



Swift & iOS Uygulama Geliştirme Eğitimi

Proje Hakkında



- Amaç: Kurs süresince öğrenilenleri gerçek bir uygulama geliştirme sürecine dönüştürmek.
- Yöntem:
 - Uygulamanın içeriği ve senaryosu belirlenir.
 - Uygulamada olacak ekranlar belirlenir.
 - Sunucu tarafı Parse kullanılarak hazırlanır.
 - Belirli seviye geçildikten sonra uygulama her hafta geliştirilmeye devam edilir.

Geçen Hafta



Temel Değerler



- Değişken ve sabitler
 - var
 - let
- Tipler
 - Int - Int32 – Int64 - UInt – UInt32 – UInt64
 - Float (32 bit)
 - Double (64 bit)
 - Bool (true, false)
 - String (text)

Optional Tipler



- Bu değişkenin içinde değer olabilir de, olmayabilir de

Optional Tipler



- Bu değişkenin içinde değer olabilir de, olmayabilir de
- Değer yoksa nil

Optional Tipler



- Bu değişkenin içinde değer olabilir de, olmayabilir de
- Değer yoksa nil
- Optional olmayan değişkene nil atamaya çalışırsanız, uygulama kapanır. (crash)

Optional Tipler



- Bu değişkenin içinde değer olabilir de, olmayabilir de
- Değer yoksa nil
- Optional olmayan değişkene nil atamaya çalışırsanız, uygulama kapanır. (crash)
- Bir değişkene atanacak değerın nil olma ihtimali var onu optional olarak tanımlamak gerekir.

Optional Tipler



- Bu değişkenin içinde değer olabilir de, olmayabilir de
- Değer yoksa nil
- Optional olmayan değişkene nil atamaya çalışırsanız, uygulama kapanır. (crash)
- Bir değişkene atanacak değerın nil olma ihtimali var onu optional olarak tanımlamak gerekir.
- ?

Optional Tipler



- Bu değişkenin içinde değer olabilir de, olmayabilir de
- Değer yoksa nil
- Optional olmayan değişkene nil atamaya çalışırsanız, uygulama kapanır. (crash)
- Bir değişkene atanacak değerın nil olma ihtimali var onu optional olarak tanımlamak gerekir.
- Optional değişkene ilk değer atamak zorunda değilsiniz.

Optional Tipler



• ?

Optional Tipler



- ?
- `var serverResponseCode : Int ?`

Optional Tipler



- ?
- `var serverResponseCode : Int ?`
- `var code:Int = serverResponseCode!`

Optional Tipler



- ?
- `var serverResponseCode : Int ?`
- `var code:Int = serverResponseCode!`
- !

Optional Tipler



- ?
- `var serverResponseCode : Int ?`
- `var code:Int = serverResponseCode!`
- !
- `var serverResponseCode : Int !`

Optional Tipler



- ?
- `var serverResponseCode : Int ?`
- `var code:Int = serverResponseCode!`
- !
- `var serverResponseCode : Int !`
- `var code:Int = serverResponseCode`

Optional Binding



```
if let constantName = someOptional {  
    statements  
}
```

Optional Binding



```
let possibleNumber = "123"
let convertedNumber = possibleNumber.toInt()

if let actualNumber = convertedNumber {

    println(" \'\'(possibleNumber)\'\' içerikli String, sayıya
çevrilebildi: \'\'(actualNumber)\'\'")

}
else {

    println(" \'\'(possibleNumber)\'\' içerikli String, sayıya
çevrilemedi!")

}
```

Function



```
// Parametre almayan ve dönüş değeri olmayan fonksiyon
func merhabaDe() {
    println("Merhaba!")
}
```

```
// Bir Parametre alan ve dönüş değeri olmayan fonksiyon
func merhabaDe(isim:String) {
    println("Merhaba: \ (isim)")
}
```

Function



```
// İki Parametre alan ve dönüş değeri olmayan fonksiyon
func birseySoyle(birsey:String , isim:String) {
    println(birsey + " " + isim)
}
```

```
// İki Parametre alan ve dönüş değeri Int olan fonksiyon
func topla(ilkSayi:Int , ikinciSayi:Int) -> Int {
    return ilkSayi + ikinciSayi
}
```

Function



```
// Int parametre alan ve Tuple dönen fonksiyon
func ikiyeBol(sayi:Int) -> (Int,Int) {
    var bolum = sayi / 2
    var kalan = sayi % 2

    return (bolum,kalan)
}
```

Struct



```
1. struct Vapur {  
2.     var kalkış : String  
3.     var varış : String  
4.     var yolcuSayısı: Int  
5. }
```

```
var kadıköyVapuru = Vapur(kalkış: "Karaköy", varış: "Kadıköy", yolcuSayısı: 150)
```

Struct vs Class



1. `class SomeClass {`
2. `var name: String`
3. `init(name: String) {`
4. `self.name = name`
5. `}`
6. `}`

7. `var aClass = SomeClass(name: "Bob")`
8. `var bClass = aClass` // aClass and bClass now reference the same instance!
9. `bClass.name = "Sue"`

10. `println(aClass.name)` // "Sue"
11. `println(bClass.name)` // "Sue"

Struct vs Class



1. `struct SomeStruct {`
2. `var name: String`
3. `init(name: String) {`
4. `self.name = name`
5. `}`
6. `}`

7. `var aStruct = SomeStruct(name: "Bob")`
8. `var bStruct = aStruct // aStruct and bStruct are two structs with`
`the same value!`
9. `bStruct.name = "Sue"`

10. `println(aStruct.name) // "Bob"`
11. `println(bStruct.name) // "Sue"`

View



- “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



(0,0)

X

- 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

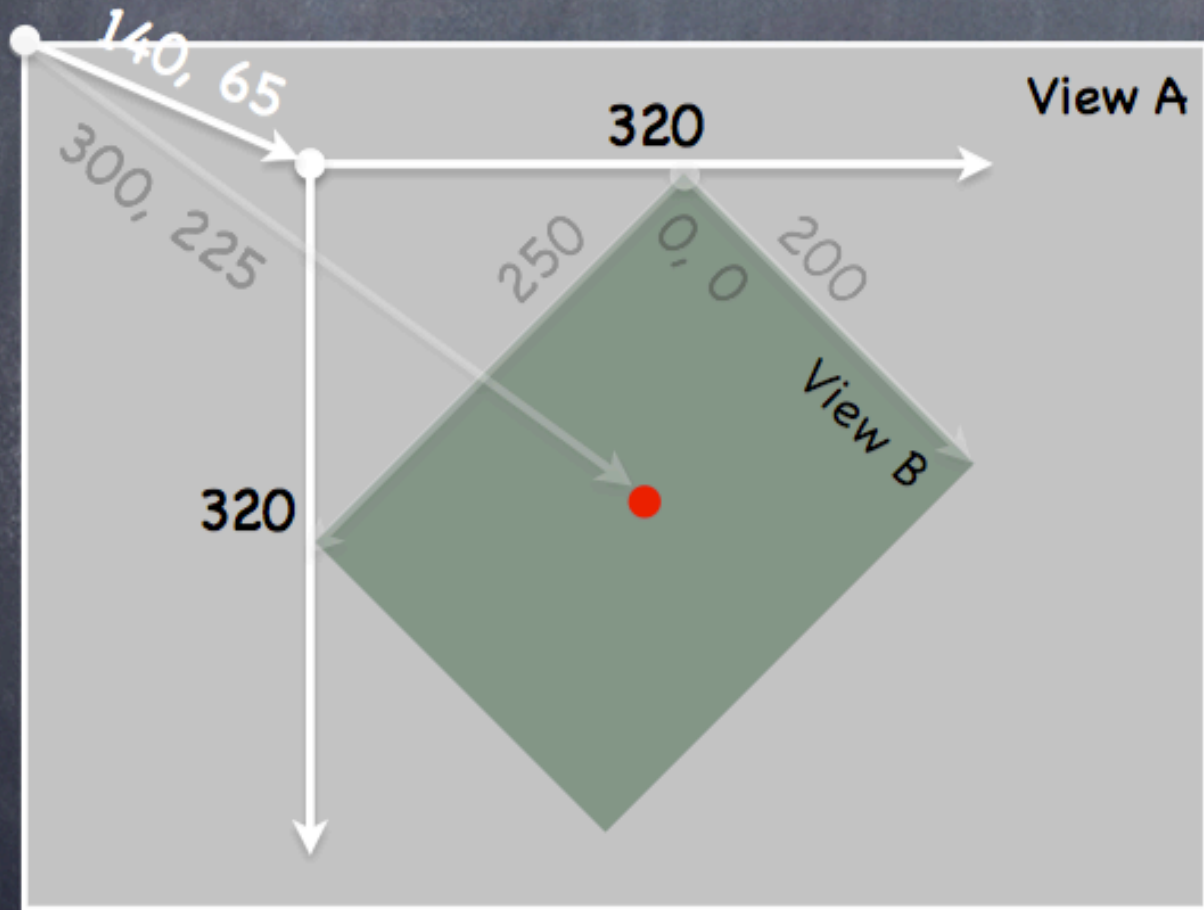
Y

View Coordinates

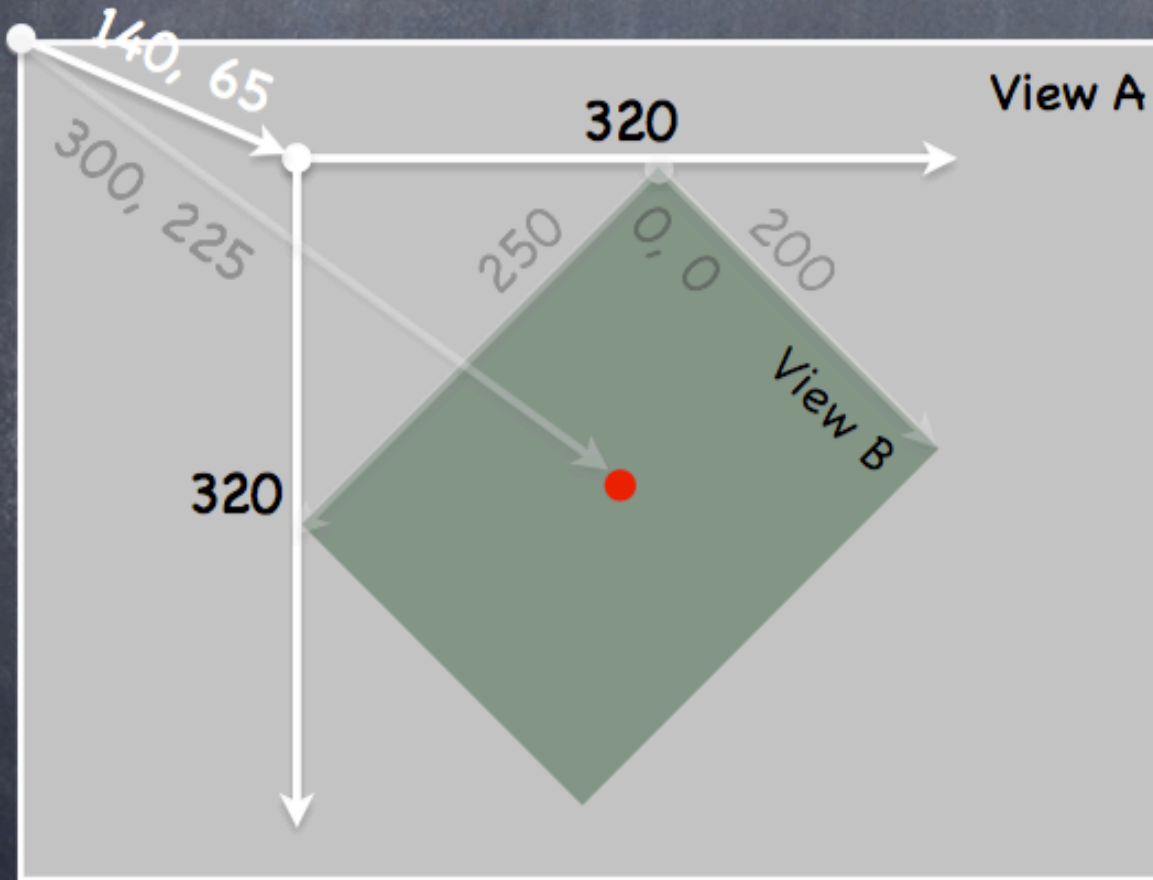


- **bounds**
 - Bir view'in kendisine göre koordinat durumu
- **frame**
 - Bir view'in superView'ine göre olan koordinat durumu
- **center**
 - Bir view'in superView'e göre merkez noktası

View Coordinates

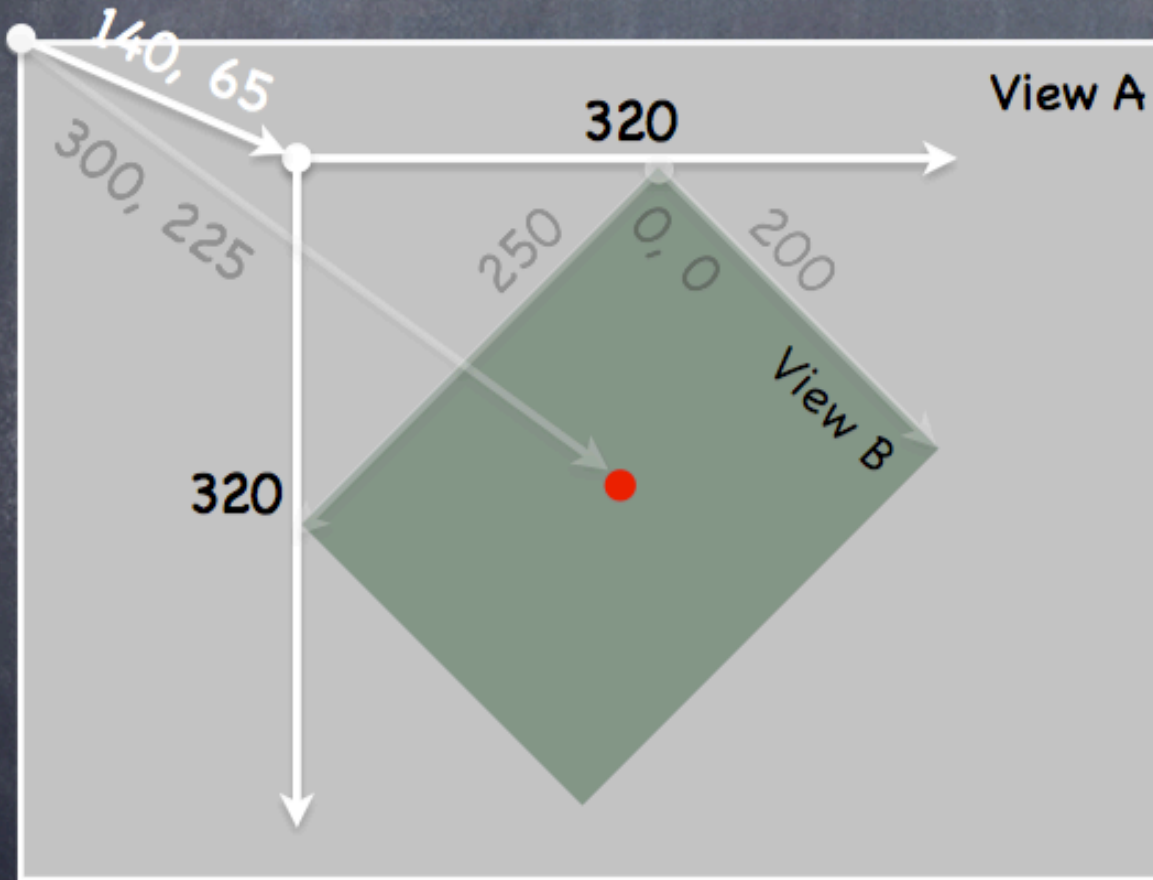


View Coordinates



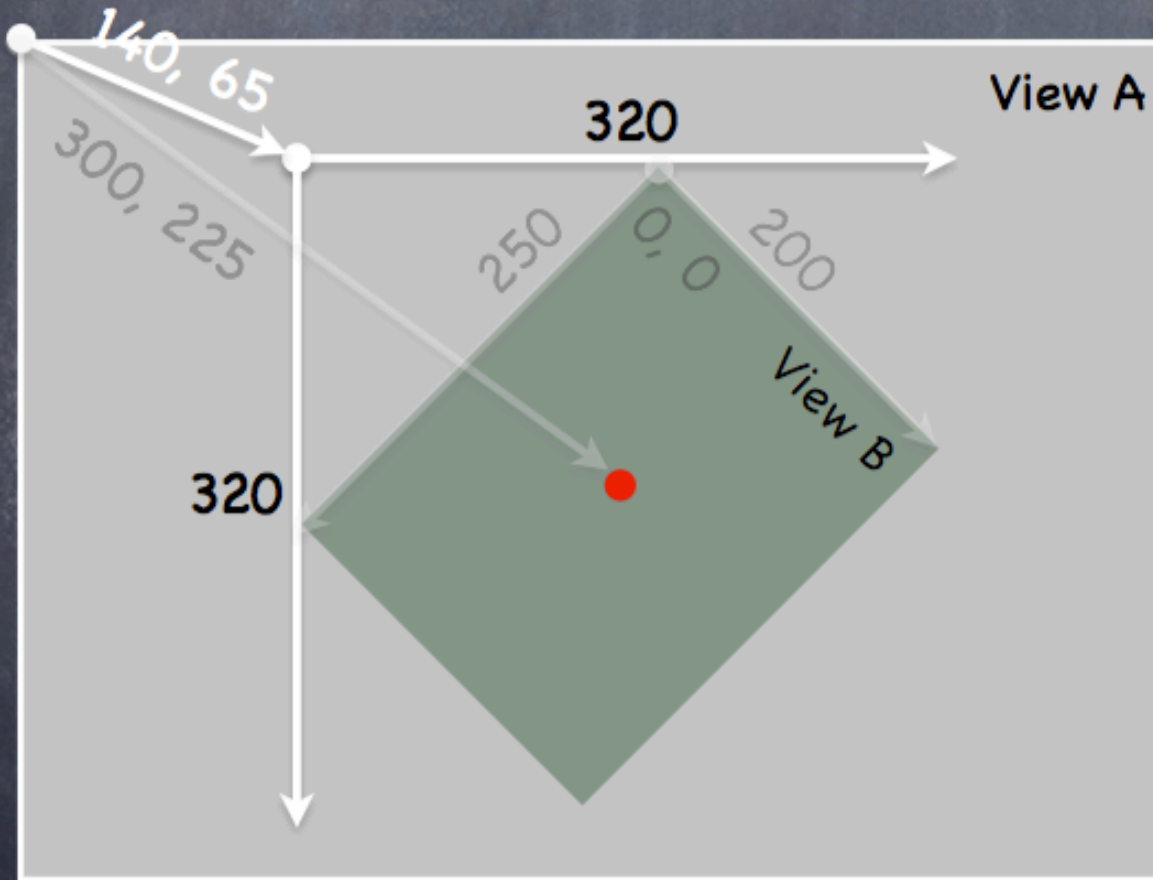
B'nin **bounds** değeri ?

View Coordinates



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

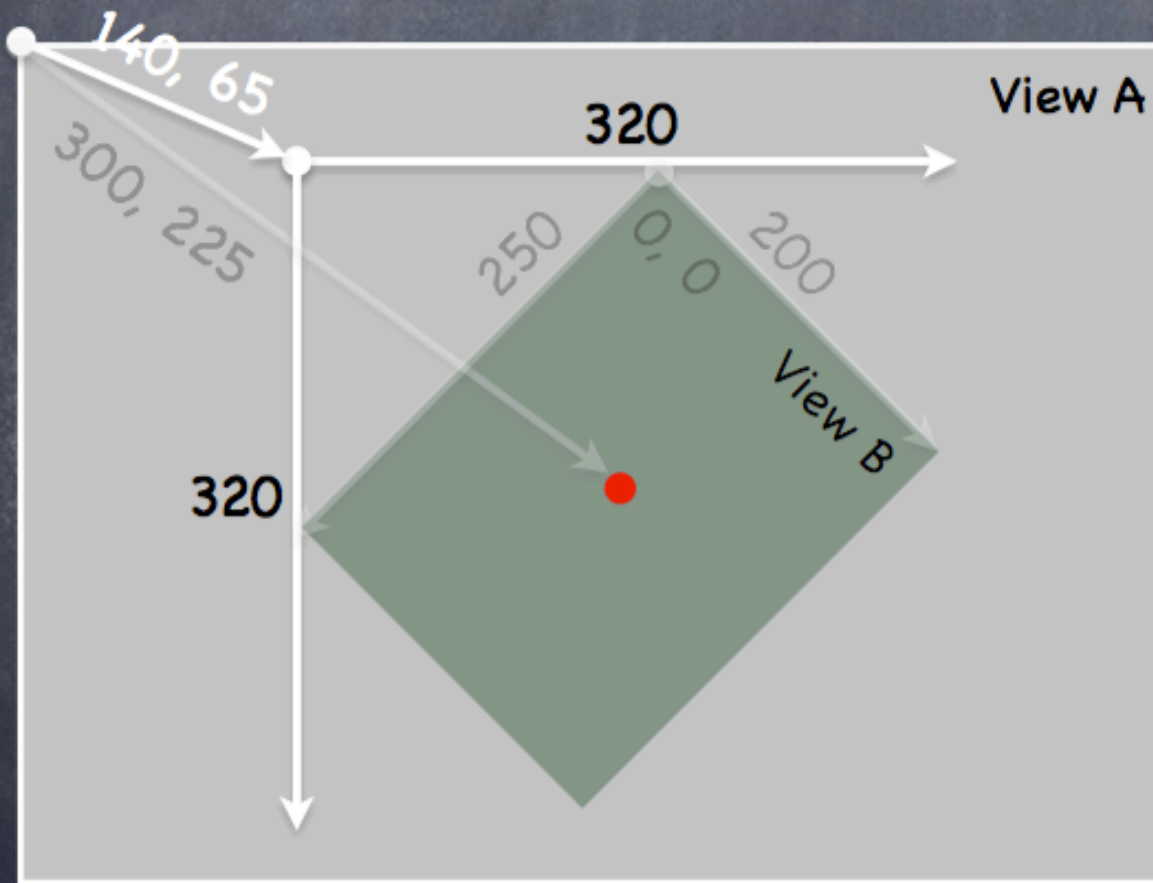
View Coordinates



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

B'nin **frame** değeri ?

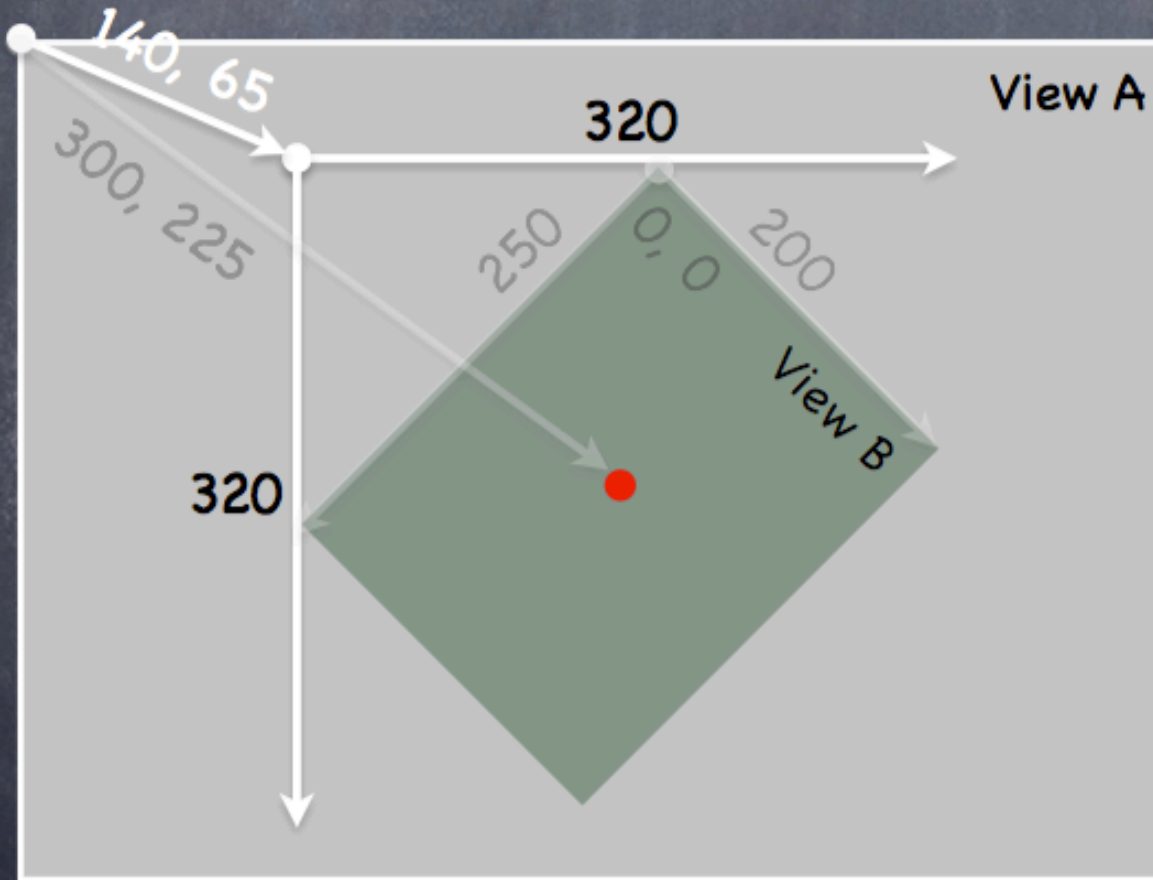
View Coordinates



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

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

View Coordinates

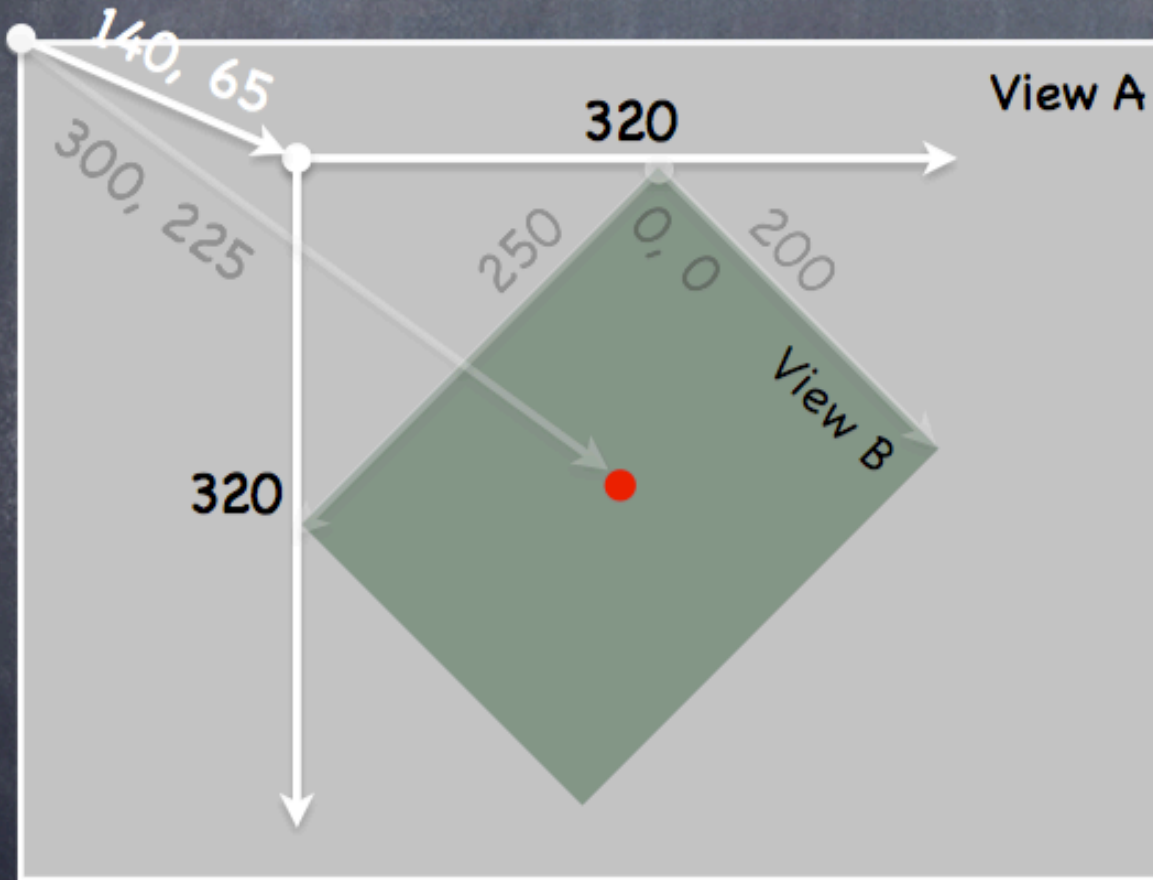


B'nin **bounds** değeri
((0,0) , (200,250))

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