

Beta 2.0

# Full Stack iOS Entwicklung mit Swift

WPF im MIM - WS 17/18  
Alexander Dobrynin, M.Sc.

# Heute

Layout und Views

Autolayout

StackView

Animationen

Size Classes

Demo

Assignment

# Layout und Views

- Jedes Layout besteht aus einer View Hierarchie
- Jede View hat eine **SuperView** und 0 bis n **SubViews**
- Ursprung (0,0) des Koordinatensystems ist oben Links
- Einheit sind **Points**, eine Abstraktion über Pixel
- Wichtige Datentypen im Koordinatensystem sind CGFloat, CGPoint (x, y), CGSize (w, h) und CGRect (point, size)
  - `var bounds: CGRect` // Eigenes Koordinatensystem
  - `var frame: CGRect` // SuperView's Koordinatensystem
- Für Custom-Views eine gleichnamige .xib- und .swift Datei erstellen
  - **File's Owner** der .xib auf die gleichnamige .swift Klasse setzen
  - .swift Klasse von **CustomView\*** erben lassen
- **Document Outline** und **Size Inspector** sind die wichtigsten Tools im Storyboard



iPhone X



iPhone 6+, 6s+, 7+, 8+



iPhone 6, 6s, 7, 8



iPhone 5, 5s, 5c, SE



iPhone 4, 4s

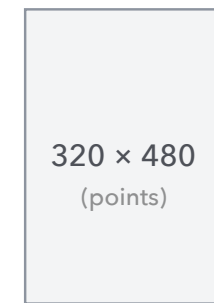
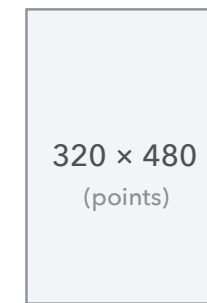
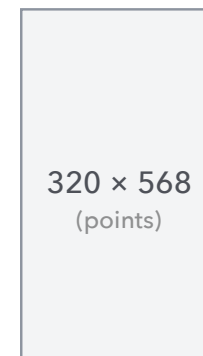
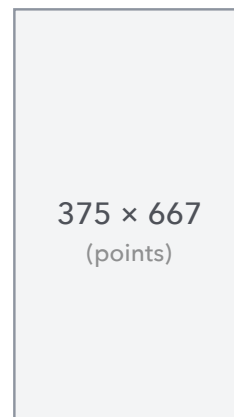
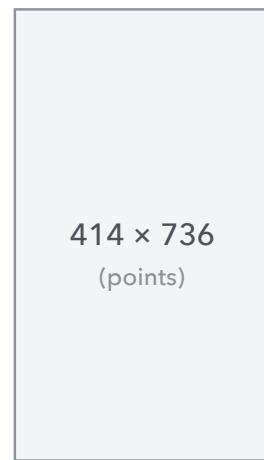
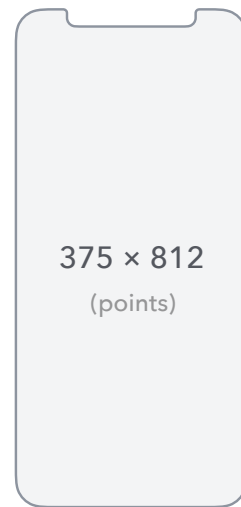


iPhone 2G, 3G, 3GS

### Points

At the beginning, coordinates of all drawings are specified in *points*.

Points are abstract units, they only make sense in this mathematical coordinate space.



render at 3x

render at 3x

render at 2x

render at 2x

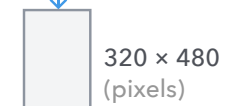
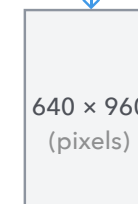
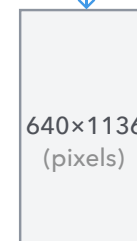
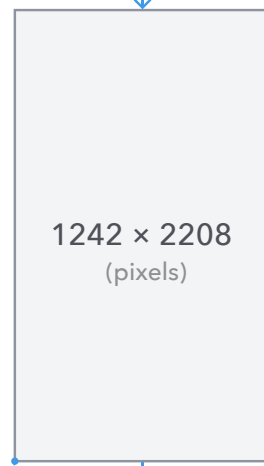
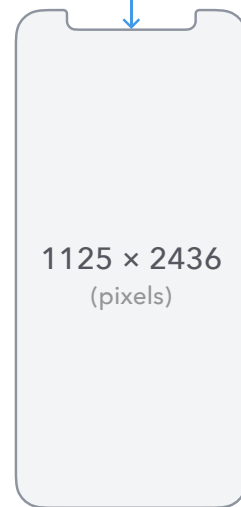
render at 2x

render at 1x

### Rendered Pixels

Point-based drawings are rendered into pixels. This process is known as rasterization.

Point coordinates are multiplied by scale factor to get pixel coordinates. Higher scale factors result in higher level of detail.

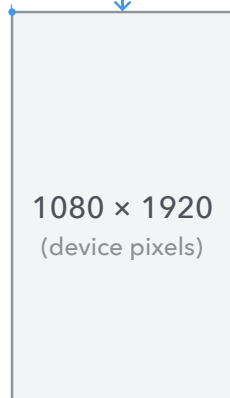


downsampling / 1.15

### Physical Pixels

The device screen may have lower pixel resolution than the image rendered in previous step.

Before the image can be displayed, it must be downsampled (resized) to lower pixel resolution.



# Heute

Layout und Views

Autolayout

StackView

Animationen

Size Classes


Demo

Assignment

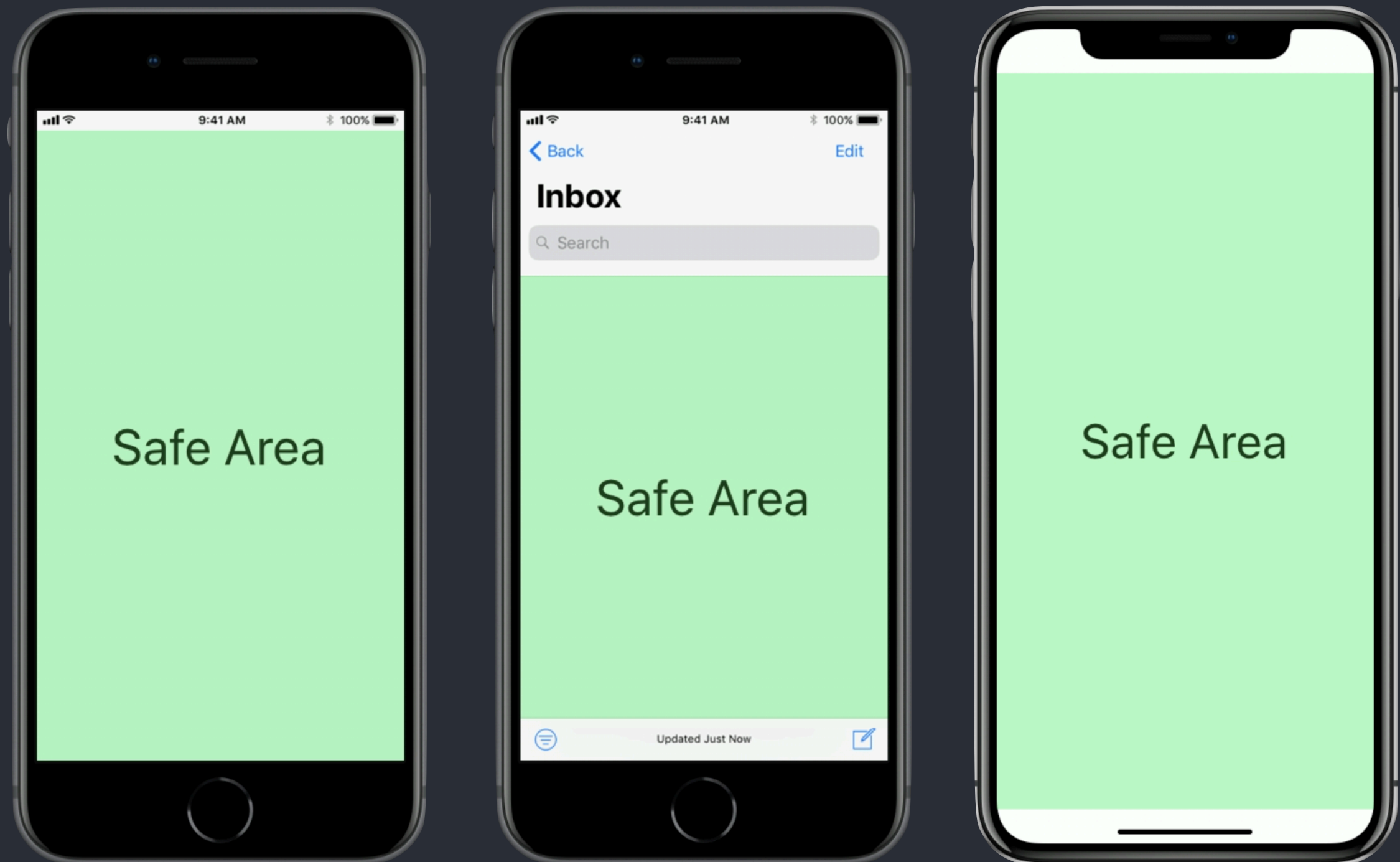
# Motivation

Warum Autolayout?

# Autolayout

- “*Universal*” Apps mittels automatischer Adaption von Layout
- “*Design (almost) once, ship anywhere*”
- Autolayout findet innerhalb der Safe Area statt 
- Autolayout und Custom-Controls sind die Basis vom UI-Design
- Automatische Adaption von
  - Views für unterschiedliche Displaygrößen und Auflösungen
  - Rotationen (Landscape und Portrait)
  - Text-Styles (Fonts) für Barrierefreiheit
  - Designs für “Links zu Rechts” und “Rechts zu Links” Sprachen
- Advanced: Size Classes (Compact und Regular), um displayspezifische Anpassungen vorzunehmen

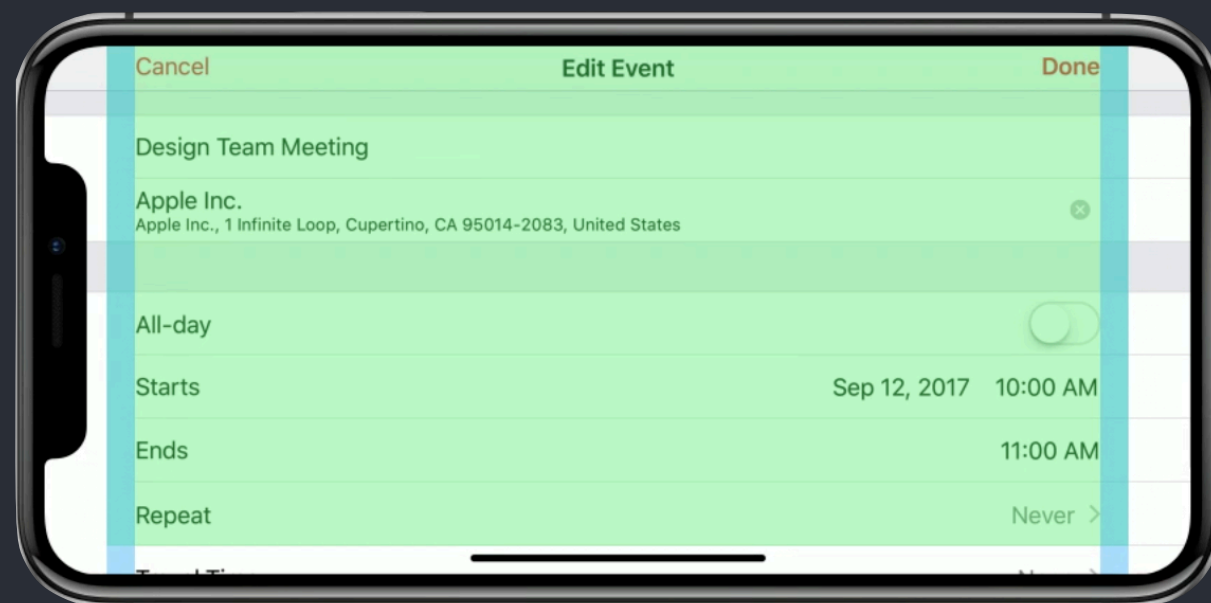
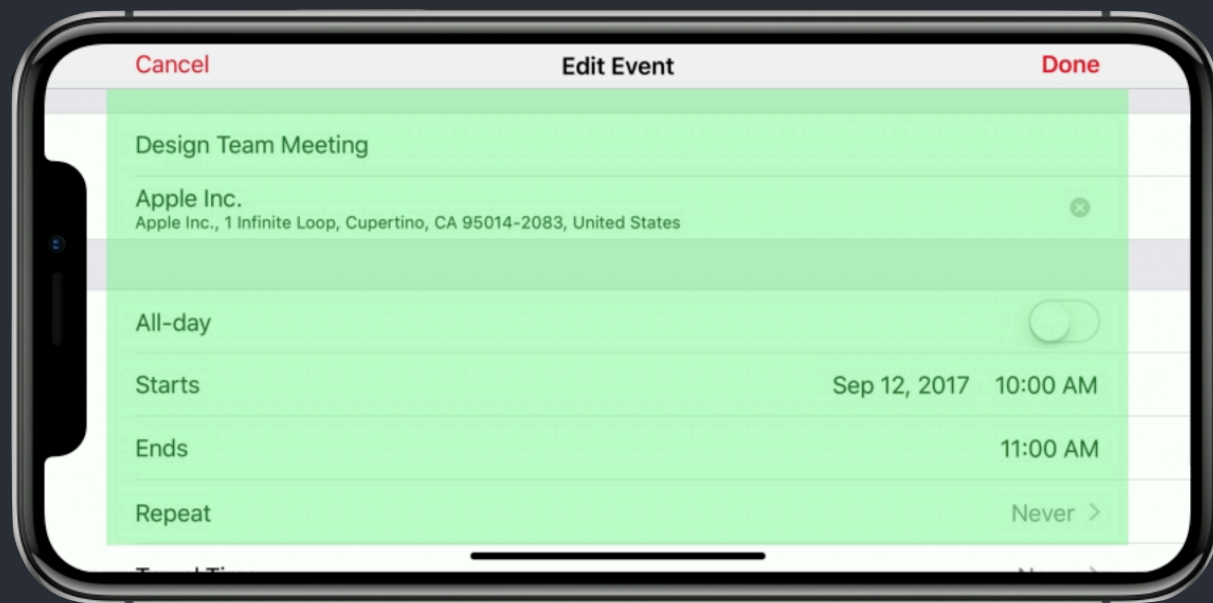
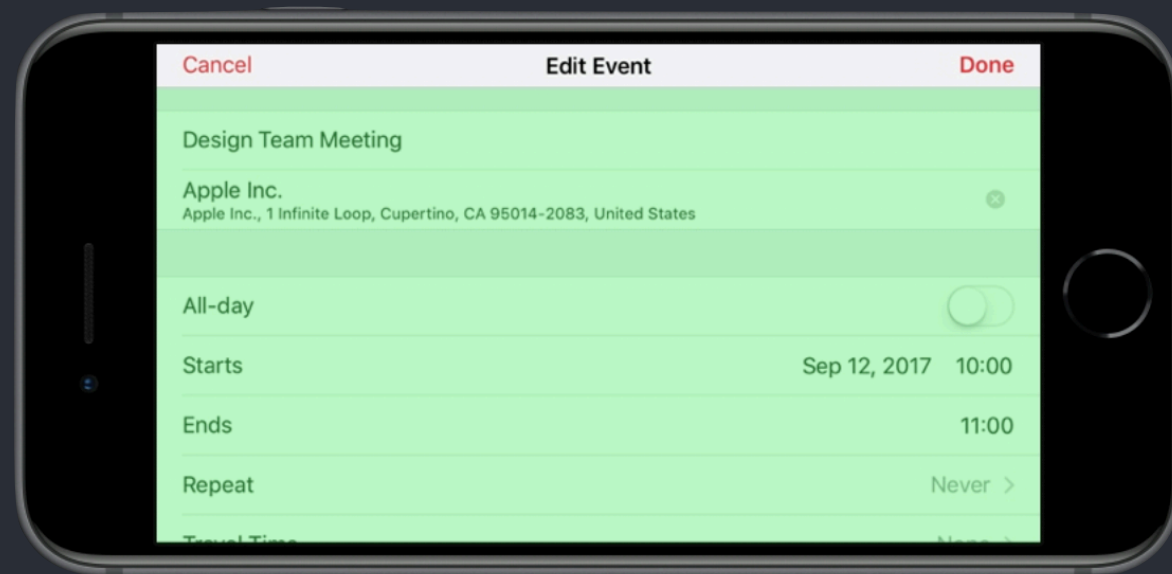
# Autolayout



<https://developer.apple.com/videos/play/fall2017/801/>



# Autolayout

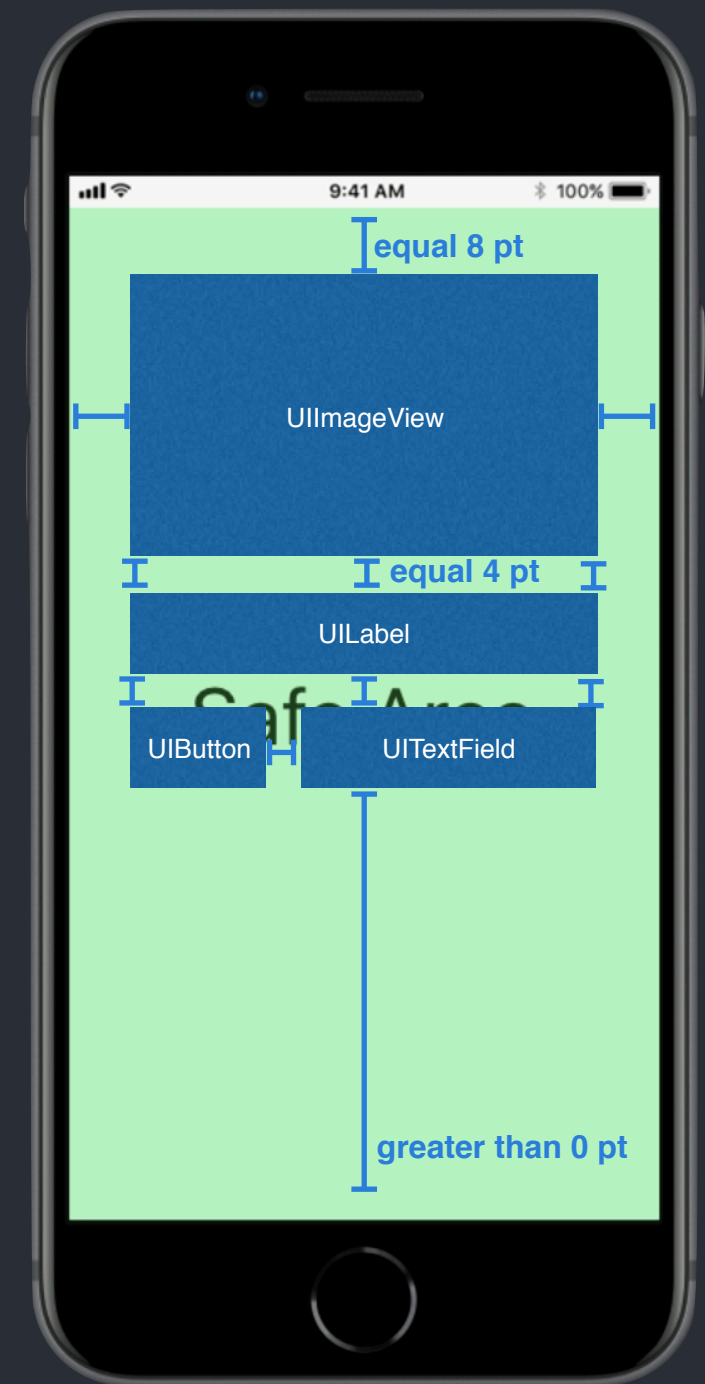
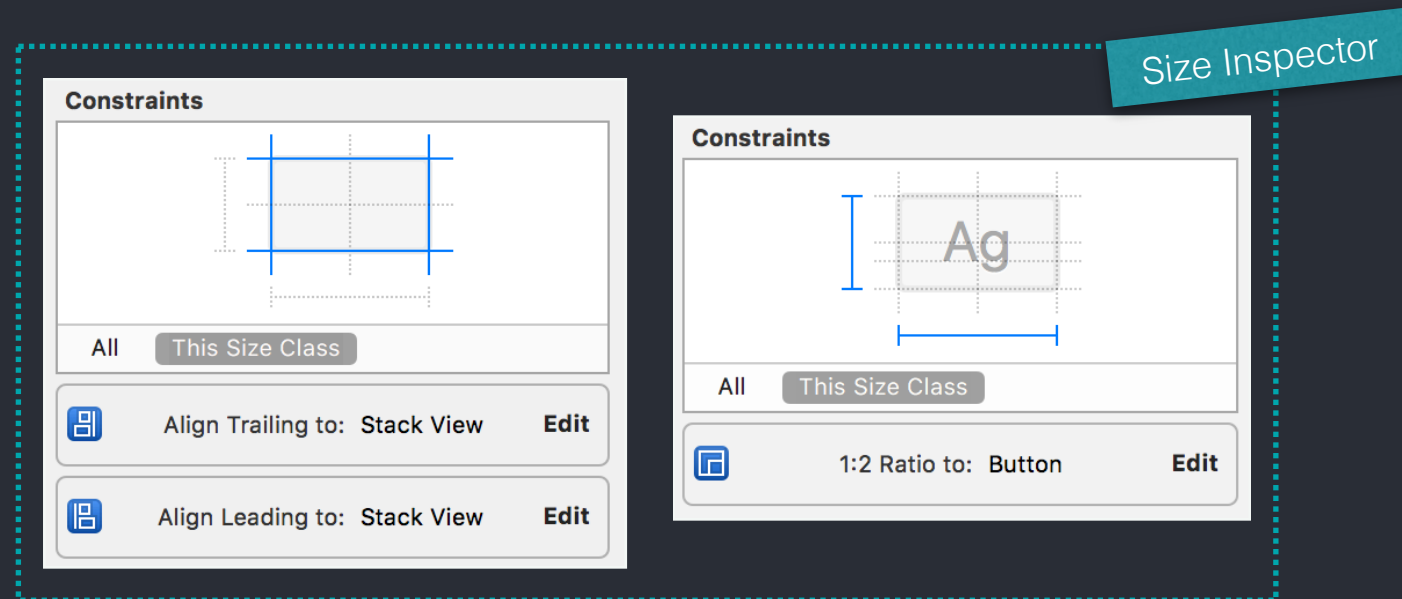


# Autolayout

- Setzen von **Constraints** für Views
- Ausrichtung von Views an den **blauen Linien**
  - entweder **Reset to Suggested Constraints**
  - oder **Ctrl-Drag** von View zu View oder von View zu Kanten (Edges)
- Verifizieren im **Size Inspector**
- Eliminieren von **Magic Numbers**, indem die **Offsets** auf **Default oder 0** gesetzt werden
  - **Achtung**: Safe Area erlaubt aktuell keinen Default Offset (scheint ein Bug zu sein), weshalb Magic Numbers notwendig sind
- Beim Autolayout können Konflikte entstehen, wenn die Constraints nicht eindeutige (**ambiguous**) Informationen liefern
- Konflikte werden in der **Document Outline** oder im **Attribute Inspector** analysiert und behoben
- Constraints können sowohl mithilfe des Storyboards (primär) als auch im Code gesetzt werden
- Zudem lassen sich Constraints jederzeit aktivieren und deaktivieren

# Autolayout

- **Constraints** beschreiben eine **Beziehung** zwischen zwei **Views**
- Beispielhafte Beziehungen sind
  - **Leading-**, **Trailing-**, **Top-**, **Bottom-** **Spaces** oder **Alignments**
  - **Center Vertically** oder **Horizontally**
- Zudem gibt es auf sich bezogene Beziehungen wie **Width**, **Height** oder **Aspect-Ratio**
- Jede Beziehung hat zusätzlich einen **Offset** von **n**, **0** oder **Default** und einem **Multiplier**
- Zudem ist die Beziehung **Equal**, **Greater Than** or Equal oder **Less Than** or Equal



# Heute

Layout und Views

Autolayout

**StackView**

Animationen

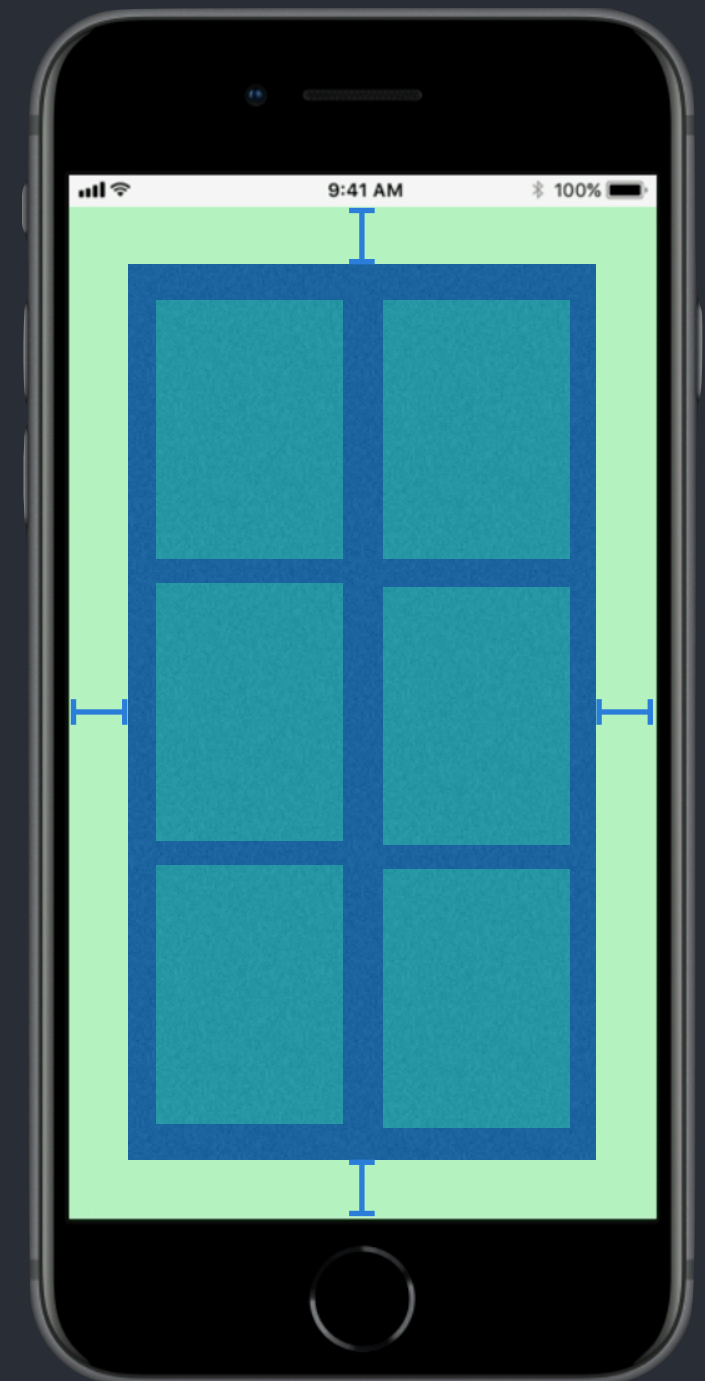
Size Classes

Demo

Assignment

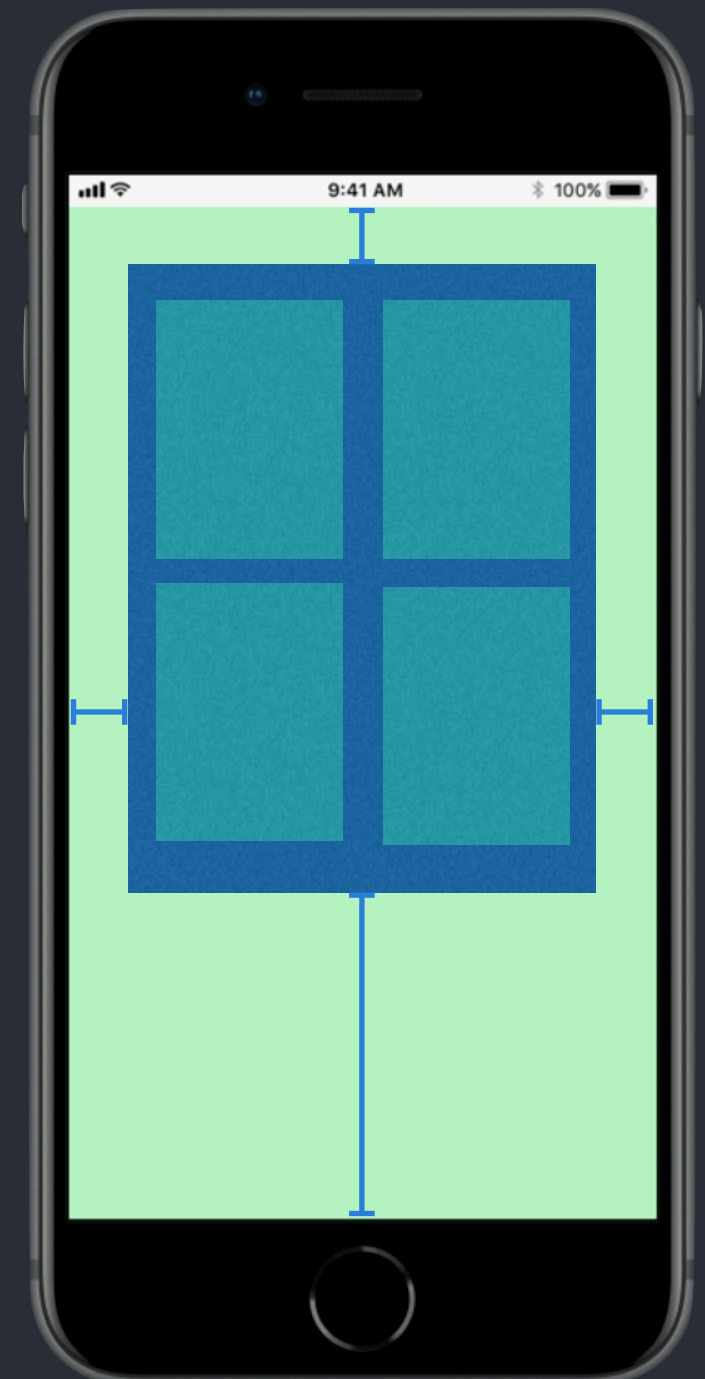
# StackView

- **Container** für mehrere Views
- Vorteil: StackView ausrichten und konfigurieren, SubViews passen sich an
- StackView wird mit Autolayout ausgerichtet
- Attribute der StackView steuern die interne Ausrichtung der Views auf **Basis der äußeren Constraints**
  - Horizontale oder Vertikale Orientierung der Views
  - Alignment, Distribution und Spacing
- **Zur Laufzeit** können Views
  - hinzugefügt, entfernt,
  - eingeblendet oder ausgeblendet werden.
- In jedem Falle **greifen alle Constraints der StackView**, sofern eine Änderung erfolgt ist
- Ohne StackView müsste man a) die View entfernen und b) die jeweiligen Autolayout Constraints finden und anpassen



# StackView

- **Container** für mehrere Views
- Vorteil: StackView ausrichten und konfigurieren, SubViews passen sich an
- StackView wird mit Autolayout ausgerichtet
- Attribute der StackView steuern die interne Ausrichtung der Views auf **Basis der äußeren Constraints**
  - Horizontale oder Vertikale Orientierung der Views
  - Alignment, Distribution und Spacing
- **Zur Laufzeit** können Views
  - hinzugefügt, entfernt,
  - eingeblendet oder ausgeblendet werden.
- In jedem Falle **greifen alle Constraints der StackView**, sofern eine Änderung erfolgt ist
- Ohne StackView müsste man a) die View entfernen und b) die jeweiligen Autolayout Constraints finden und anpassen



# Heute

Layout und Views

Autolayout

StackView

Animationen

Size Classes

Demo

Assignment

# Animationen

- UIView FSIOS Scope
  - Sichtbarkeit, Größe und Position, Transformationen, Alpha, Farbe, ...
  - Interpolation von Properties oder Transitionen zwischen Views
  - **Closure** beschreibt die Animation, welche über Zeit sichtbar wird
  - **UIViewAnimationOptions** beschreiben das Verhalten
    - curveLinear, curveEaseIn, curveEaseInEaseOut, allowUserInteraction, ...
- UIViewController Transitionen, Rotationen etc.
- Core Animation als fundamentales Framework
- Dynamic Animator für physikalische Animationen
  - Collision, Gravity, Push, Snap, ...



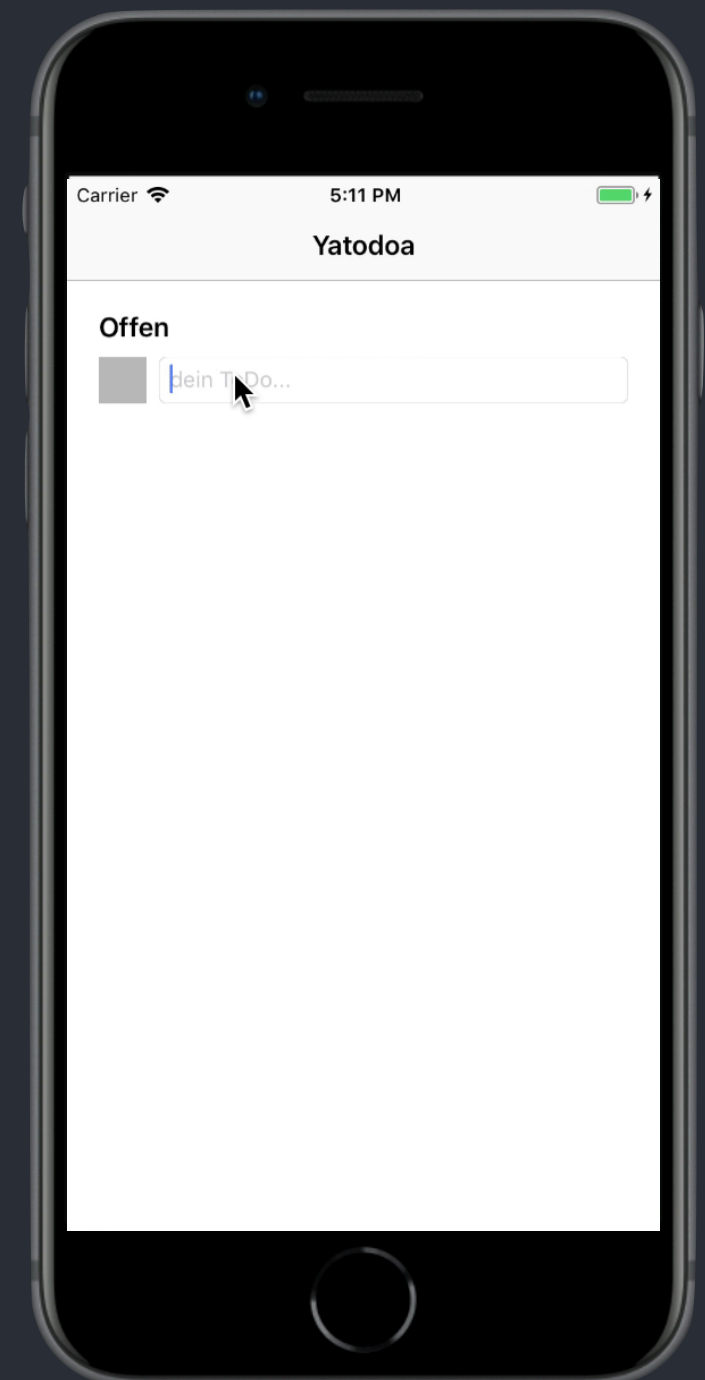
# Animationen

```
extension UIView {  
  
    open class func animate(  
        withDuration duration: TimeInterval,  
        delay: TimeInterval,  
        options: UIViewAnimationOptions = [],  
        animations: @escaping () -> Swift.Void,  
        completion: ((Bool) -> Swift.Void)? = nil  
    )  
  
    open class func animate(  
        withDuration duration: TimeInterval,  
        animations: @escaping () -> Swift.Void,  
        completion: ((Bool) -> Swift.Void)? = nil  
    ) // delay = 0.0, options = 0  
  
    open class func animate(  
        withDuration duration: TimeInterval,  
        animations: @escaping () -> Swift.Void  
    ) // delay = 0.0, options = 0, completion = NULL  
}
```

UIKit API

```
let view = ToDoView()  
view.isHidden = true // 'view' is hidden  
  
todosStackView.addArrangedSubview(view) // 'view still hidden'  
  
UIView.animate(withDuration: 0.3, animations: {  
    view.isHidden = false // interpolation of property within  
    0.3s from 'hidden = true' to 'hidden = false'  
})
```

Benutzung



# Heute

Layout und Views

Autolayout

StackView

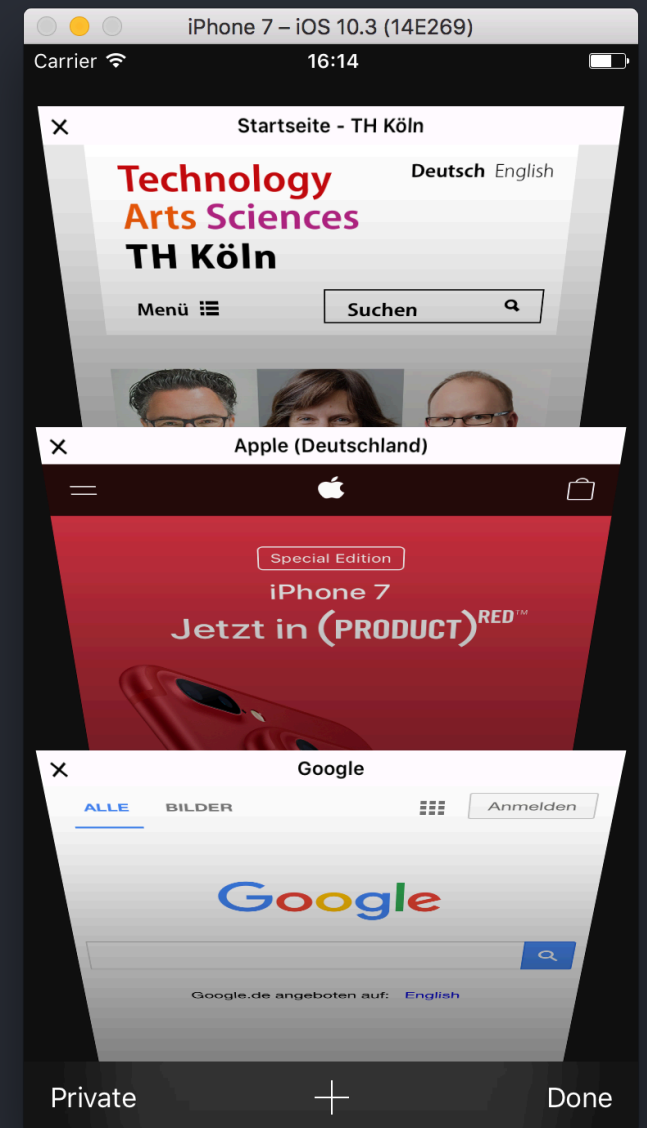
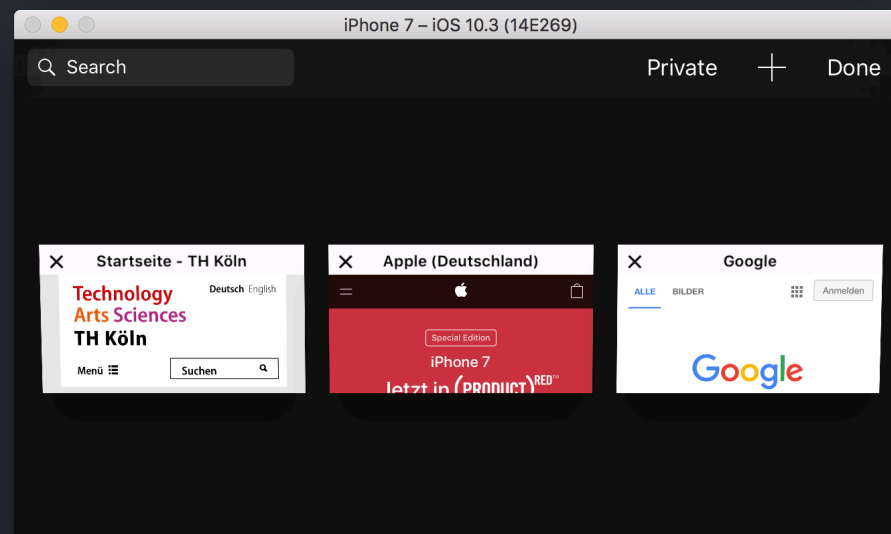
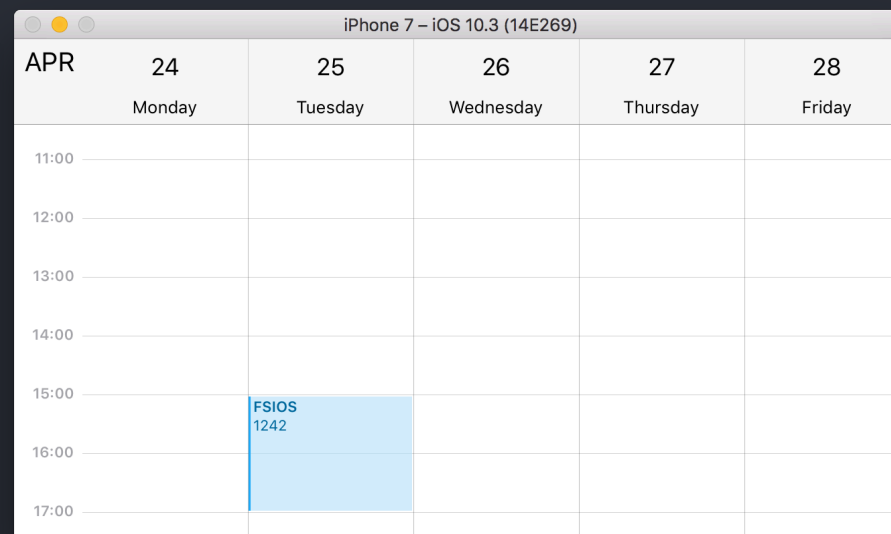
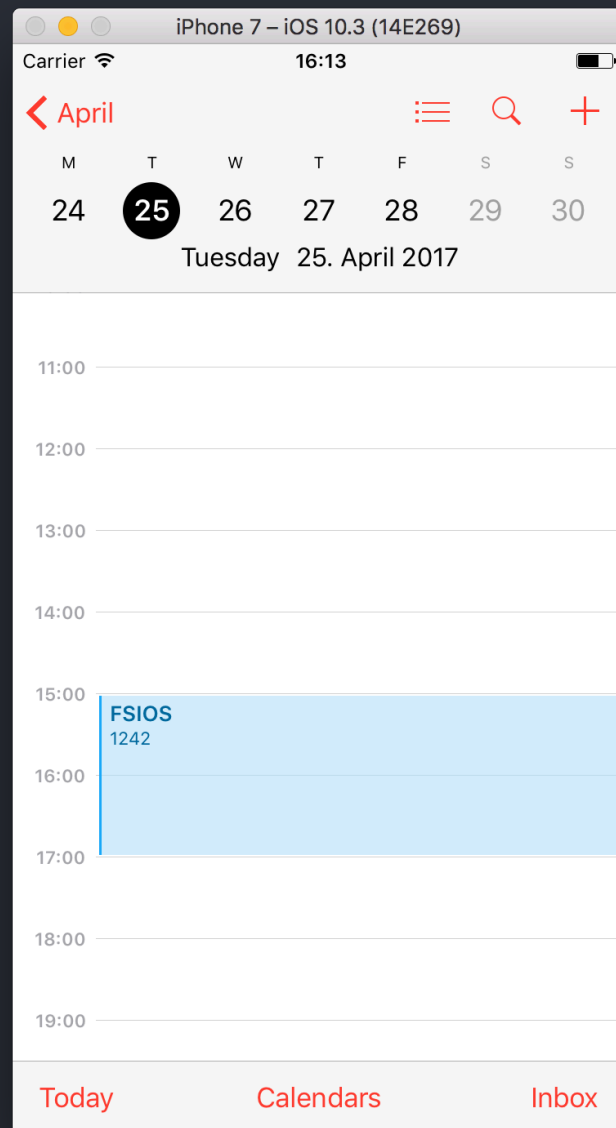
Animationen

Size Classes

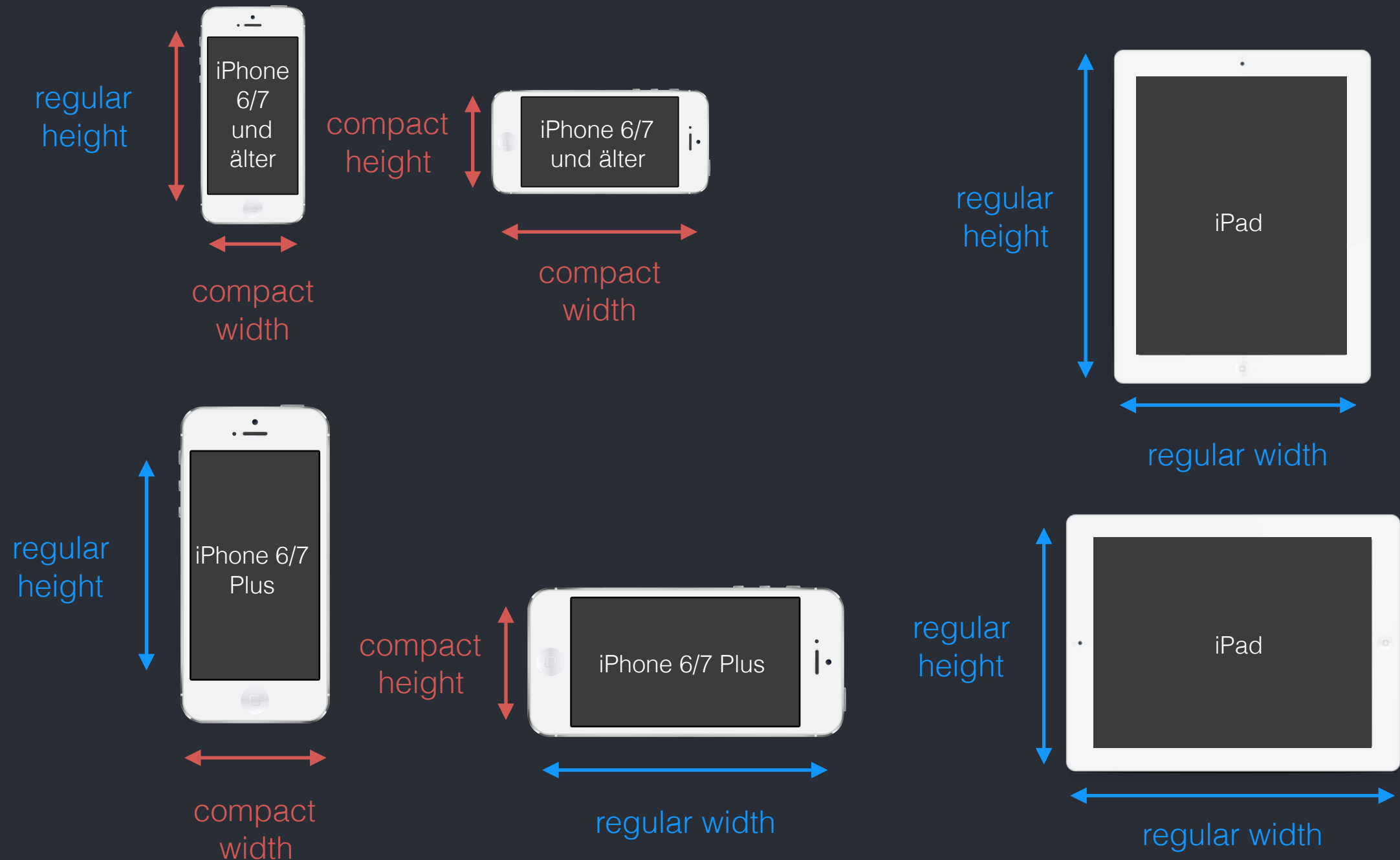
Demo

Assignment

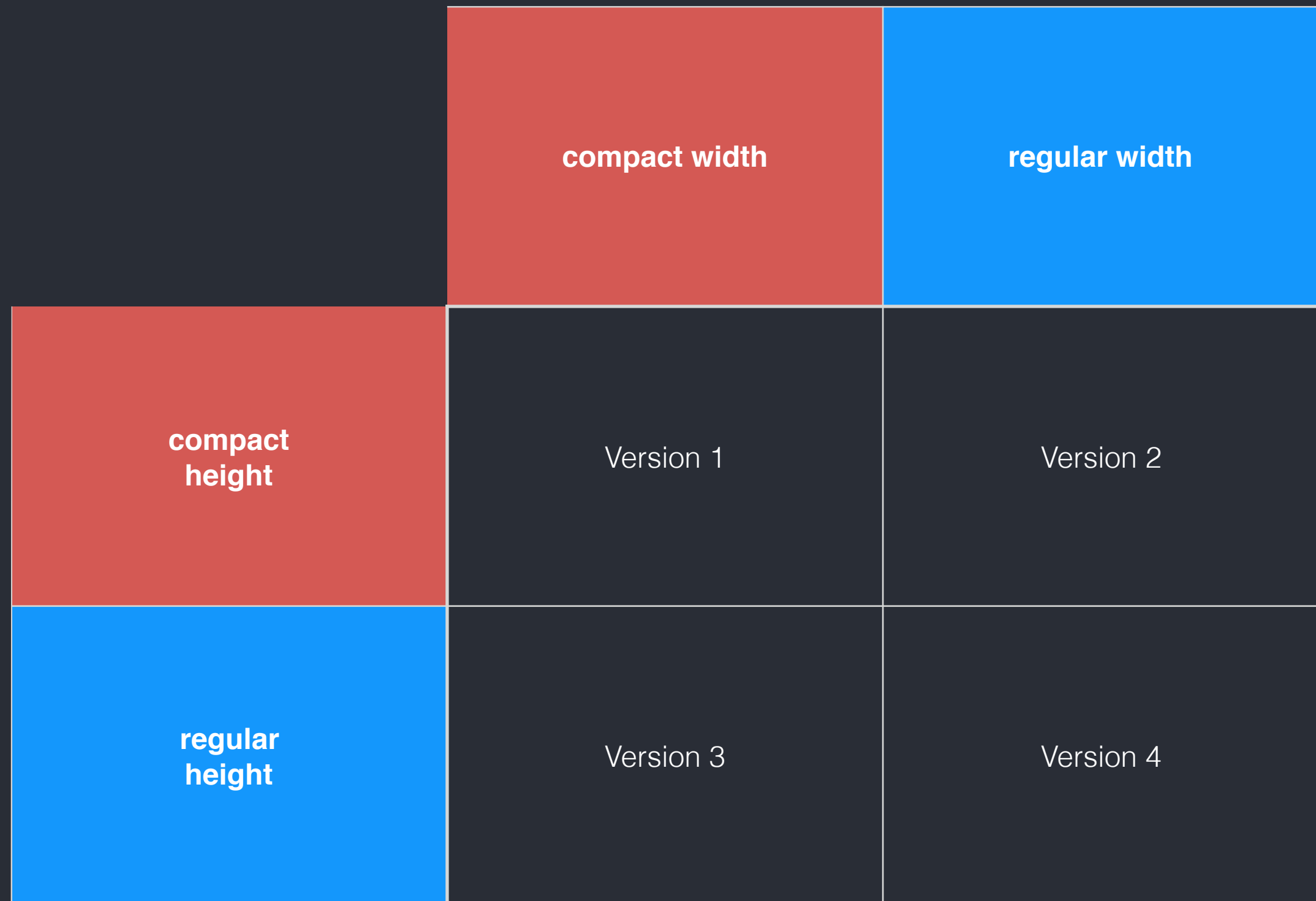
# Size Classes



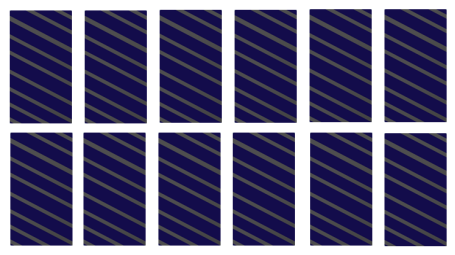
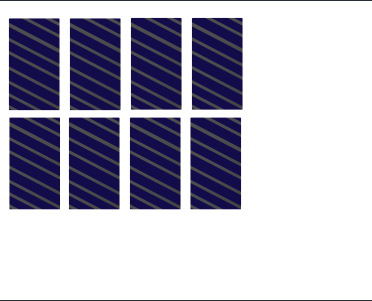
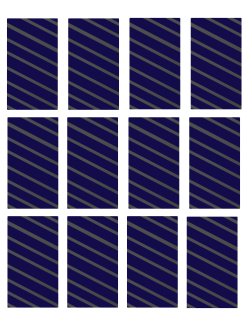
# Size Classes



# Size Classes

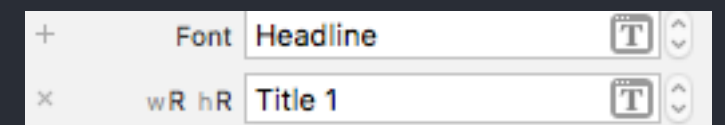


# Size Classes

	compact width	any width	regular width
compact height			
any height			
regular height			

# Size Classes

- **Initiales Design** für wA (Any Width) und hA (Any Height)
- **Angepasstes Design** für wC/R und hC/R
  - Constraints hinzufügen oder entfernen
  - Views einblenden oder ausblenden Neu mit Xcode 9
- **Angepasste Attribute** für eine bestimmte Size Class
  - Gegeben ist wC hR mit *Headline* als Font
  - Eine Variation der Font für wR hR setzt die Font auf *Title 1*



# Heute

Layout und Views

Autolayout

StackView

Animationen

Size Classes

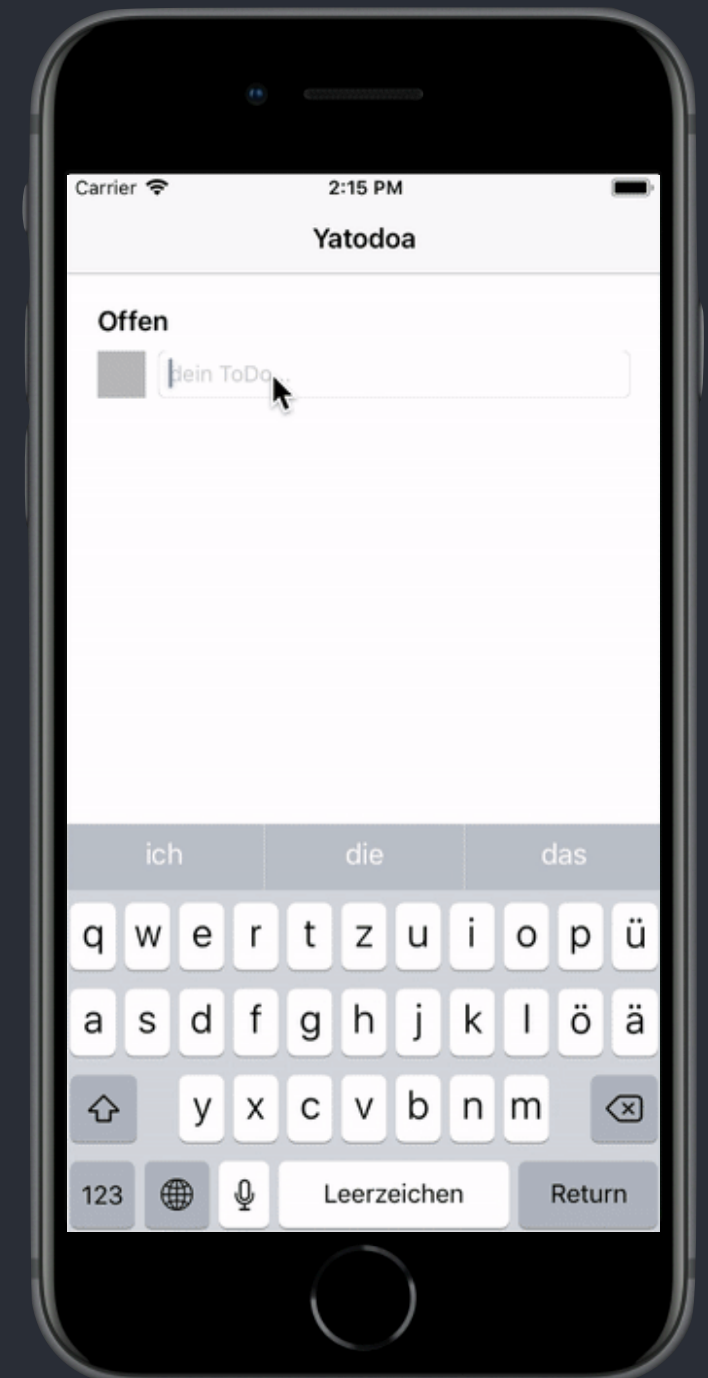
Demo

Assignment



# Yatodoa - Demo

- UINavigationController und NavigationItem
- Verschachtelte Views
- Autolayout
- ScrollView
- UIStackView
  - add, insert und remove arranged subviews
- Animationen
  - alpha und visibility



# Heute

Layout und Views

Autolayout

StackView

Animationen

Size Classes

Demo

Assignment

# Yatodoa - Assignment

- Zweiter Stack für Erledigte Todos
- Sobald ein Todo erledigt wird, soll das Todo in der dafür vorgesehenen StackView erscheinen und vice versa
- Sowohl das Hinzufügen als auch das Entfernen aus den jeweiligen StackViews ist animiert
- Das “Erledigt” Label zählt die Anzahl der erledigten Todos und aktualisiert sich bei jeder Aktion
- Falls es keine erledigten Todos gibt, verschwindet das Label (ebenfalls animiert)
- Tipps
  - Das Hinzufügen und Entfernen zuerst ohne Animationen implementieren, um die Funktionalität zu sichern
  - Hier bauen Animationen aufeinander auf. Verwendet hierfür die passende `UIView.animate` Funktion, welche einen ``completion`` handler entgegennimmt
- Sonstige Änderungen und Verbesserungen sind Willkommen
- Bis zum 21:11, 13:59 Uhr per Pull-Request einreichen

