

# Building Adaptive Apps

Let's clear the layout mist in iOS

Jinghan Wang iOS Dev@Sea Labs

# Adaptive Layout

Jinghan Wang iOS Dev@Sea Labs

# Agenda

- Safe Area
- Layout Margins
- UIScrollView
- What's next?

# Agenda

- Safe Area
- Layout Margins
- UIScrollView
- What's next?

API Checklist



Brief Intro



Demo



Recap

Safe Area

# API Checklist (Safe Area)

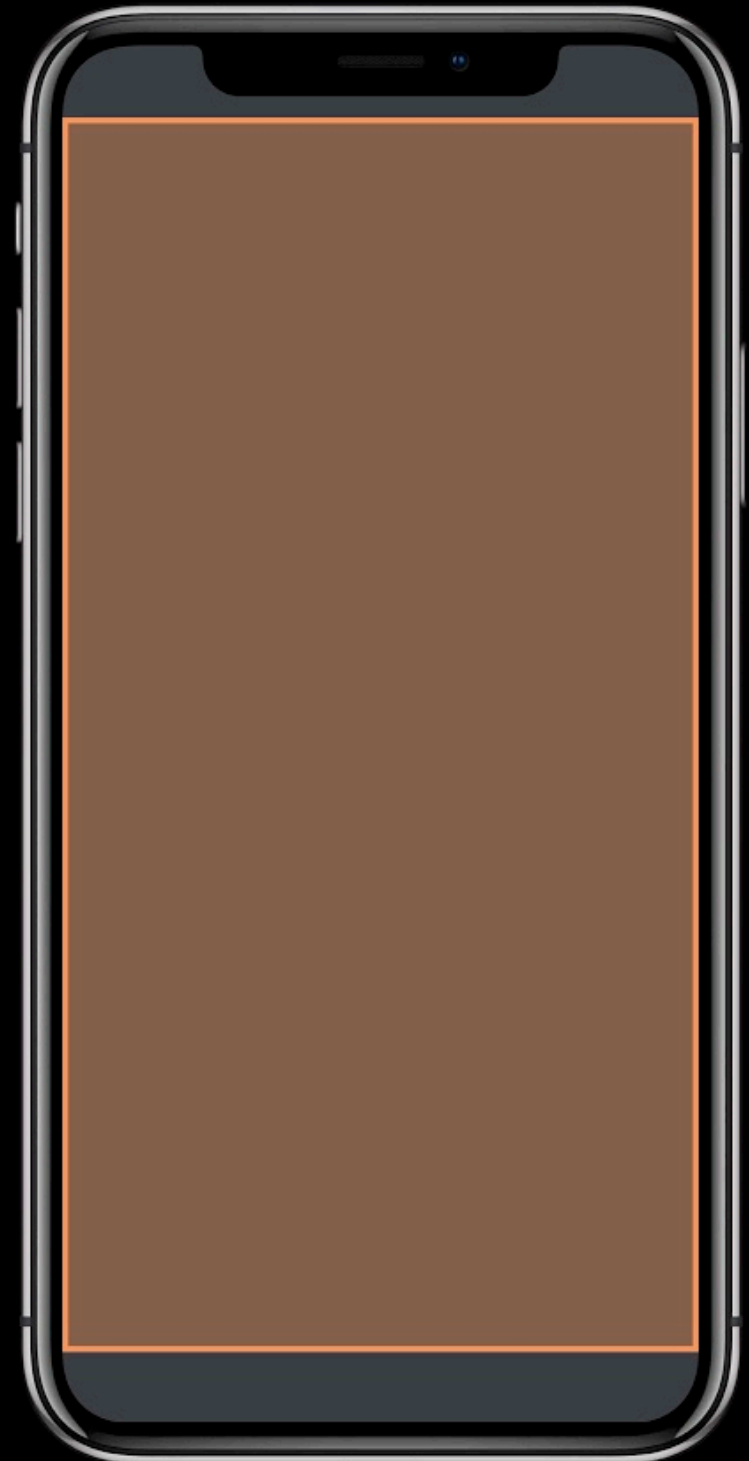
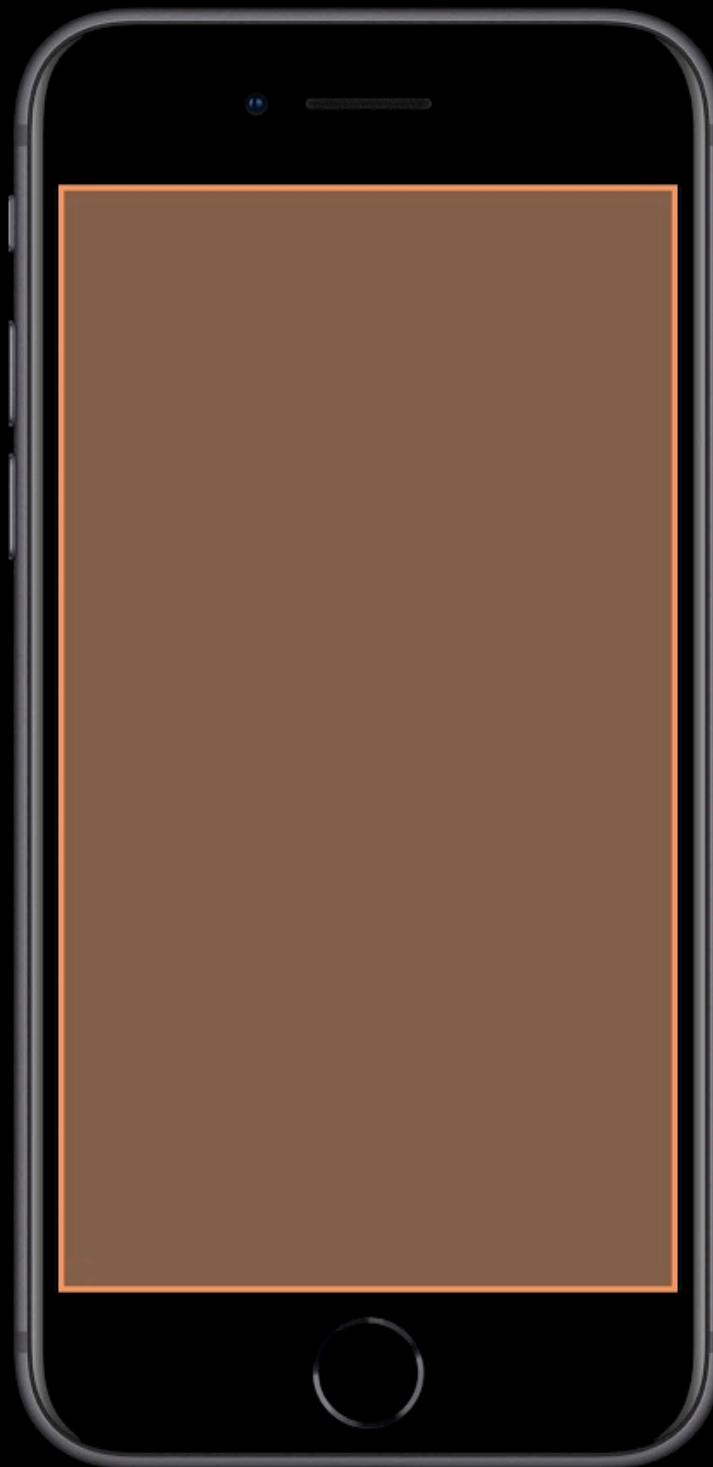
## UIView

- `.safeAreaInsets`
- `.safeAreaLayoutGuide`
- `.safeAreaInsetsDidChange()`

## UIViewController

- `.additionalSafeAreaInsets`
- `.viewSafeAreaInsetsDidChange()`

# Safe Area



# Safe Area (Landscape)





# Safe Area (Overscan Compensation)



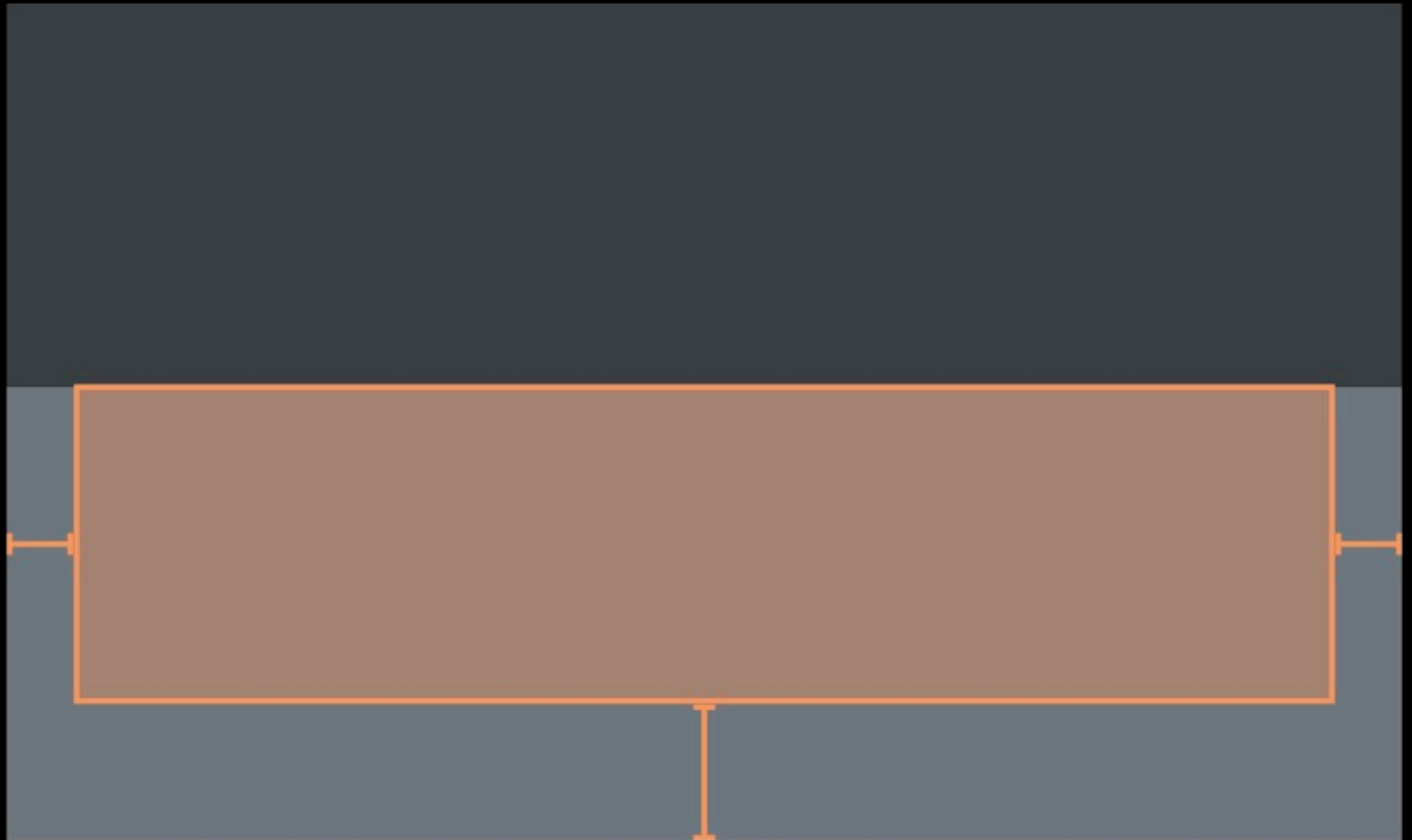
# Safe Area



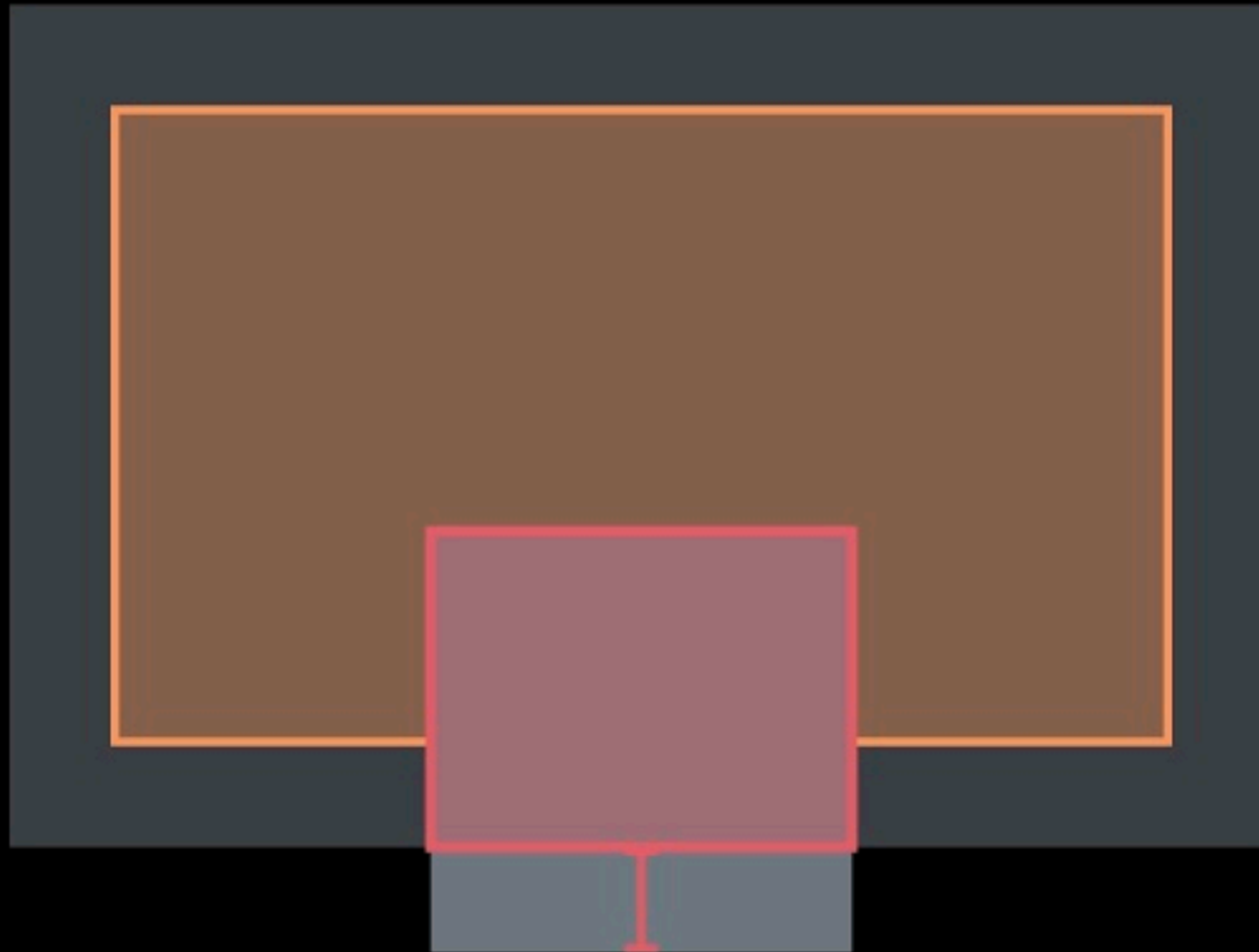
# Demo - Safe Area Behaviours

1. Subview propagation
2. Adding insets

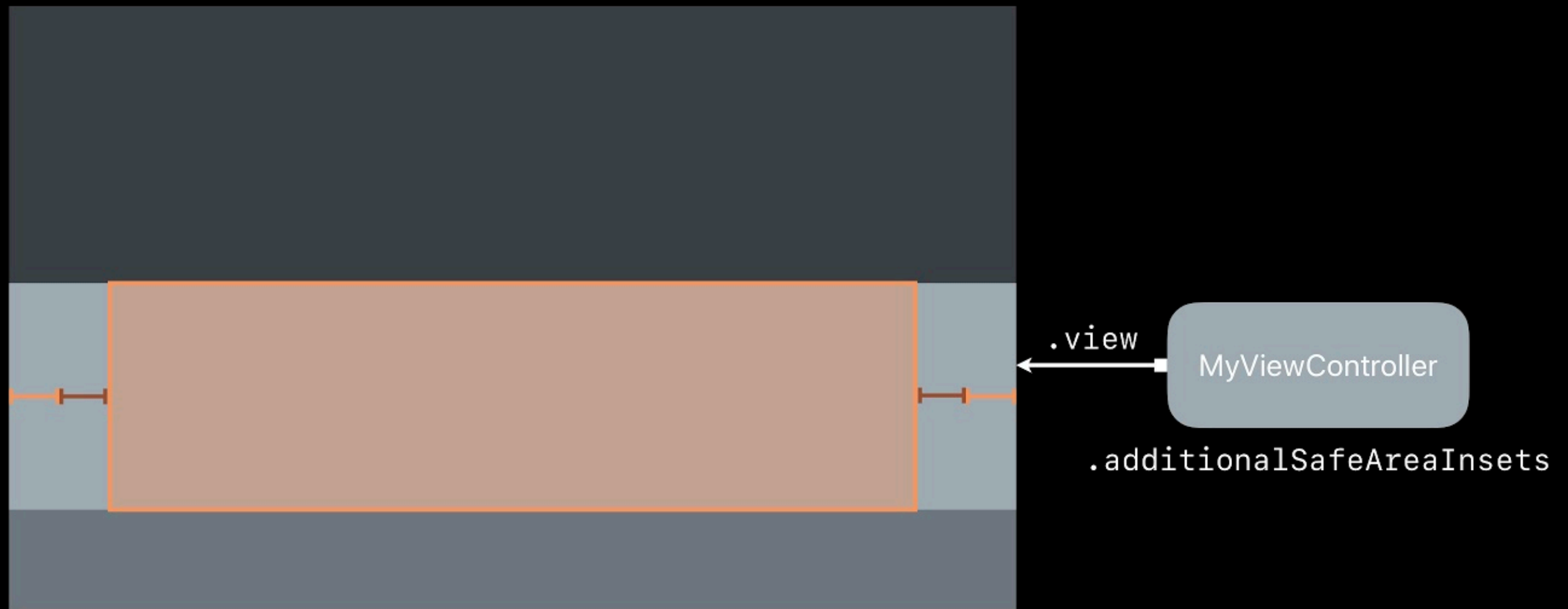
# Safe Area Propagation



# Safe Area Propagation



# Additional Safe Area



# API Checklist (Safe Area)

## UIView

- `.safeAreaInsets`
- `.safeAreaLayoutGuide`
- `.safeAreaInsetsDidChange()`

## UIViewController

- `.additionalSafeAreaInsets`
- `.viewSafeAreaInsetsDidChange()`

# Layout Margins



# API Checklist (Layout Margins)

## UIView

- .layoutMargins
- .layoutMarginsGuide
- .directionalLayoutMargins
- .layoutMarginsDidChange()
- .insetsLayoutMarginsFromSafeArea
- .preserveSuperviewLayoutMargins

## UIViewController

- .viewLayoutMarginsDidChange()
- .systemMinimumLayoutMargins
- .viewRespectsSystemMinimumLayoutMargins

# Layout Margins



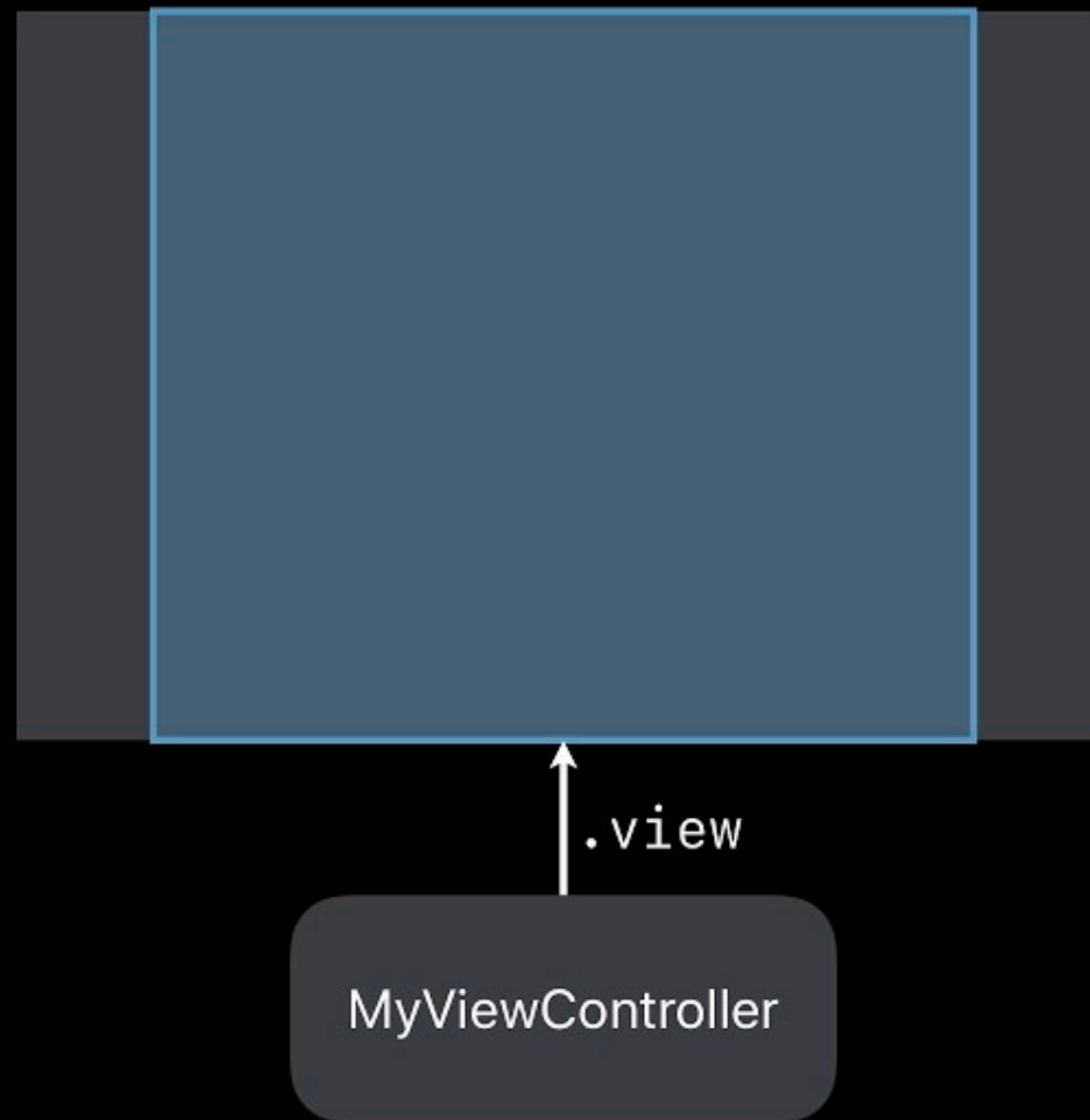
`.layoutMargins`

`.directionalLayoutMargins`

# Demo - Layout Margins Behaviours

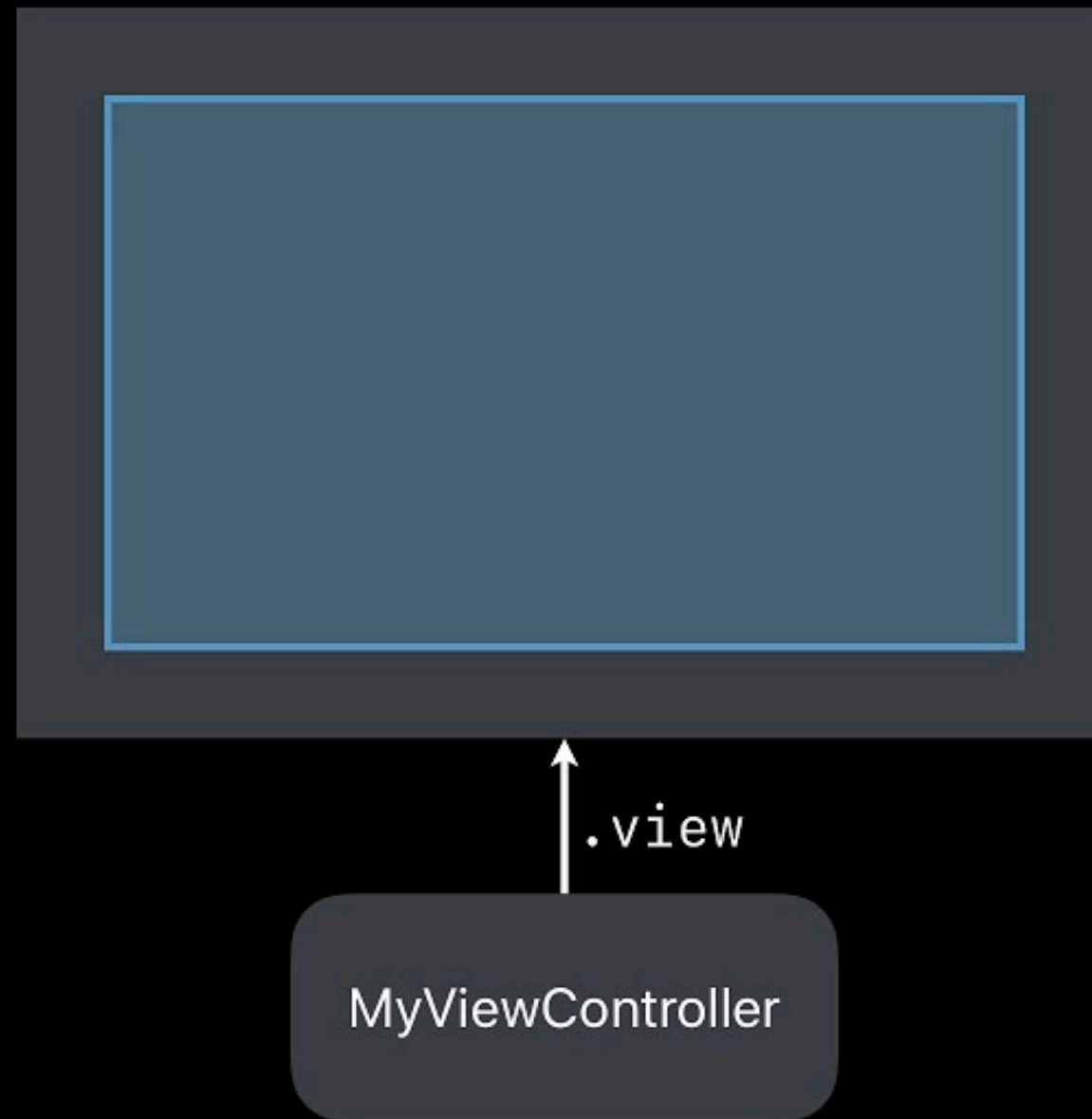
1. System minimum layout margins
2. Layout margins and safe area
3. Margin propagation
4. Callback behaviours

# System Margins



```
.viewRespectsSystemMinimumLayoutMargins = true
```

# System Margins



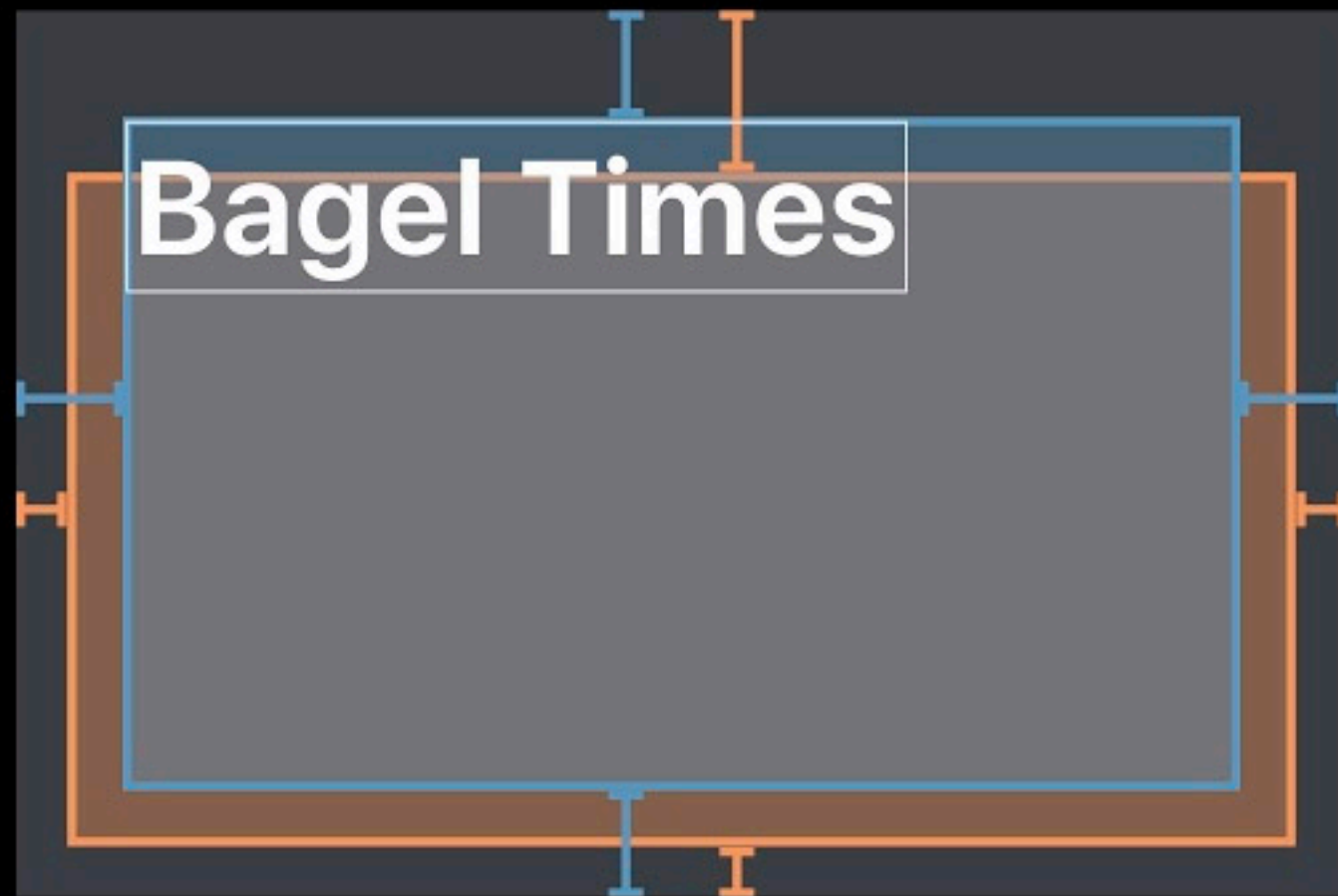
```
.viewRespectsSystemMinimumLayoutMargins = false
```

# Layout Margins (with safe area)



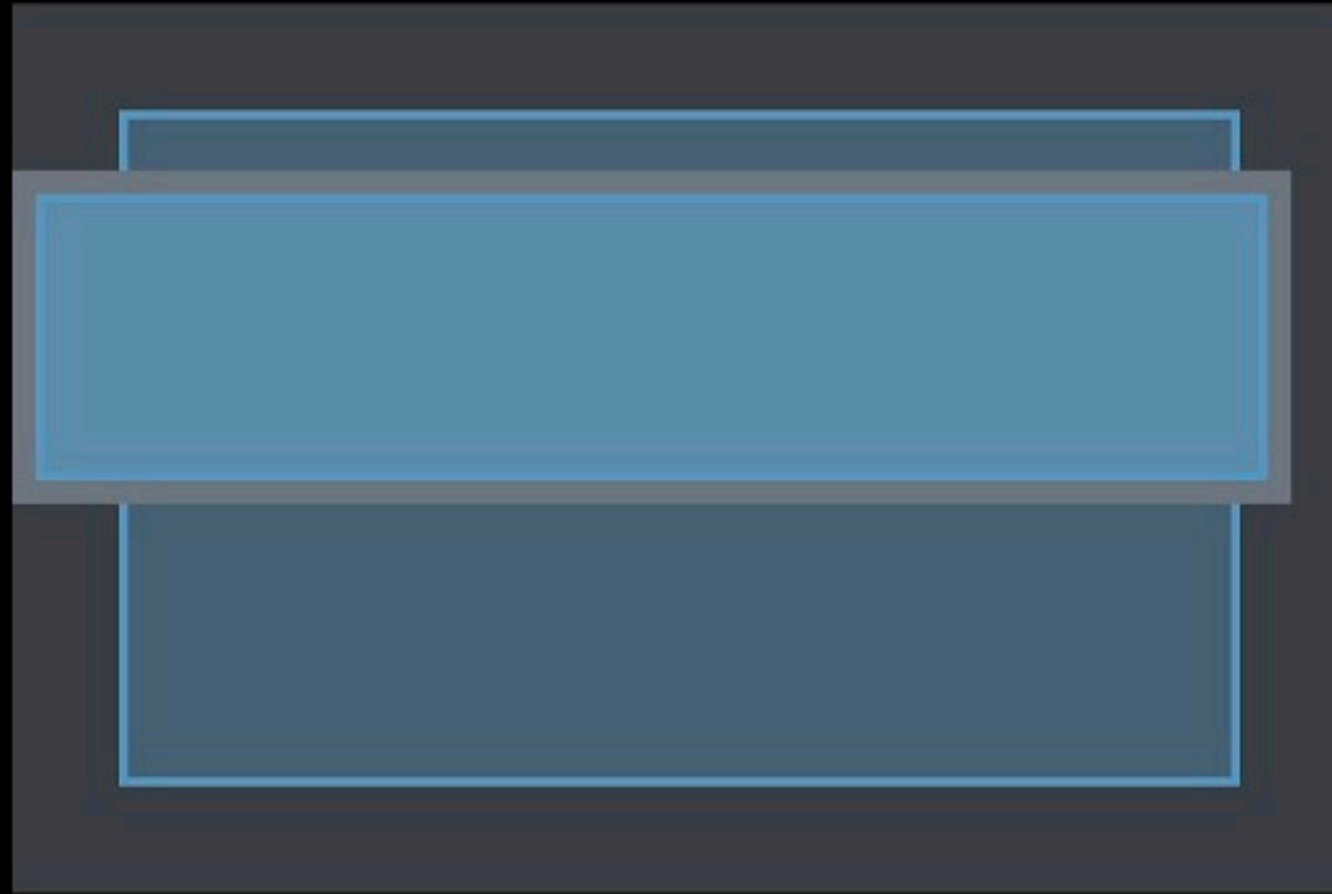
```
.insetsLayoutMarginsFromSafeArea = true
```

# Layout Margins (with safe area)



```
.insetsLayoutMarginsFromSafeArea = false
```

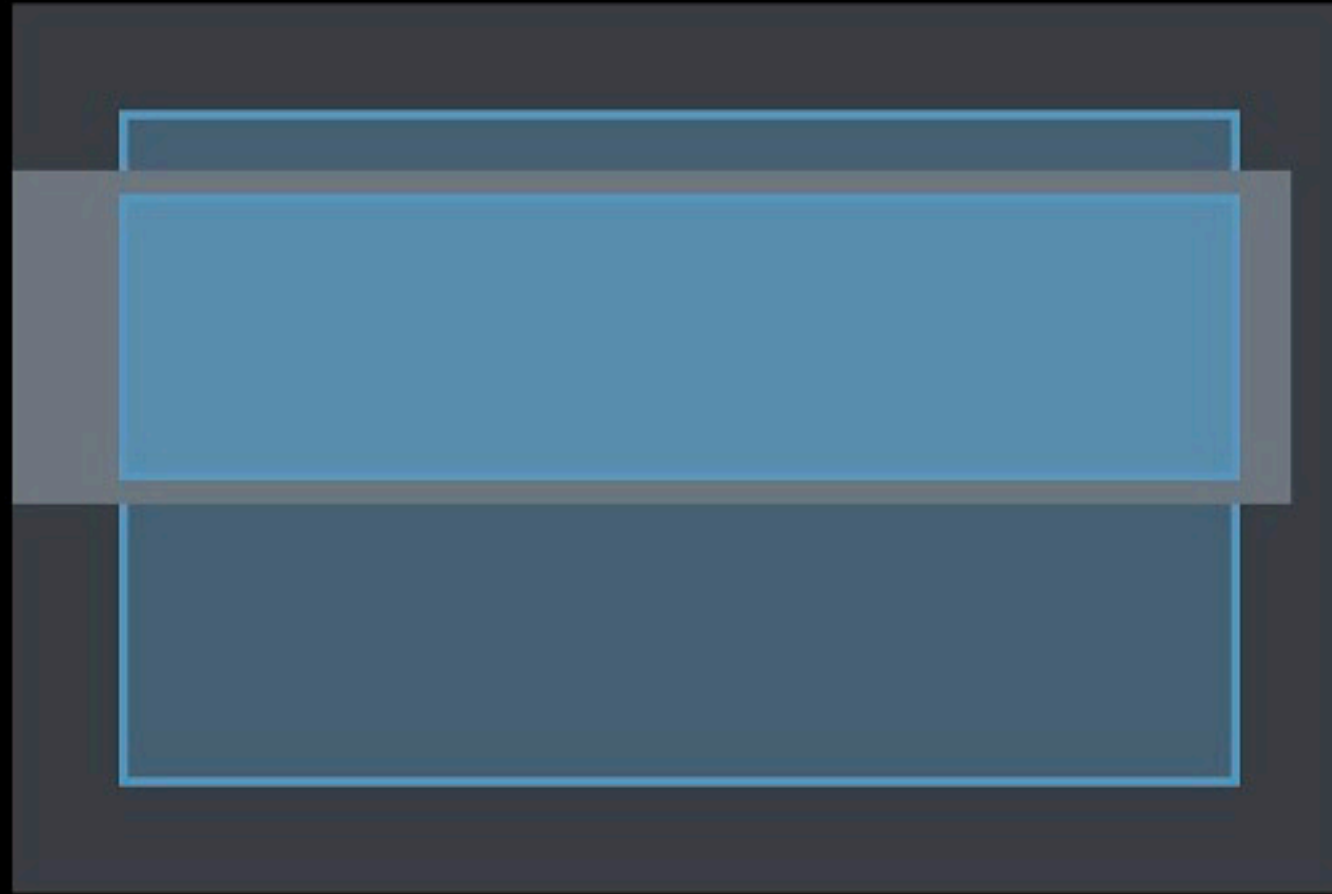
# Layout Margin Propagation



`.preservesSuperviewLayoutMargins = false`



# Layout Margin Propagation



```
.preservesSuperviewLayoutMargins = true
```

# When a view controller is being shown...

controller.viewWillAppear



view.didMoveToWindow



view.updateSafeAreaInsets



Handle System Minimum Layout Margin (In UINavigationController)



controller.viewDidAppear

# Whenever safe area insets is updated...

`view.layoutMarginsDidChange`



`controller.viewLayoutMarginsDidChange`



`view.safeAreaInsetsDidChange`



`controller.viewSafeAreaInsetsDidChange`

# API Checklist (Layout Margins)

## UIView

- .layoutMargins
- .layoutMarginsGuide
- .directionalLayoutMargins
- .layoutMarginsDidChange()
- .insetsLayoutMarginsFromSafeArea
- .preserveSuperviewLayoutMargins

## UIViewController

- .viewLayoutMarginsDidChange()
- .systemMinimumLayoutMargins
- .viewRespectsSystemMinimumLayoutMargins

# UIScrollView

# API Checklist (UIScrollView)

## UIViewController

`.automaticallyAdjustsScrollViewInsets`

## UIScrollView

`.bounds`

`.contentSize`

`.contentOffset`

`.contentInset`

`.adjustedContentInset`

`.contentInsetAdjustmentBehavior`

# Fundamentals

```
bounds.size.height = 400
```

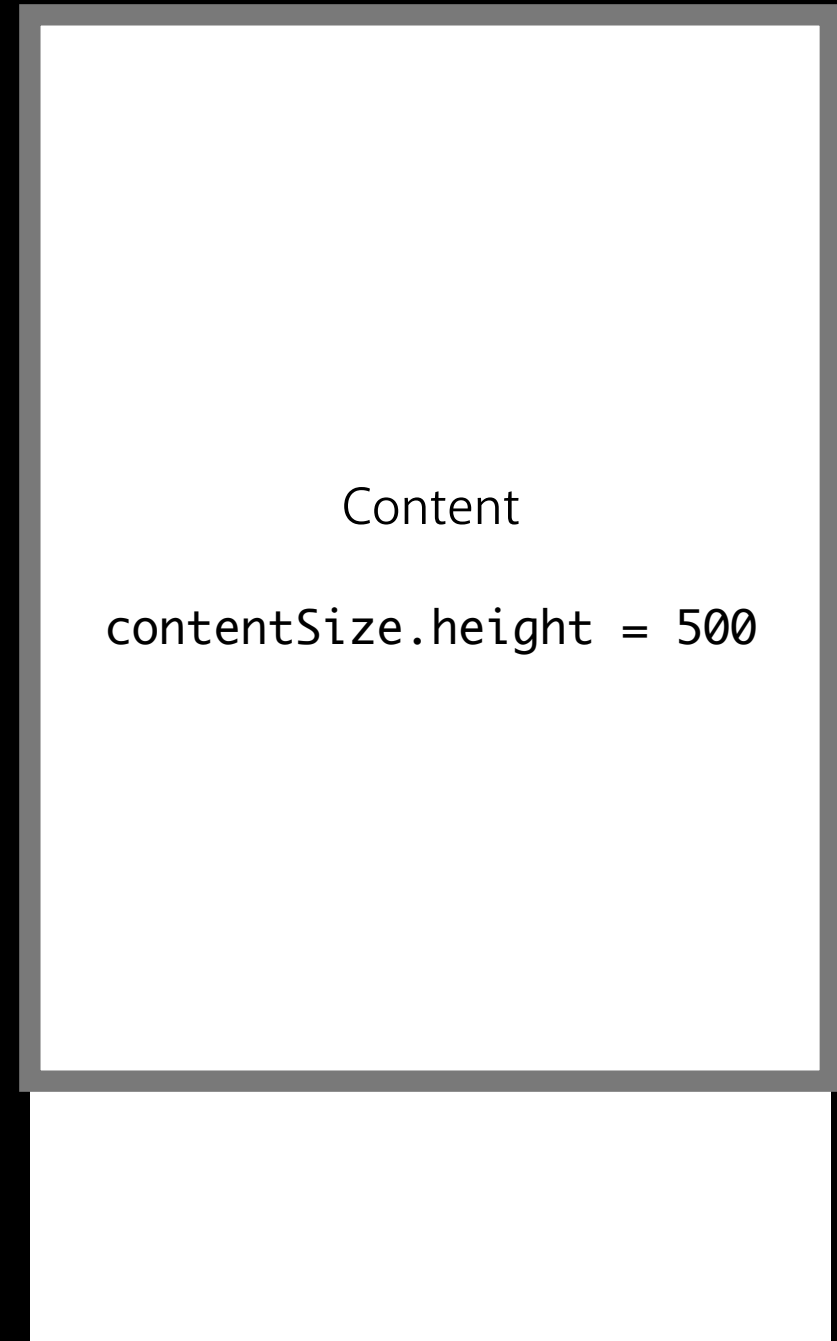


Content

# Fundamentals

`contentOffset.y = 0`

`bounds.size.height = 400`

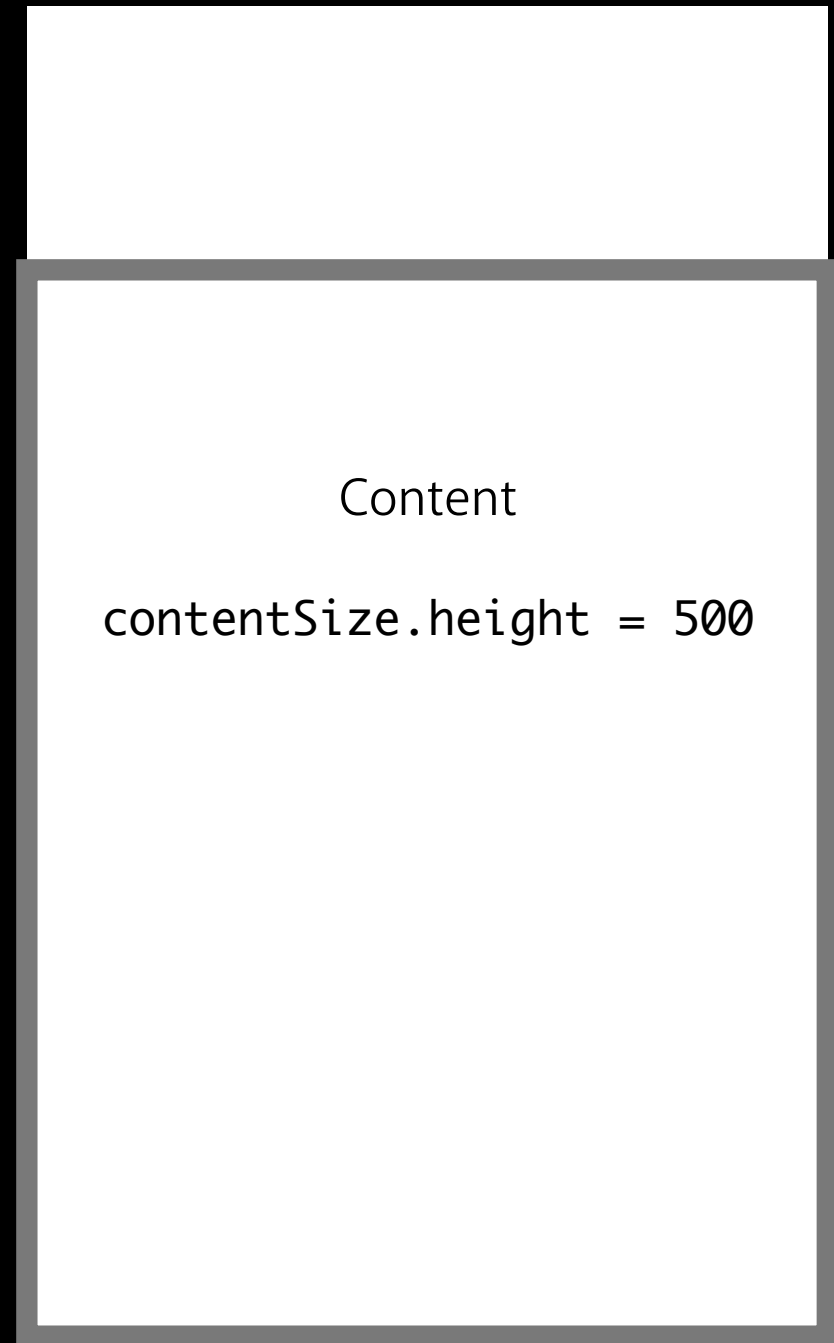




# Fundamentals

`contentOffset.y = 100`

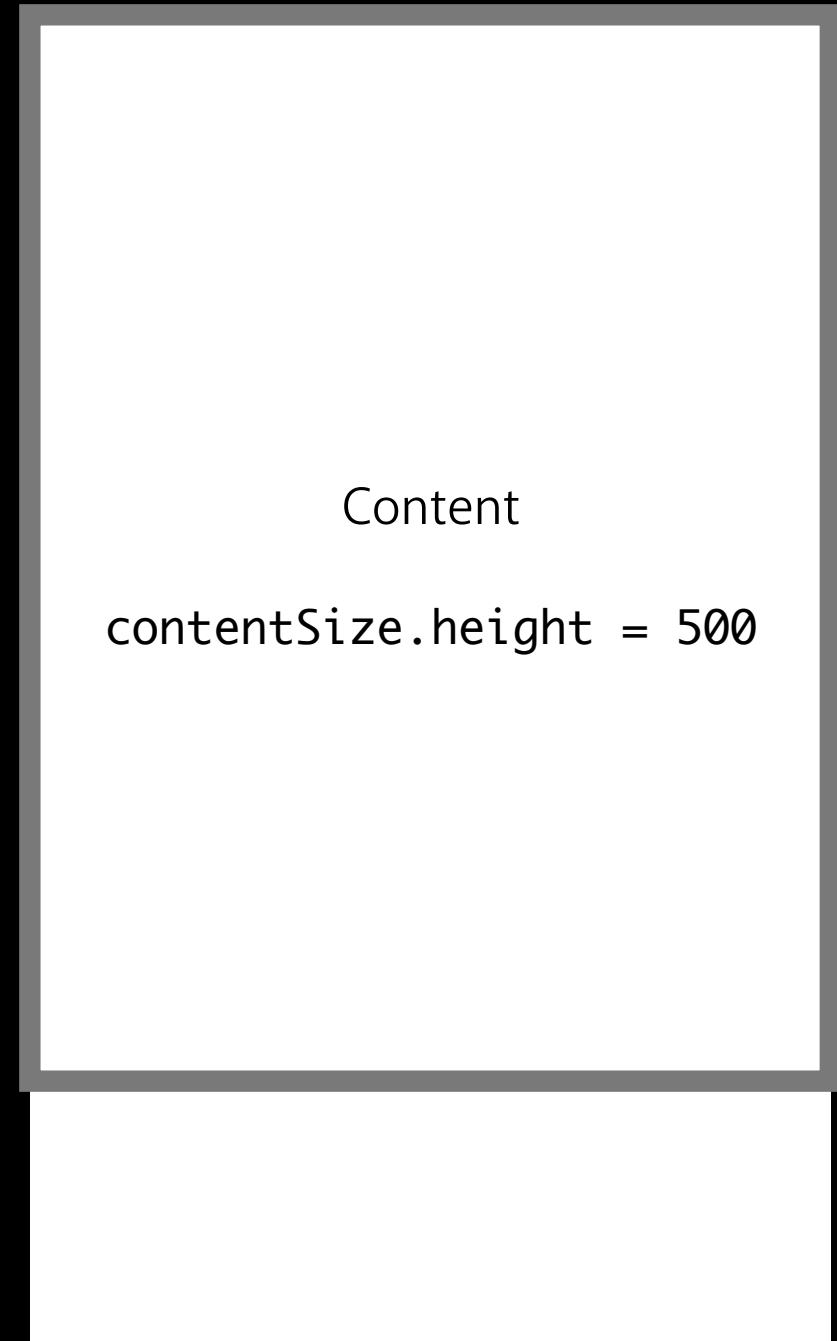
`bounds.size.height = 400`



# Fundamentals

`contentOffset.y = 0`

`bounds.size.height = 400`



# Fundamentals

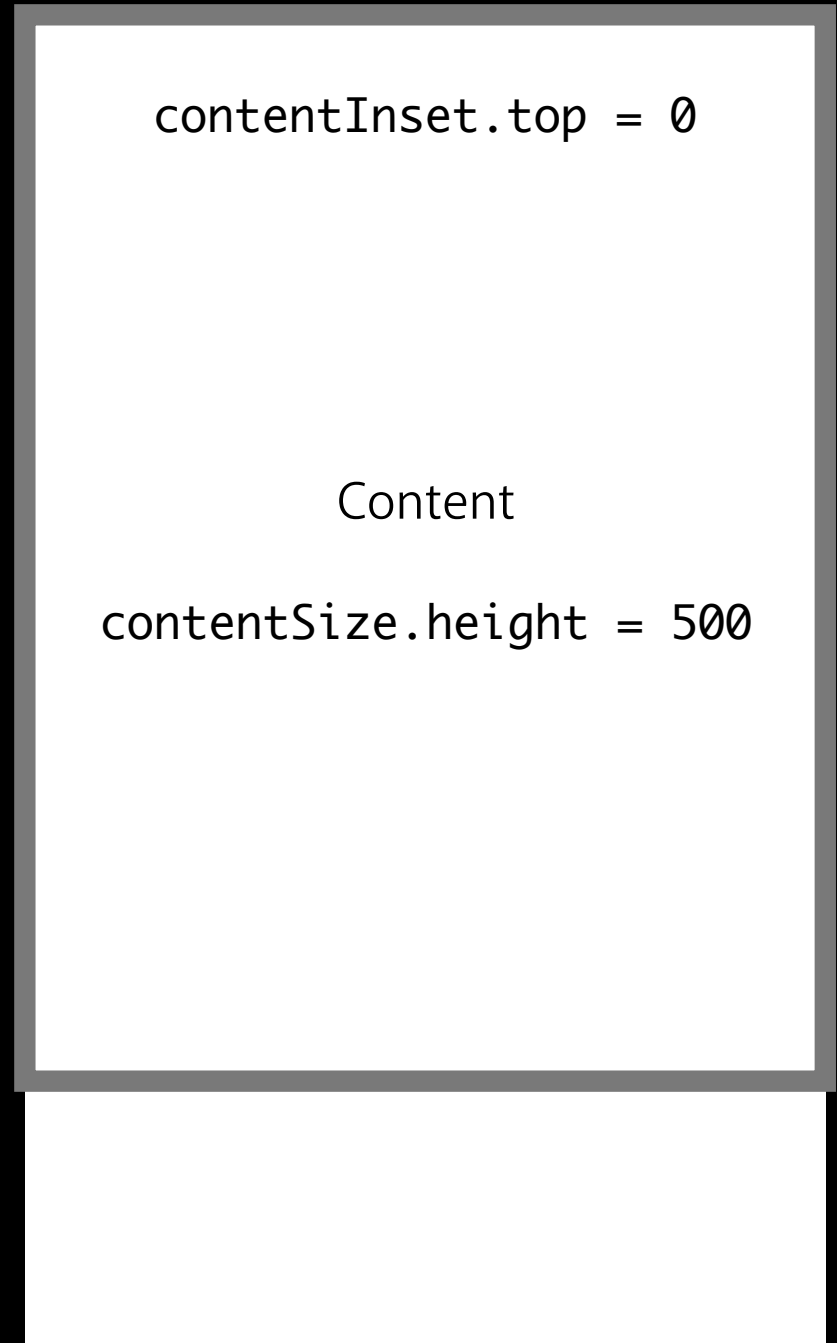
`contentOffset.y = 0`

`bounds.size.height = 400`

`contentInset.top = 0`

Content

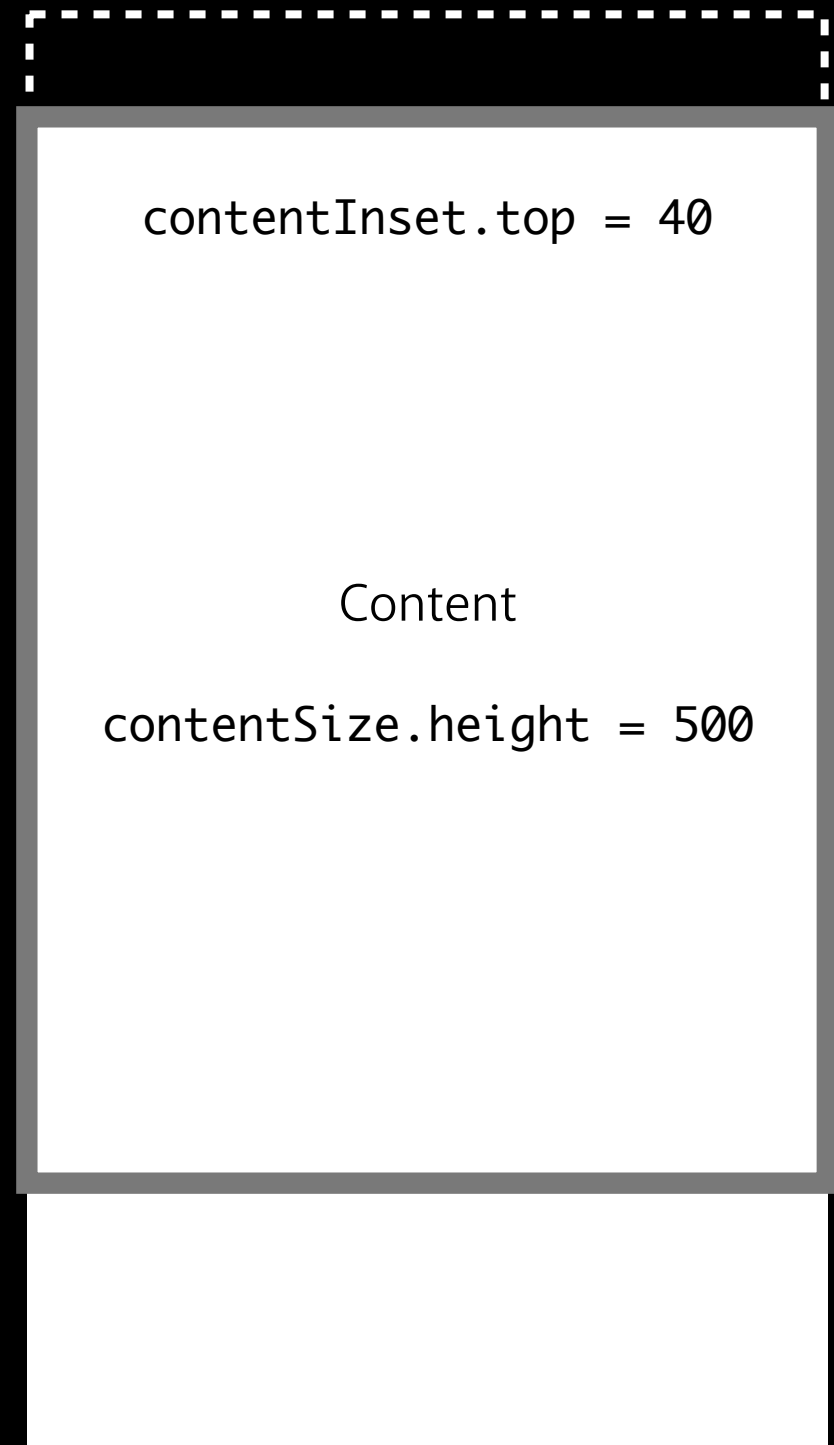
`contentSize.height = 500`



# Fundamentals

`contentOffset.y = 0`

`bounds.size.height = 400`



# Fundamentals

`contentOffset.y = 100`

`bounds.size.height = 400`

`contentInset.top = 40`

Content

`contentSize.height = 500`

# Fundamentals

`contentOffset.y = -40`

`bounds.size.height = 400`



# Automatic Content Inset

```
class UIViewController: UIResponder {  
    var automaticallyAdjustsScrollViewInsets: Bool  
}
```

# Automatic Content Inset

```
class UIViewController: UIResponder {  
    var automaticallyAdjustsScrollViewInsets: Bool  
}
```

Deprecated



# Automatic Content Inset

```
class UIScrollView: UIView {  
    var adjustedContentInset: UIEdgeInsets { get }  
}
```

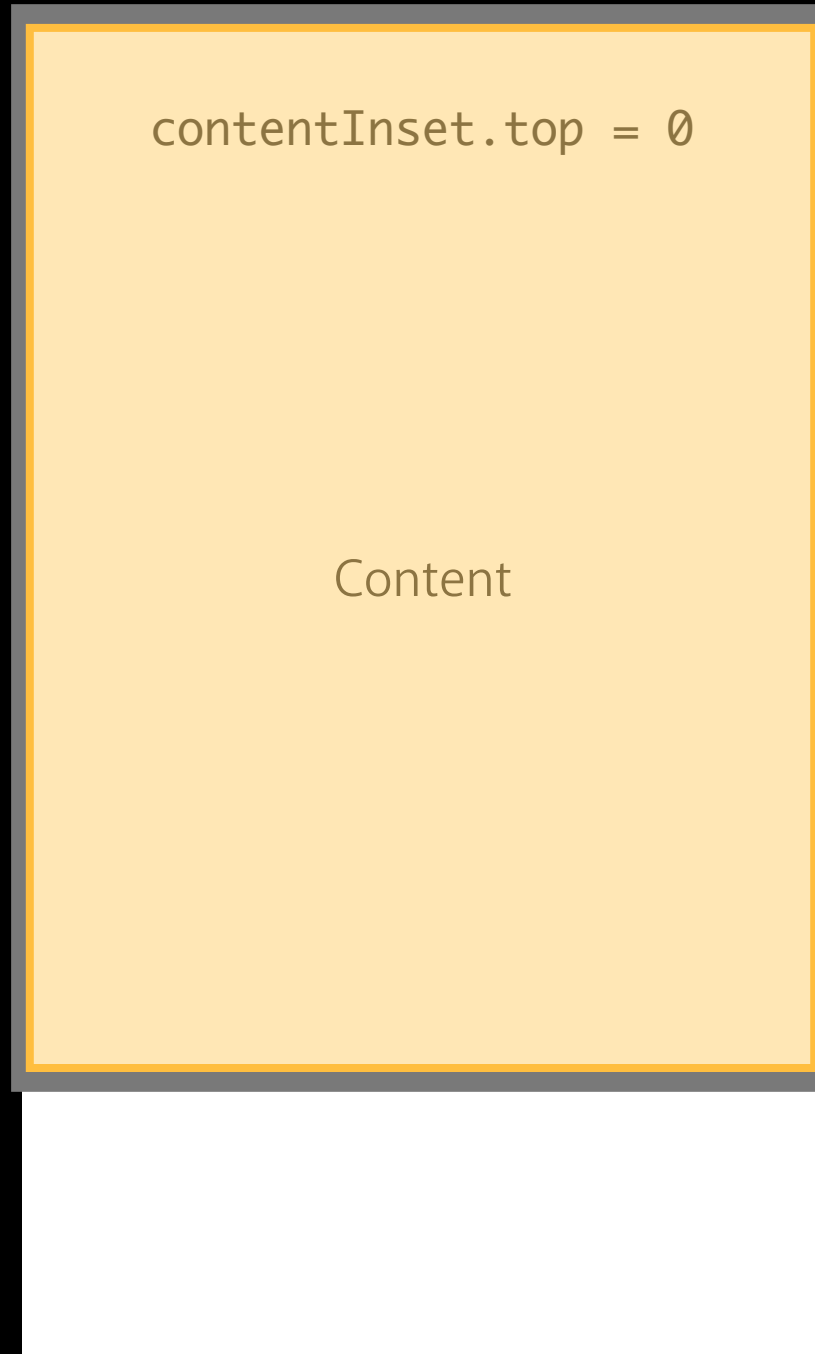
# Automatic Content Inset

```
class UIScrollView: UIView {  
    var adjustedContentInset: UIEdgeInsets { get }  
}
```

$\text{adjustedContentInset} = \text{contentInset} + \text{system inset}$

# ScrollView and Safe Area

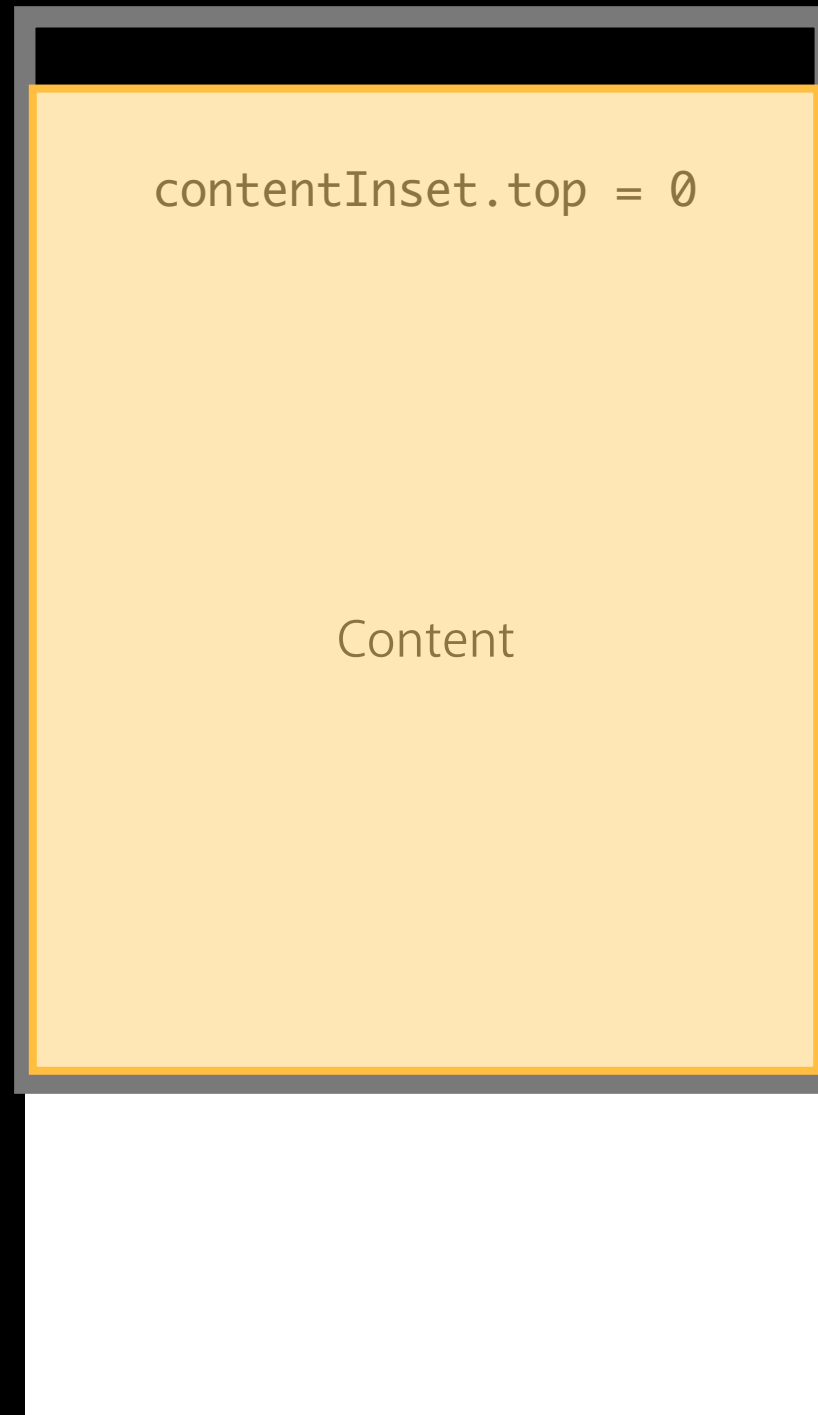
`adjustedContentInset.top = 0`



`safeAreaInsets.top = 0`

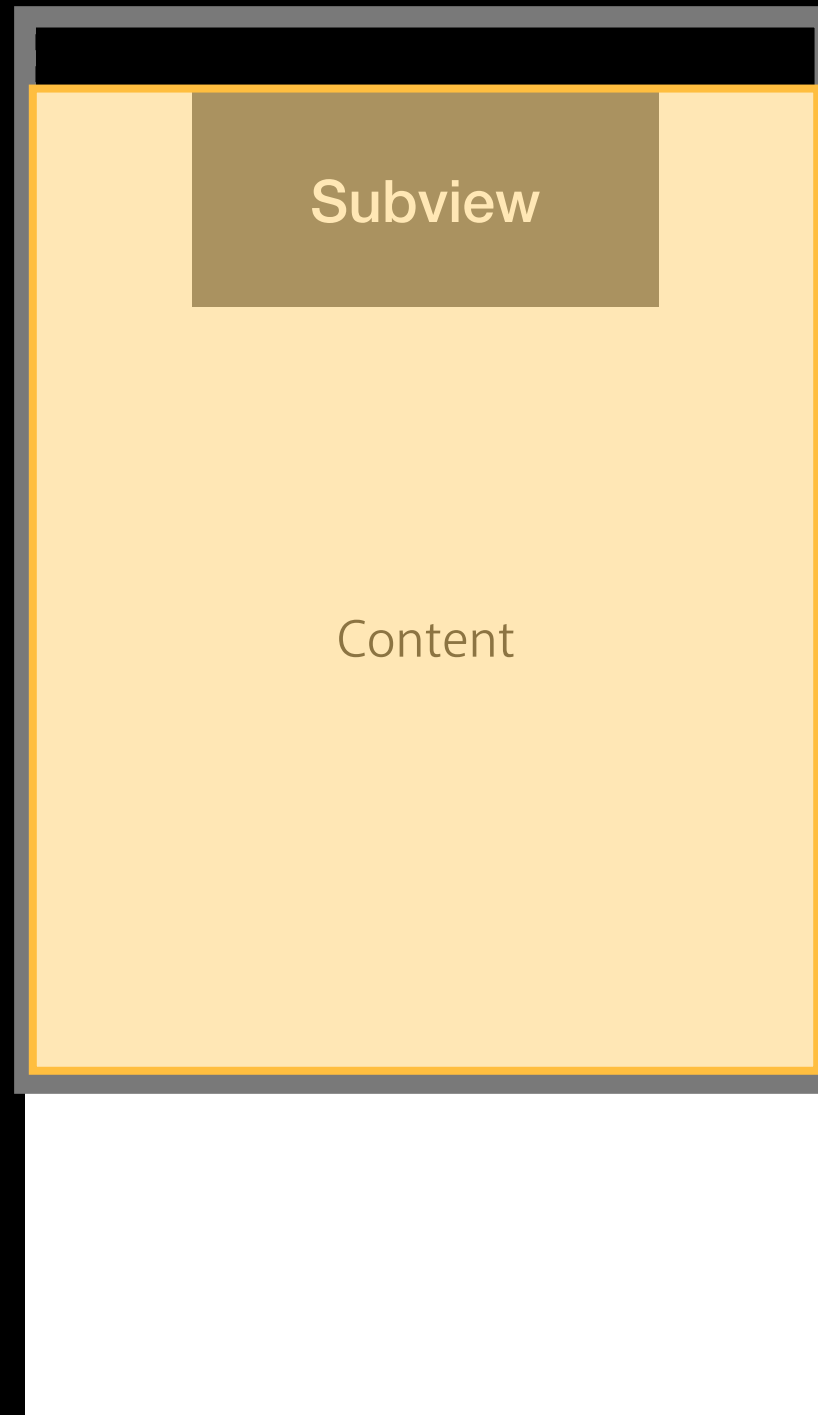
# ScrollView and Safe Area

`adjustedContentInset.top = 20`



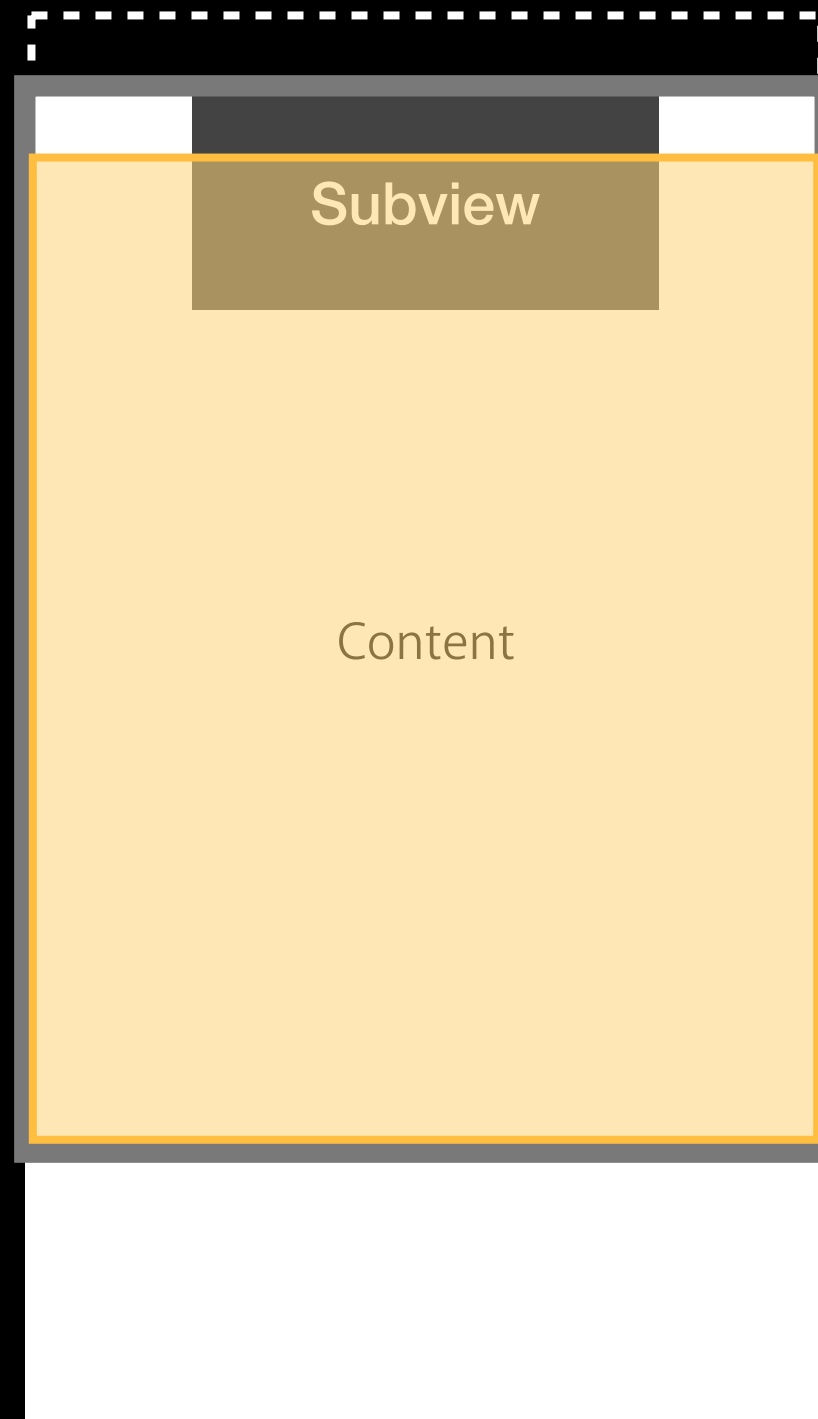
`safeAreaInsets.top = 20`

# ScrollView and Safe Area



`safeAreaInsets.top = 20`

# Scroll View and Safe Area



```
safeAreaInsets.top = 20
```

# Content Inset Adjustment Behaviour

```
class UIScrollView: UIView {  
    var contentInsetAdjustmentBehavior: ContentInsetAdjustmentBehavior  
  
    enum ContentInsetAdjustmentBehavior: Int {  
        case .always  
        case .scrollableAxes  
        case .automatic  
        case .never  
    }  
}
```

# Demo - Content Inset Adjustment Behaviour

1. Walkthrough of all options
2. Pitfalls



**.always**

Unexpected System Insets



Extra **adjustedContentInset**



Extra Scrollable Area



Extra Axis to Scroll

# .scrollableAxes

Insufficient ContentSize



Not Scrollable



No Adjustment



Unexpected Overlap

`.automatic`

This is the one!

Be careful with the deprecated

`automaticallyAdjustsScrollViewInsets`

**.never**

No Adjustment at All



Safe Area Propagation



Unexpected Layout Change on Scroll

# API Checklist (UIScrollView)

## UIViewController

`.automaticallyAdjustsScrollViewInsets`

## UIScrollView

`.bounds`

`.contentSize`

`.contentOffset`

`.contentInset`

`.adjustedContentInset`

`.contentInsetAdjustmentBehavior`

# API Checklist

## UIView

- .safeAreaInsets
- .safeAreaLayoutGuide
- .safeAreaInsetsDidChange()
- .layoutMargins
- .layoutMarginsGuide
- .directionalLayoutMargins
- .layoutMarginsDidChange()
- .insetsLayoutMarginsFromSafeArea
- .preserveSuperviewLayoutMargins

# API Checklist

## UIViewController

- .additionalSafeAreaInsets
- .viewSafeAreaInsetsDidChange()
- .viewLayoutMarginsDidChange()
- .systemMinimumLayoutMargins
- .viewRespectsSystemMinimumLayoutMargins
- .automaticallyAdjustsScrollViewInsets

# API Checklist

## UIScrollView

- .bounds
- .contentSize
- .contentOffset
- .contentInset
- .adjustedContentInset
- .contentInsetAdjustmentBehavior



# What's Next about Adaptive Layouts

- Readable Content Layout Guide
- UITableView and Safe Area
- Sizing Classes
- Dynamic Type of Fonts

# Slides and Demo Download

- <https://github.com/null09264/Adaptive-Layout-Demo/raw/master/Slides.pdf>
- <https://github.com/null09264/Adaptive-Layout-Demo>

## References

- <https://developer.apple.com/videos/play/wwdc2018/235/>
- <https://developer.apple.com/design/human-interface-guidelines/ios/visual-design/adaptivity-and-layout/>

# We are hiring!

<https://www.seagroup.com/career>

Thanks!