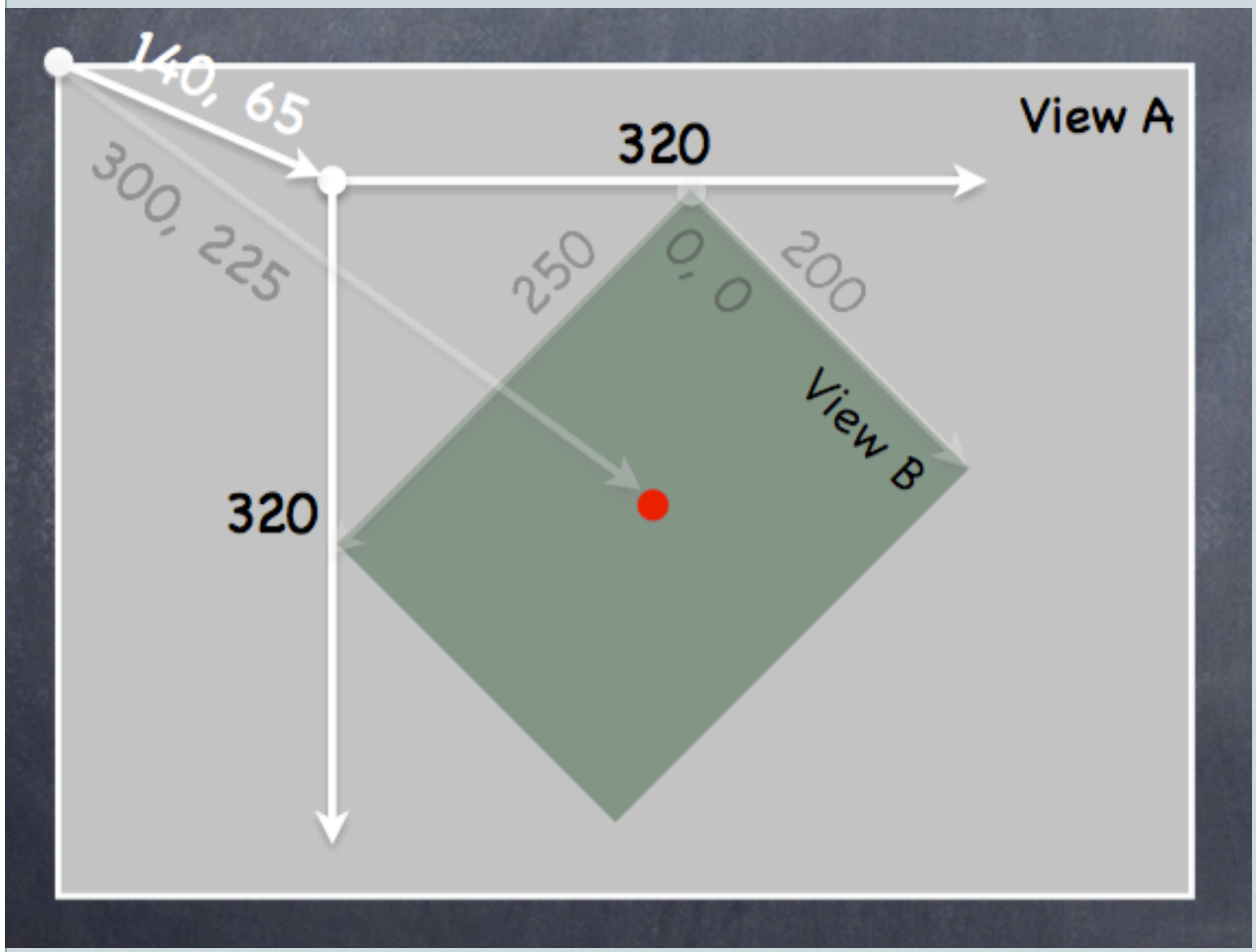


B'nin **bounds** değeri
((0,0) , (200,250))

B'nin **frame** değeri
((140,65),(200,250))

B'nin **center** değeri ?

View Coordinates



B'nin **bounds** değeri
((0,0) , (200,250))

B'nin **frame** değeri
((140,65),(320,320))

B'nin **center** değeri
(300,225)

View Coordinates

- self.frame = self.superView.frame
  yerine
- self.frame = self.superView.bounds
  kullanın

View Animations

```
extension UIView {

    @availability(iOS, introduced=4.0)
    class func animateWithDuration(duration:
NSTimeInterval, delay: NSTimeInterval, options:
UIViewAnimationOptions, animations: () -> Void, completion:
((Bool) -> Void)?)

    @availability(iOS, introduced=4.0)
    class func animateWithDuration(duration:
NSTimeInterval, animations: () -> Void, completion: ((Bool)
-> Void)?) // delay = 0.0, options = 0

    @availability(iOS, introduced=4.0)
    class func animateWithDuration(duration:
NSTimeInterval, animations: () -> Void) // delay = 0.0,
options = 0, completion = NULL

...
}
```

UIViewAnimation

- duration = NSTimeInterval
- delay = NSTimeInterval
- option = UIViewAnimationOptions
- animations = ( ) -> Void
- completion = (Bool) -> Void?

UIViewAnimationOptions

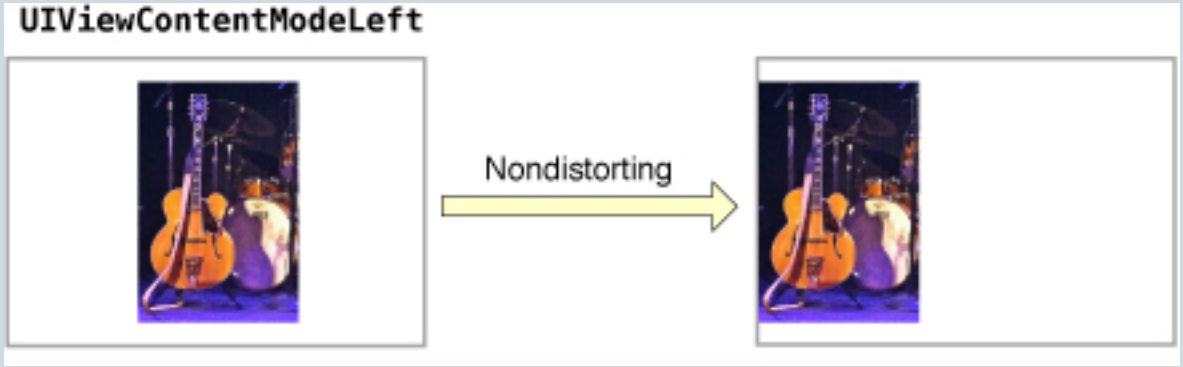
```
struct UIViewAnimationOptions : RawOptionSetType {
    ...
    static var CurveEaseInOut: UIViewAnimationOptions { get } // default
    static var CurveEaseIn: UIViewAnimationOptions { get }
    static var CurveEaseOut: UIViewAnimationOptions { get }
    static var CurveLinear: UIViewAnimationOptions { get }

    static var TransitionNone: UIViewAnimationOptions { get } // default
    static var TransitionFlipFromLeft: UIViewAnimationOptions { get }
    static var TransitionFlipFromRight: UIViewAnimationOptions { get }
    static var TransitionCurlUp: UIViewAnimationOptions { get }
    static var TransitionCurlDown: UIViewAnimationOptions { get }
    static var TransitionCrossDissolve: UIViewAnimationOptions { get }
    static var TransitionFlipFromTop: UIViewAnimationOptions { get }
    static var TransitionFlipFromBottom: UIViewAnimationOptions { get }
    ...
}
```

UIView Animation In Action

```
var aView = UIView(frame: CGRect(x: 0, y: 0, width: 100, height:
100))
aView.backgroundColor = UIColor.greenColor()
UIView.animateWithDuration(1.0, animations: { () -> Void in
    var newFrame = aView.frame
    newFrame.size.width = 200
    newFrame.size.height = 200
    aView.frame = newFrame
    }) { (completed:Bool) -> Void in
        aView.frame = CGRect(x: 100, y: 100, width: 10, height: 10)
}
```

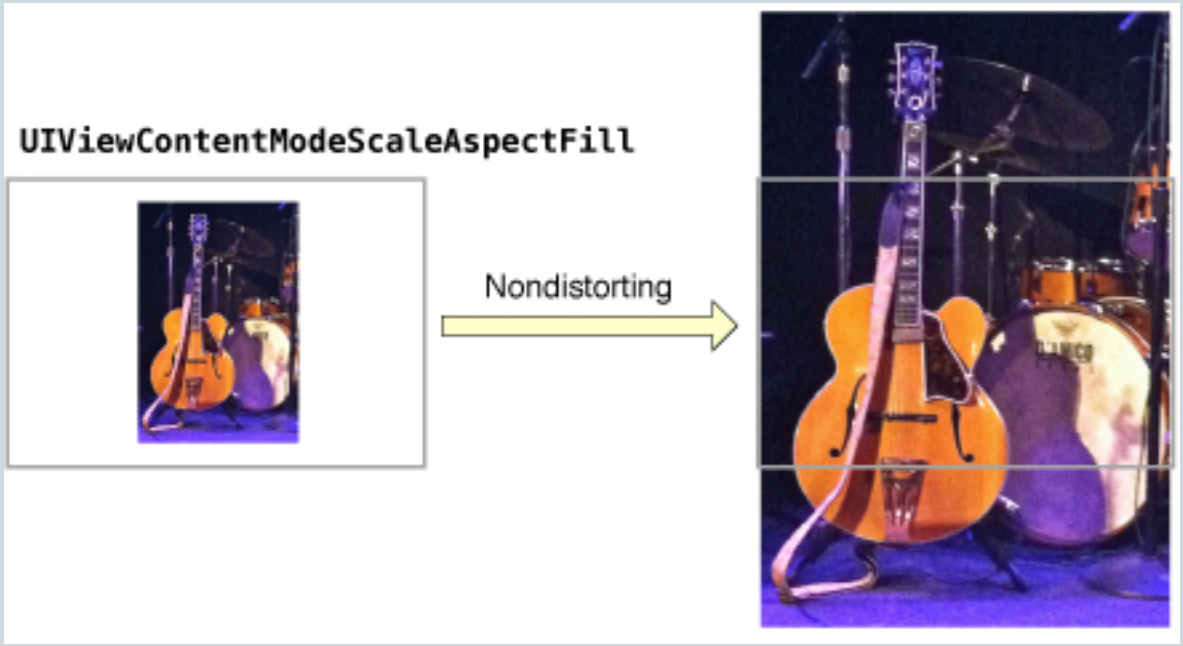
Content Modes

UIViewContentModeLeft



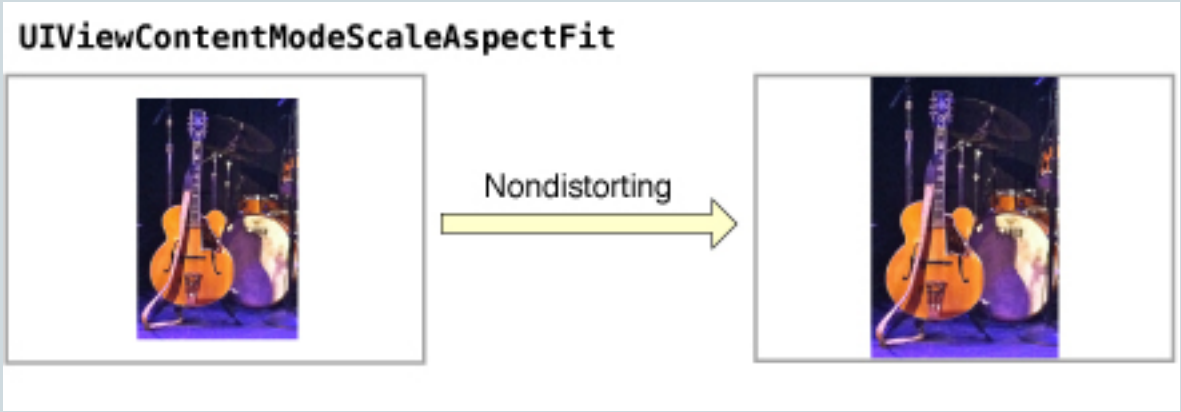
Content Modes

UIViewContentModeAspectFill



Content Modes

UIViewContentModeAspectFit



Content Modes

UIViewContentModeScaleToFill