

**PG5600**

**iOS programming**

**Forelesning 5**

# Sist gang

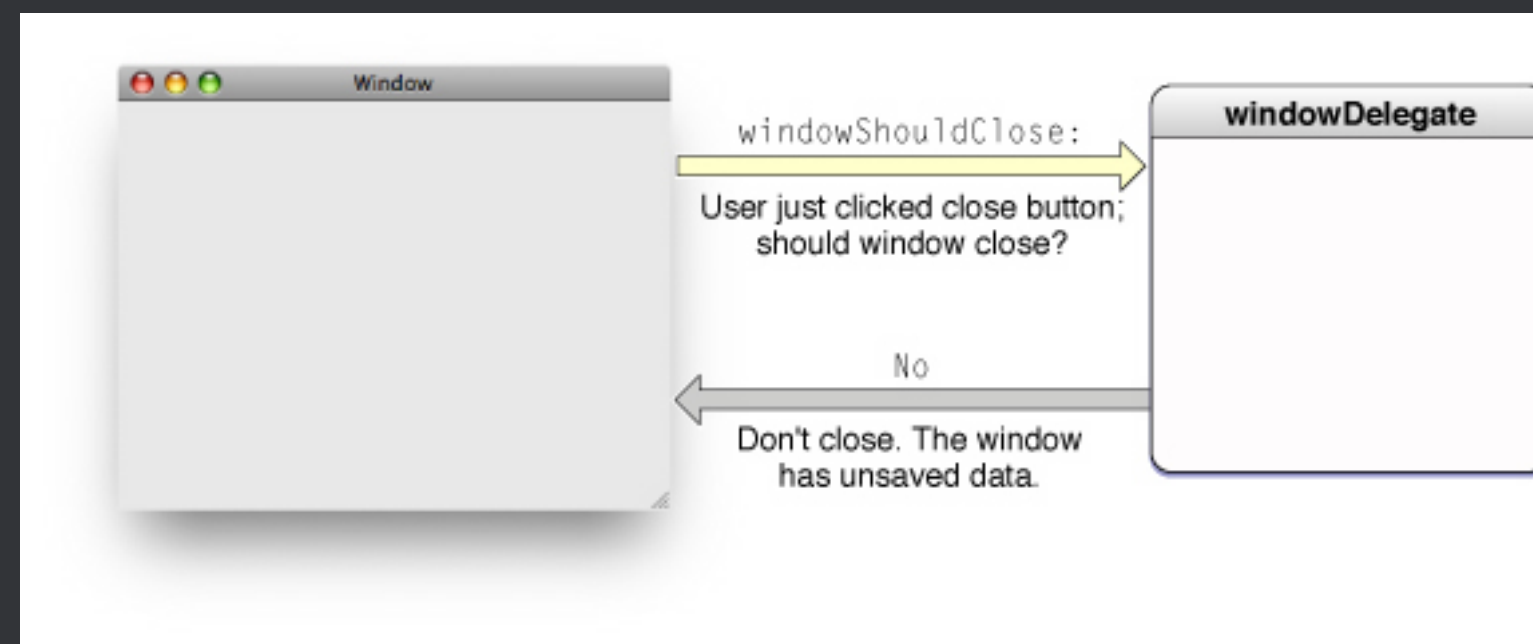
- Sette opp et nytt iOS prosjekt i XCode
- Komponentene i en iOS app
- Launch flow
- Application lifecycle
- MVC
- View og view controllers

# Agenda

- Delegate pattern
- UINavigationController
- UITableView og UITableViewController
- UICollectionView og UICollectionViewController
- Auto Layout
- Unified Storyboard og Size classes

# Delegate pattern

- Design pattern for å få et objekt til å delegere deler av sine oppgaver til et annet objekt
- Det delegerne objektet holder på en refferanse til et annet objekt kalt delegatet
- Det delegerende objektet kaller en eller flere metoder på delegatet
- Delegate patterne er mye brukt i Cocoa Touch



# Hva trengs?

# En protocol



```
protocol DataHandlerProtocol {  
    func didFinishGetting(data: NSData)  
    func didFailGettingData(error: NSError)  
}
```

**Et delegat som  
implementerer  
protocolen**

```
class A : DataHandlerProtocol {  
    func didFinishGetting(data: NSData) {  
        print(data)  
    }  
  
    func didFailGettingData(error: NSError) {  
        print(error)  
    }  
}
```

# **Et delegerende objekt**

```
class Fetcher {  
  
    var delegate: DataHandlerProtocol?  
  
    func fetch() {  
        delegate?.didFinishGetting(NSData())  
    }  
  
}
```

# Delegate-pattern Brukes blant annet av:

- UITableView
- UICollectionView
- UINavigationController

# UINavigationController

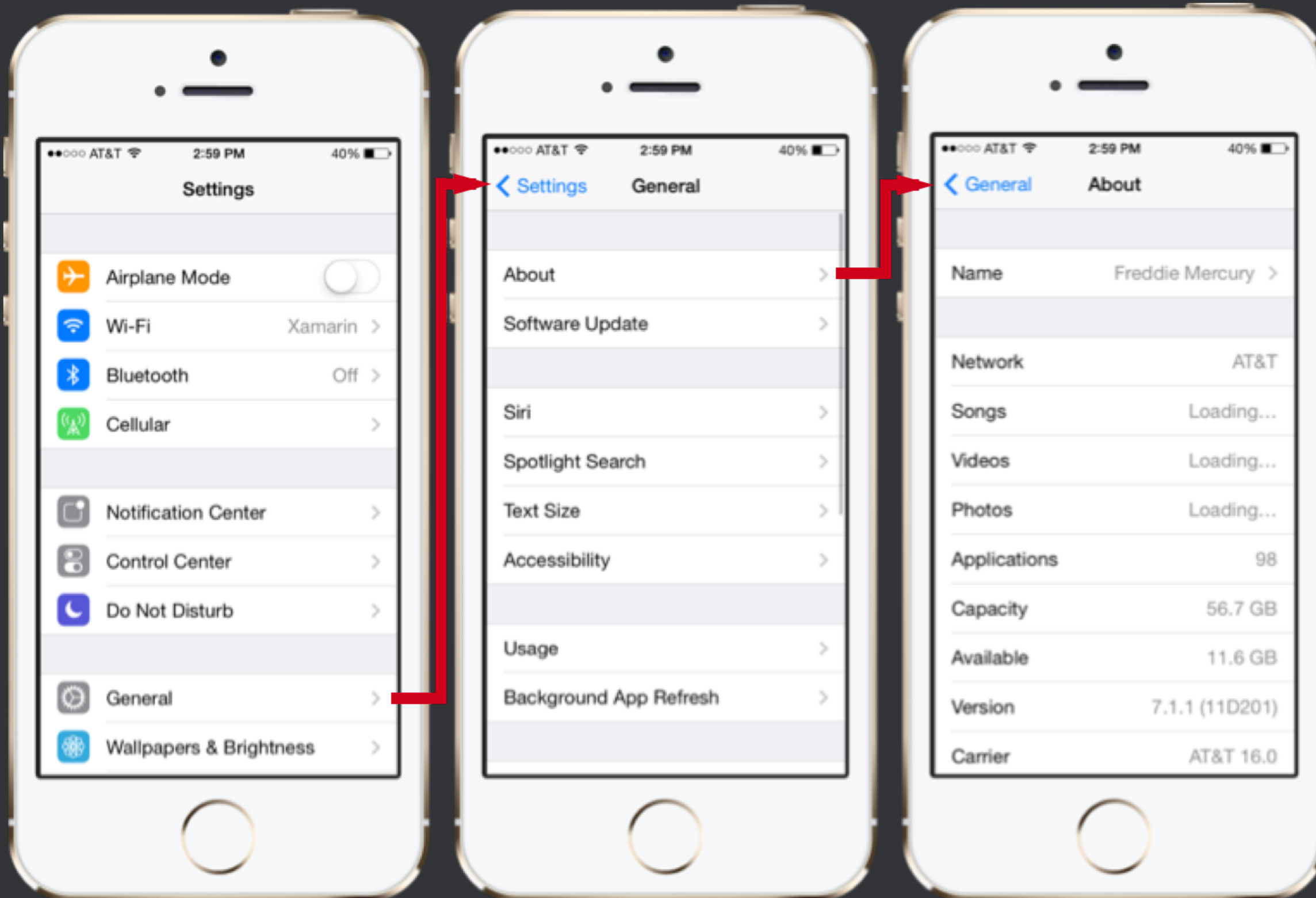
- En av standard komponentene
- Holder på en samling av UIViewController's
  - *En UIViewController holder på en samling av views*
- En UINavigationController kordinerer navigasjonen mellom UIViewController
- I tillegg til navigasjon har den også ansvar for en UINavigationController og en UIToolbar





UIViewController er håndtert av UINavigationController ved å:

- Pushe og poppe UIViewController på UINavigationController view stacken
- Når et view blir pushet, så blir nåværende view dyttet ut av skjermen til venstre og det nye kommer inn fra høyre
- Om du bruker storyboard så blir alt dette automatisk håndtert
- Hvis du bruker XIB filer eller bare kode så må du selv pushe og poppe



# Kan jeg gjøre det i kode?

```
let rootViewController = UIViewController()  
var navigationController = UINavigationController(rootViewController: rootViewController)  
presentViewController(navigationController animated:true, completion: nil)
```

# Push og pop

```
var viewController = UINavigationController()  
navController.pushViewController(viewController, animated: true)  
navController.popViewControllerAnimated(true)
```

# UIButton og UINavigationController

```
class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        var rightButton = UIBarButtonItem(
            barButtonItemSystemItem: UIBarButtonItemSystemItem.Add,
            target: self,
            action: Selector("buttonTouched:"))

        self.navigationItem.rightBarButtonItem = rightButton;
    }

    func buttonTouched(sender: AnyObject) {
        self.navigationController?.pushViewController(ViewController(), animated: true)
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
    }
}
```

# UILabel/Tittel og UINavigationController

```
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        self.navigationItem.title = "Test tittel"  
  
    }  
}
```

**UITableView**

**&**

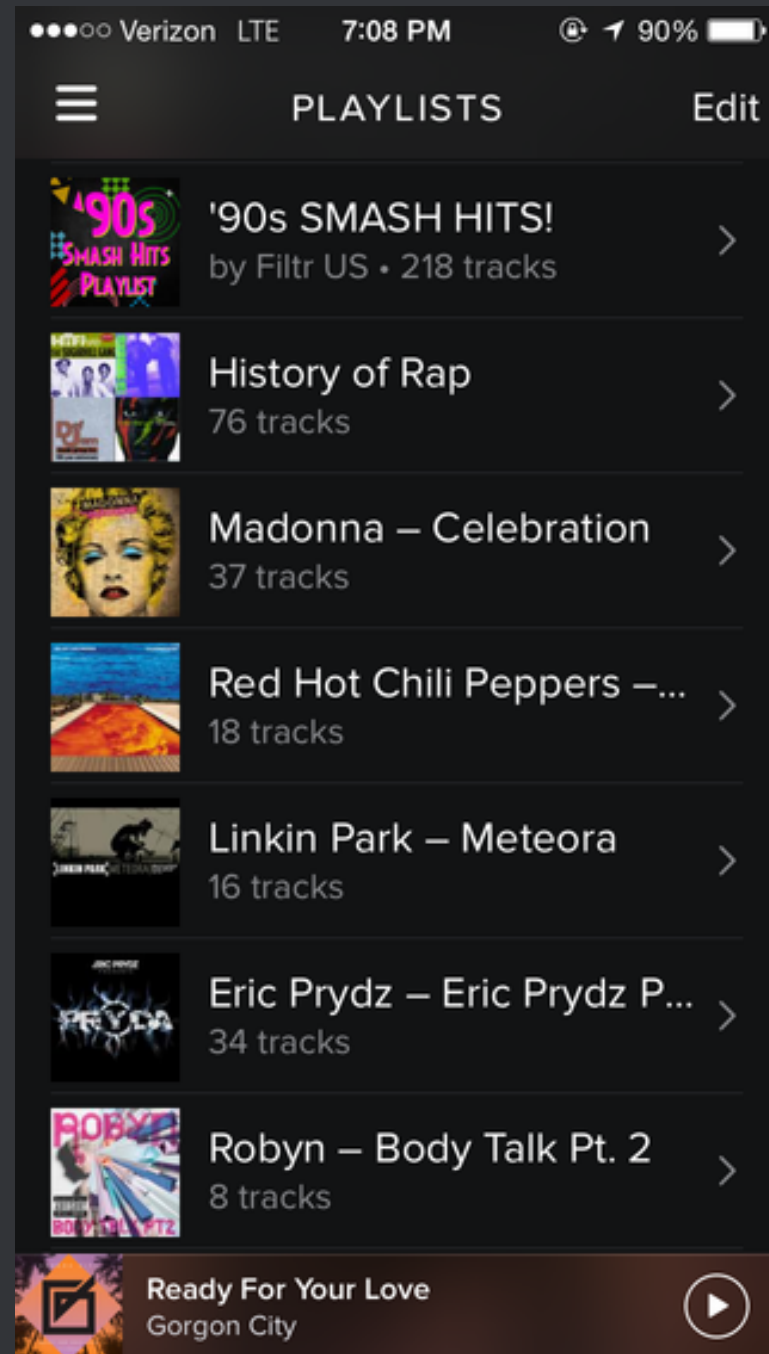
**UITableViewController**

# Noen bruksområder


- Navigere hierarkisk data
- Presentere en liste av elementer
- Vise detaljinformasjon og kontroller i visuelle grupperinger
- Vise valgbare alternativer



# Spotify



# Peek

February2014

MON	TUE	WED	THU	FRI	SAT	SUN	
27	28	29	30	31	01	02	

12:00 PM

Check-out: St. Ermins Hotel,...

01:00 PM

Check-in: Hotel Bergs

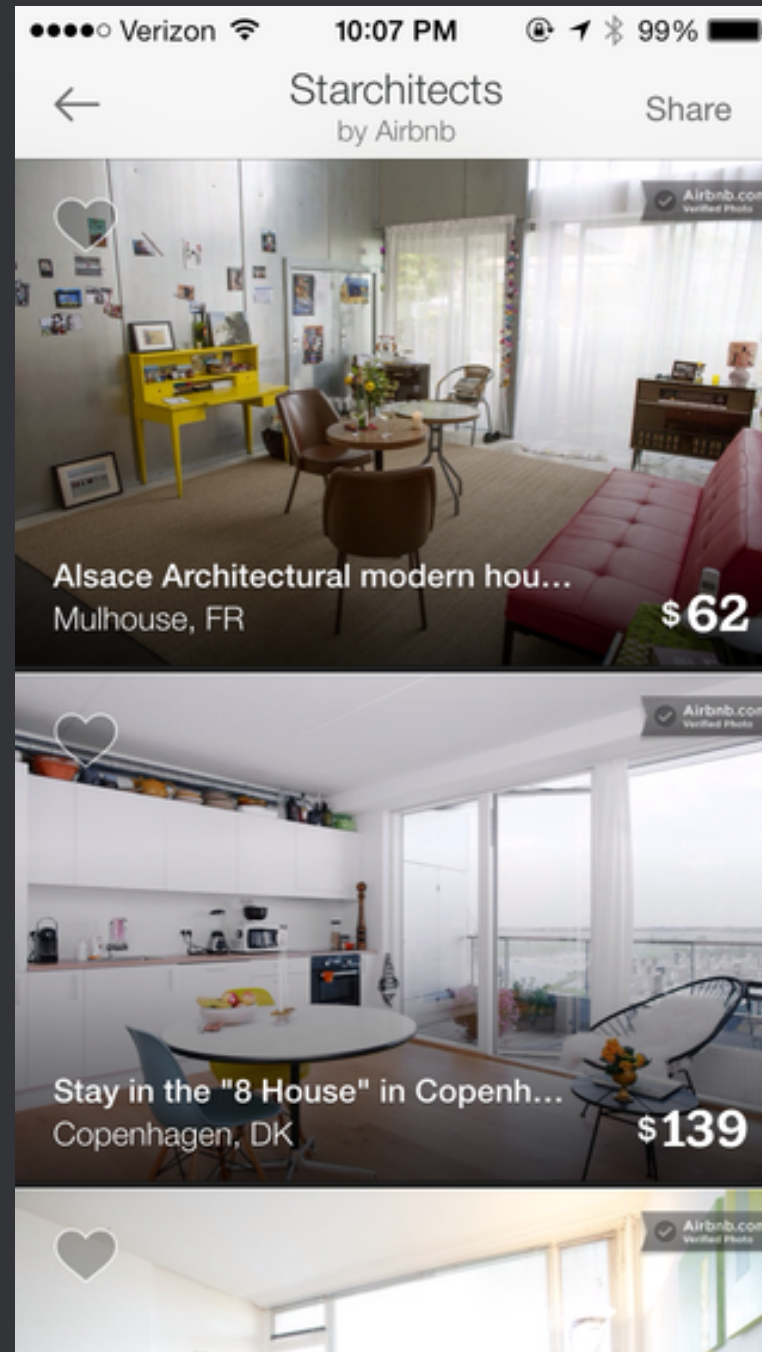
04:50 PM

BT654 LGW to RIX

07:45 PM

Directions from RIX to Hotel...

# Airbnb



# Egenskaper og innhold

- Hver rad består av en UITableViewCell
- Kommer med noen pre-definerte design (kan lage egne)
- Kan respondere på at man klikker på en celle
- Har implementert editeringsmuligheter:
  - Legg til
  - Fjern
  - Omorganisering



California
Brea
Burlingame
Canoga Park
Carlsbad
Chula Vista
Corte Madera
Costa Mesa
Emeryville
Escondido
Fresno
Glendale
Section Footer



**View Controller** - A controller that supports the fundamental view-management model in iPhone OS.

inherits from



**Table View Controller** - A controller that manages a table view.

manages & contains



**Table View** - Displays data in a list of plain, sectioned, or grouped rows.

manages & contains



**Table View Cell** - Defines the attributes and behavior of cells (rows) in a table view.



**UITableViewDelegate** - Implementing a delegate allows your class to respond to events occurring in another object. For example, in this case you can respond to a row selection in a table view.

implements



**UITableViewDataSource** - A datasource is very much like a delegate. However, it is delegated control of data. A table view is driven by an underlying data model and the datasource manages that relationship.

# Demo

**Nytt prosjekt -> Master/detail-application**

# Men hvordan sette data før man navigerer? Fra sist!

```
class ViewController: UIViewController {  
  
    @IBOutlet weak var message: UITextField!  
  
    // Kalles før overgangen skjer også med navigation controller  
    override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject!) {  
  
        if let vc = segue.destinationViewController as? ReceivingViewController {  
            vc.message = message.text  
        }  
  
    }  
  
}
```

# UICollectionView



- Er et friere alternativ til UITableView
- Endringer gjøres ved hjelp av subclassing
- Du kan implementere grids-, stacks-, sirkulære layout
- eller hva som helst annet du kan se for deg

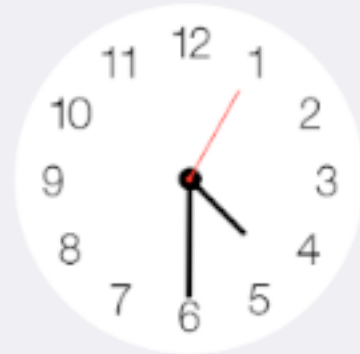
Edit

## World Clock



Cupertino

Today



New York

Today, 3 hours ahead



Paris

Today, 9 hours ahead



Beijing

Tomorrow, 15 hours ahead



Tokyo

Tomorrow, 16 hours ahead



Moscow

Tomorrow, 11 hours ahead



Cell

Paris

Chamonix

Decoration view

Supplementary view

**UICollectionViewContro  
ller består av:**

# UICollectionView

**Hoved viewet som inneholder alle de andre viewene. Dette er veldig likt UITableView**

# UICollectionViewCell

**Lik UITableViewCell i UITableView. Celler kan lages programatisk eller via interface builder**

# Supplementary Views

**Du kan legge til flere views som ikke er celler**

**Dette er typisk view i toppen og bunnen**

# Decoration View

**views som ikke legger til noe funksjonalitet, men har til hensikt å gjøre collection view visuelt penere**



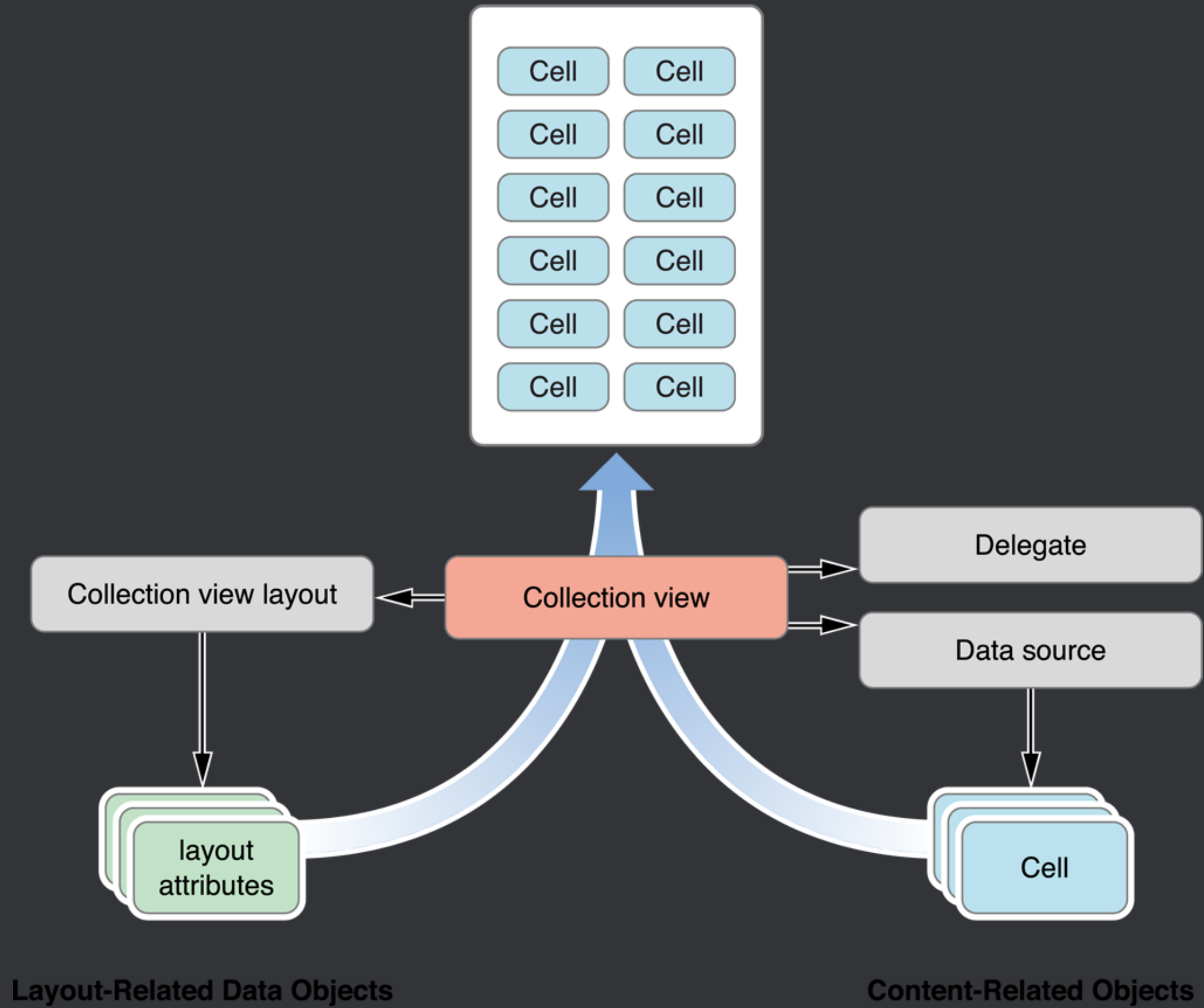
# UICollectionViewLayout

**Bestemmer hvordan layout av cellene skal være**

**Layout kan byttes ut runtime**

# **UICollectionViewFlowLayout** **Layout**

**Apple har laget et eget layout som følger med**

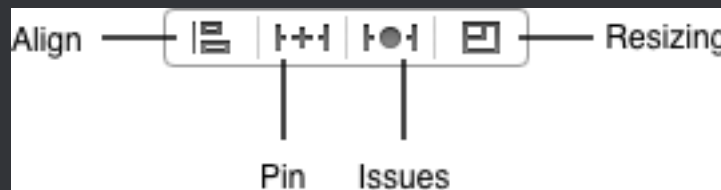


# Auto Layout

- Legger ut ditt brukergrensesnitt basert på et sett med regler
- Du definerer opp reglene ved hjelp av Auto Layout
- Målet er å lage dynamisk grensesnitt som responderer på endringer i størrelse, orientering og lokalisering (språk)
- Auto Layout er bygd inn i Interface Builder

- **Kontant verdi** - Den fysiske størrelsen eller offsettet in antall punkt for regelen
- **Relasjon** - Bruk relasjoner og større enn, mindre enn eller lik for å spesifisere for eksempel at et views bredde  $\geq 20$ , eller at et `textView.leading`  $\geq$  (`superview.leading` + 20).
- **Prioritetsgrad** - Regler kan ha prioritet, slik at noen kan telle mindre enn andre. Default er prioriteten satt til maks (påkrevd) og da vil den bli håndhevd

# Interface builder, point and click



# Programatisk

- Visual Format Language

```
let views = ["button1": UIButton(), "button2": UIButton()]
var constraints = NSLayoutConstraint.constraintsWithVisualFormat(
    "[button1]-[button2]",
    options: NSLayoutConstraintOptions.AlignAllBaseline,
    metrics: nil,
    views: views
)
```



Ikke alt kan lages ved hjelp av visual format, for eksempel:

```
imageView.width = 2 * imageView.height
```

```
// Vil gjøre det samme som eksempelet med visual format
```

```
NSLayoutConstraint(  
    item: self.button1,  
    attribute: NSLayoutConstraint.Right,  
    relatedBy: NSLayoutConstraint.Equal,  
    toItem: self.button2,  
    attribute: NSLayoutConstraint.Left,  
    multiplier: 1.0,  
    constant: -12.0  
)
```

# Finnes også alternativer

- <https://github.com/robb/Cartography>
- <https://github.com/PureLayout/PureLayout>
- <https://github.com/SnapKit/Masonry>

Anbefaler å gjøre mest i storyboard

(ikke pensum)

# Aktivere og deaktivere på NSLayoutConstraint

```
var active: Bool
```

```
class func activateConstraints(constraints: [AnyObject])
```

```
class func deactivateConstraints(constraints: [AnyObject])
```

# Potensielle problemer

- Manglende regler
- Viewet i interface builder er plassert et annet sted en reglene vil plassere det
- Konflikter
- Tvetydighet - når det er flere løsninger for et view basert på reglene
- Manglende regler

# Demo

# Constraint og Interface Builder

CTRL dra og pek

**View debugging, trykk  
pause**

# Unified Storyboard & Size classes

**Hva er det til?**

**For å gjøre det enklere  
å støtte flere  
skjermstørrelser**

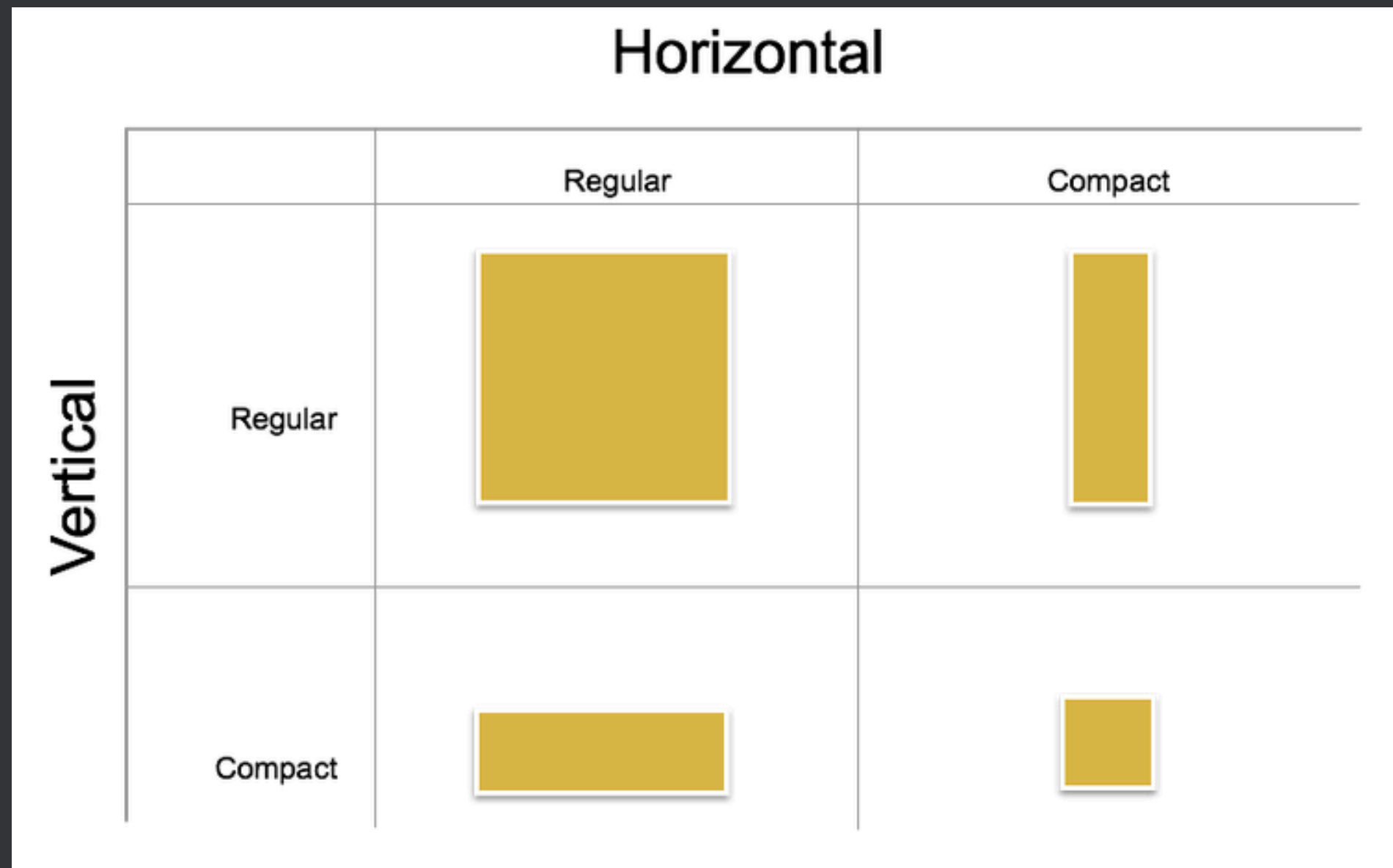


**Hvorfor?**

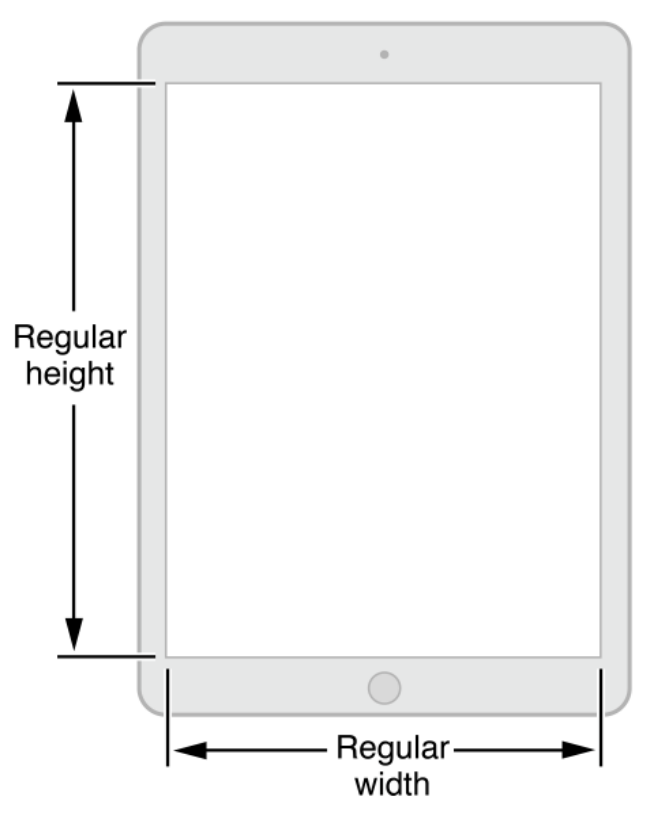
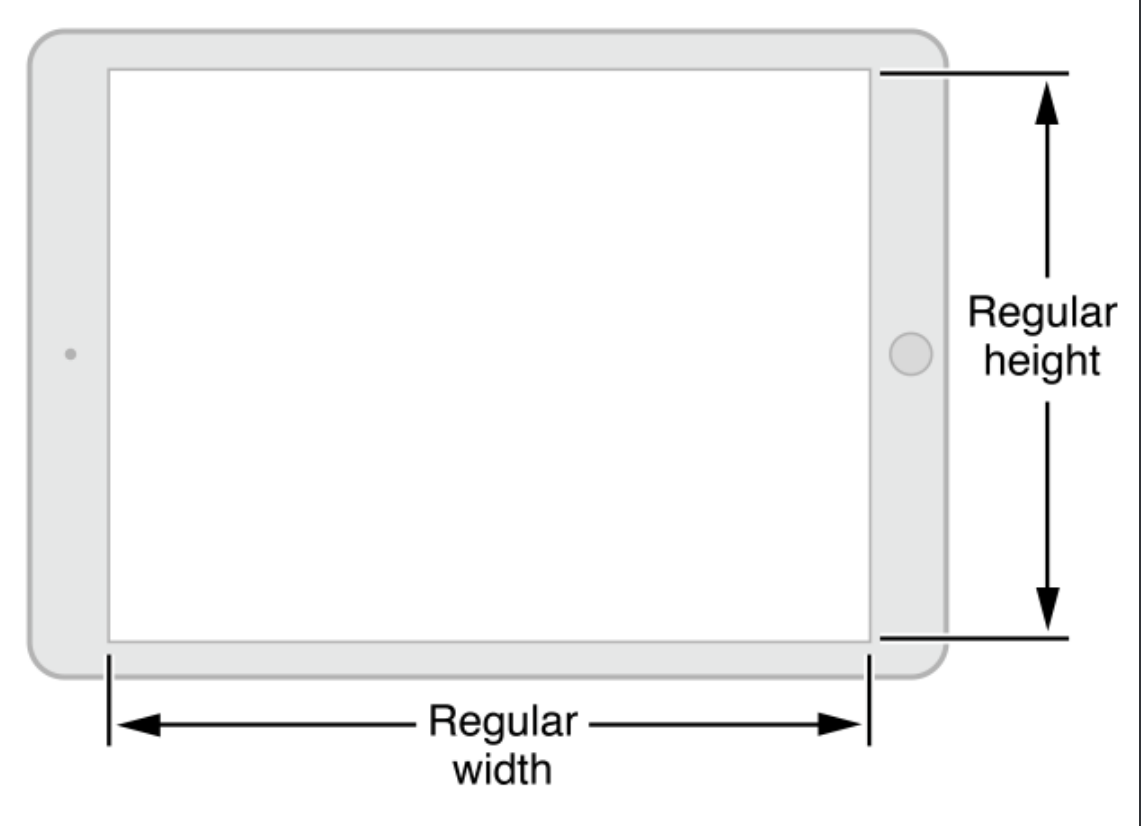
**Apps bør tilpasse seg endring i  
orientering og skjermstørrelse for å  
gi en bedre brukeropplevelse**

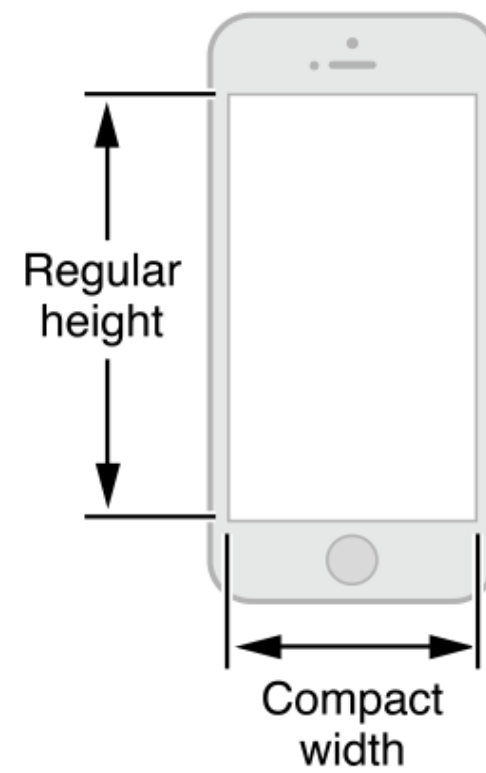
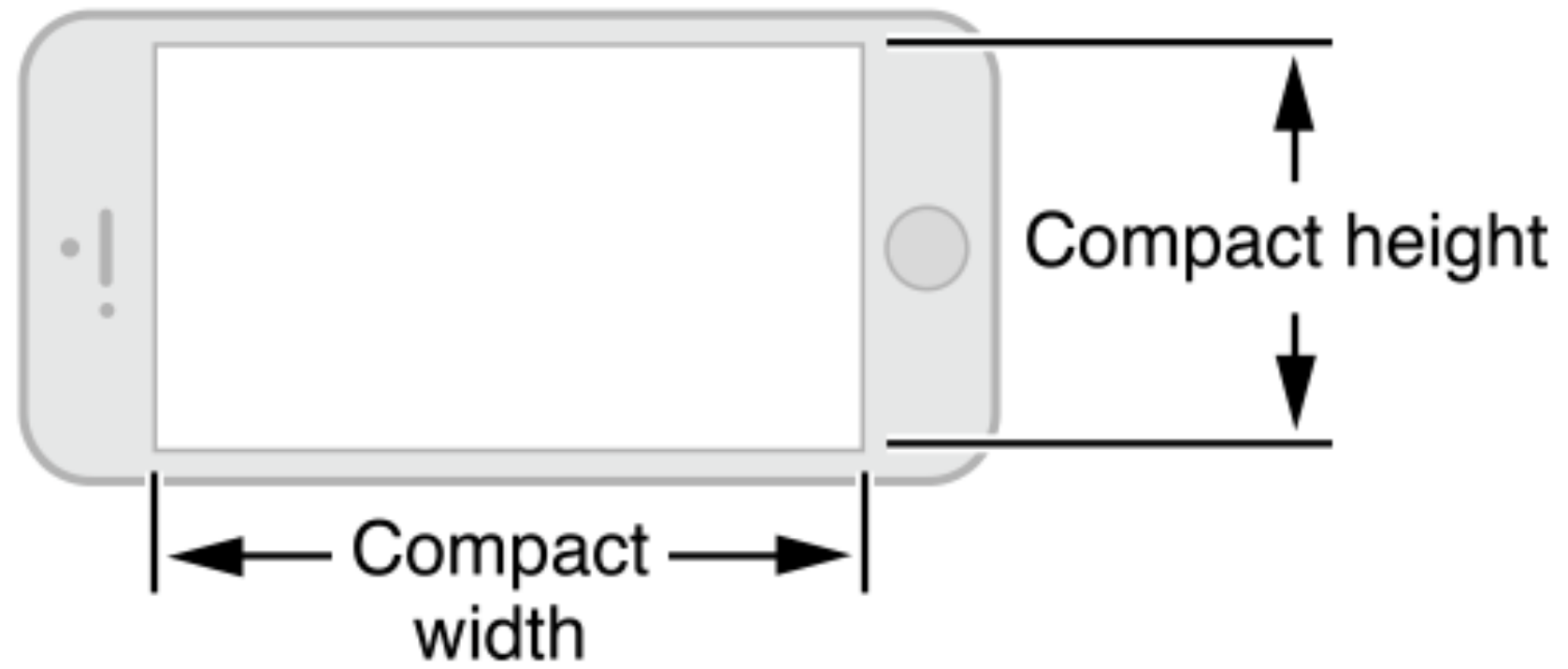
# Size classes

- Regular - Representerer mye skjermplass
- Compact - Representerer lite skjermplass

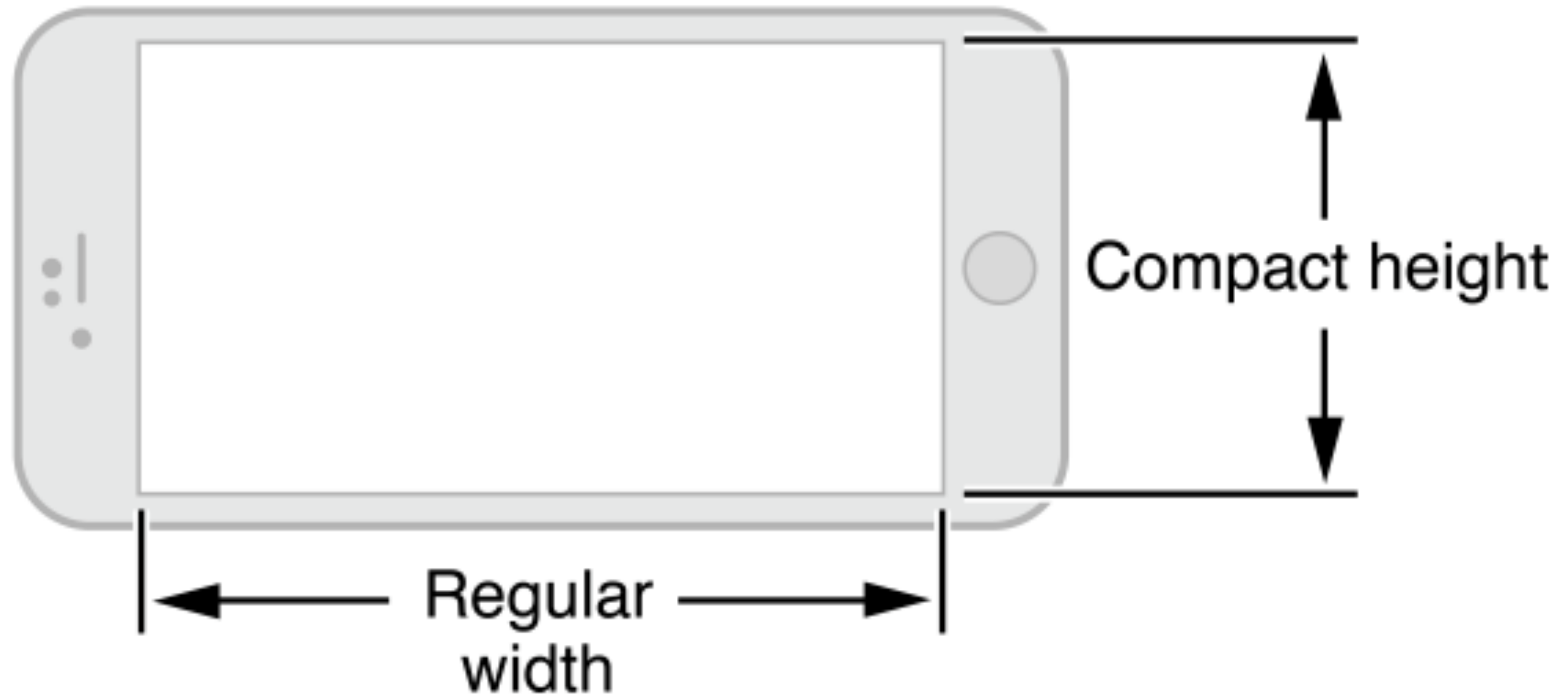


[http://developer.xamarin.com/guides/ios/platformfeatures/  
intro to unified Storyboards/](http://developer.xamarin.com/guides/ios/platformfeatures/intro%20to%20unified%20Storyboards/)





# iPhone 6+



# Traits

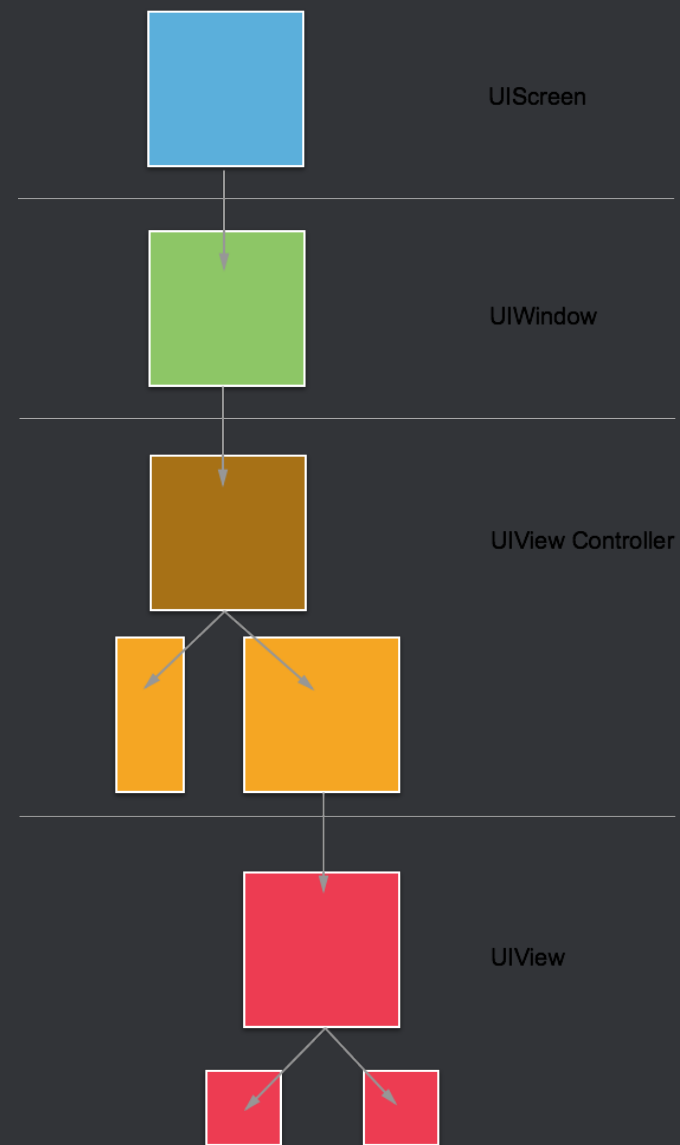
- Et system for å håndtere size classes
- Består av et sett med properties som kan brukes for å definere opp hvordan ditt layout skal endre seg ettersom miljøet endrer seg

```
class UITraitCollection : NSObject, NSCopying, NSSecureCoding, NSCoding {  
    ...  
  
    var userInterfaceIdiom: UIUserInterfaceIdiom { get }  
    var displayScale: CGFloat { get }  
    var horizontalSizeClass: UIUserInterfaceSizeClass { get }  
    var verticalSizeClass: UIUserInterfaceSizeClass { get }  
    ...  
}
```

# Tilgjengelig på følgende objekter igjennom **UITraitEnvironment**

- Screens ( UIScreen ).
- Windows ( UIWindow ).
- View Controllers ( UIViewController )
- Views ( UIView )
- Presentation Controller ( UIPresentationController )





[http://developer.xamarin.com/guides/ios/platformfeatures/  
intro to unified Storyboards/](http://developer.xamarin.com/guides/ios/platformfeatures/intro%20to%20unified%20Storyboards/)

```
func traitCollectionDidChange(previousTraitCollection: UITraitCollection)
```

# Videre lesning

- Kapittel 6 i iOS 8 Swift Programming Cookbook
- View Controller programming guide - Apple docs
  - The adaptive model

# Oppgaver

## Se Oppgaver på It's learning

**Forelesningen er basert på fjorårets foiler, laget av**

**Hans Magnus Inderberg og Mads Mobæk**