

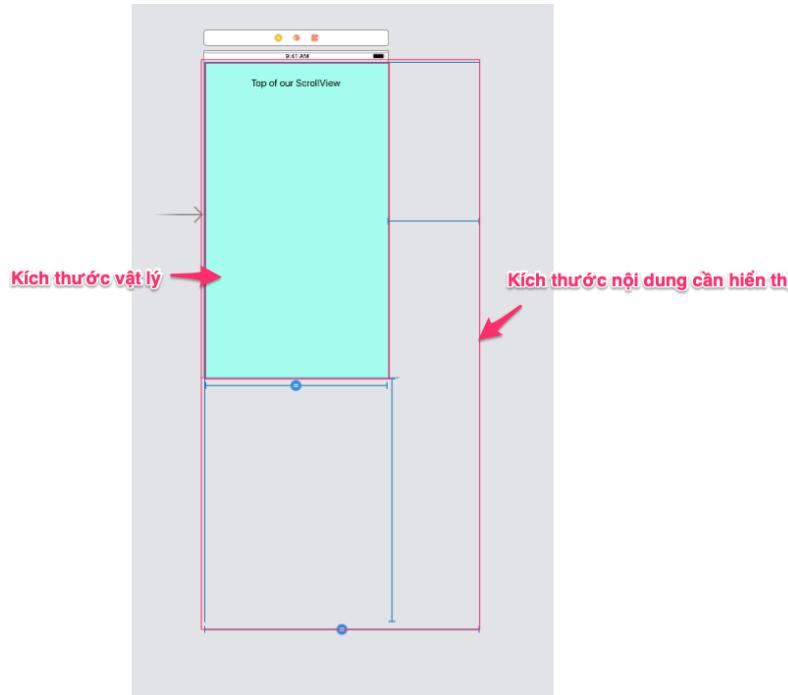
---

# UIScrollView

Cài đặt và lập trình căn bản

# Giới thiệu về UIScrollView

- UIScrollView cho phép hiển thị nội dung có kích thước lớn hơn kích thước vật lý của màn hình bằng cách cuộn hoặc zoom các đối tượng bên trong nó

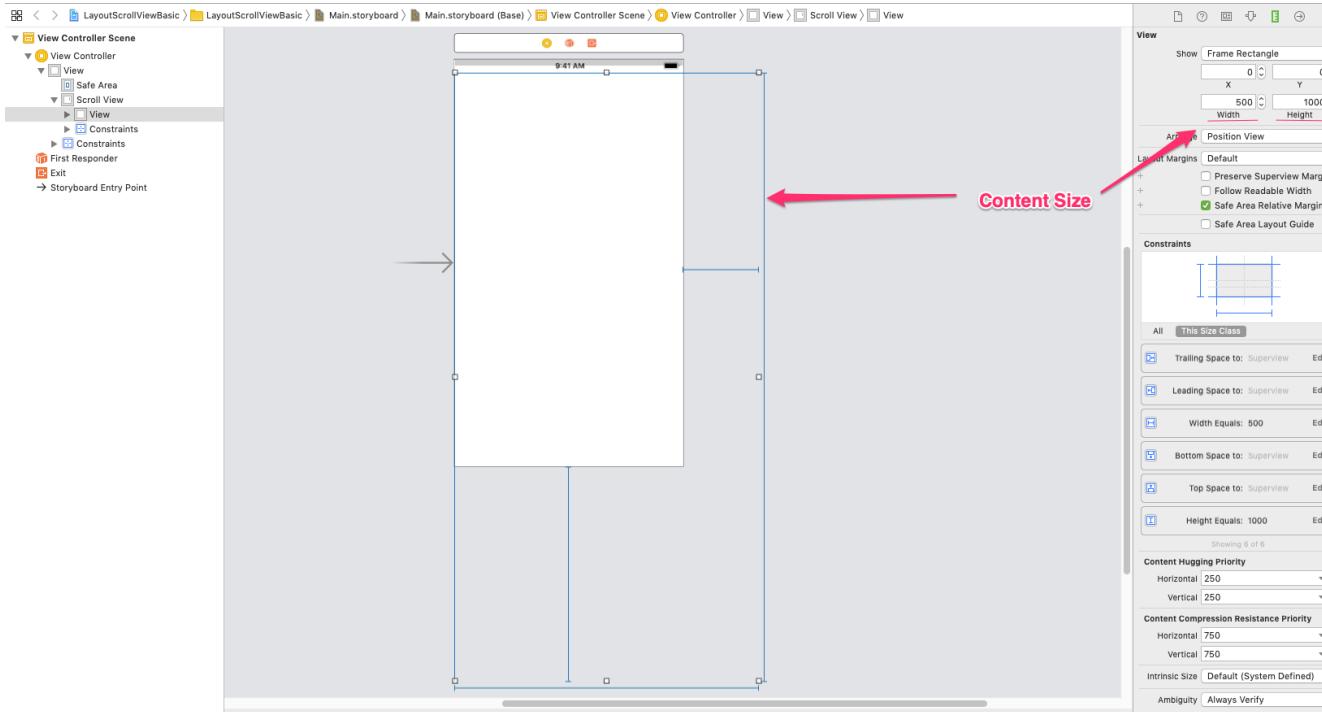


# Content Size, Content Offset, và Content Inset

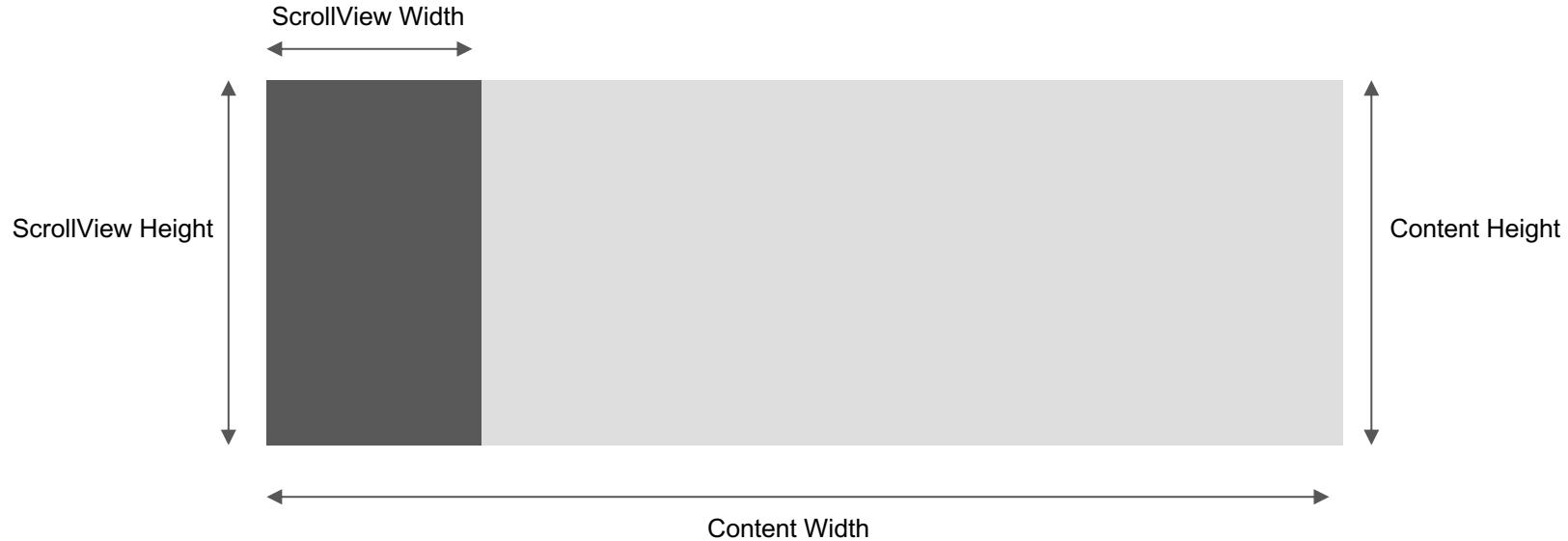


# Content Size

**Content Size:** là kích thước nội dung mà chúng ta muốn đặt trong chế độ cuộn (hoặc zoom) của UIScrollView.



## ScrollView theo chiều ngang:

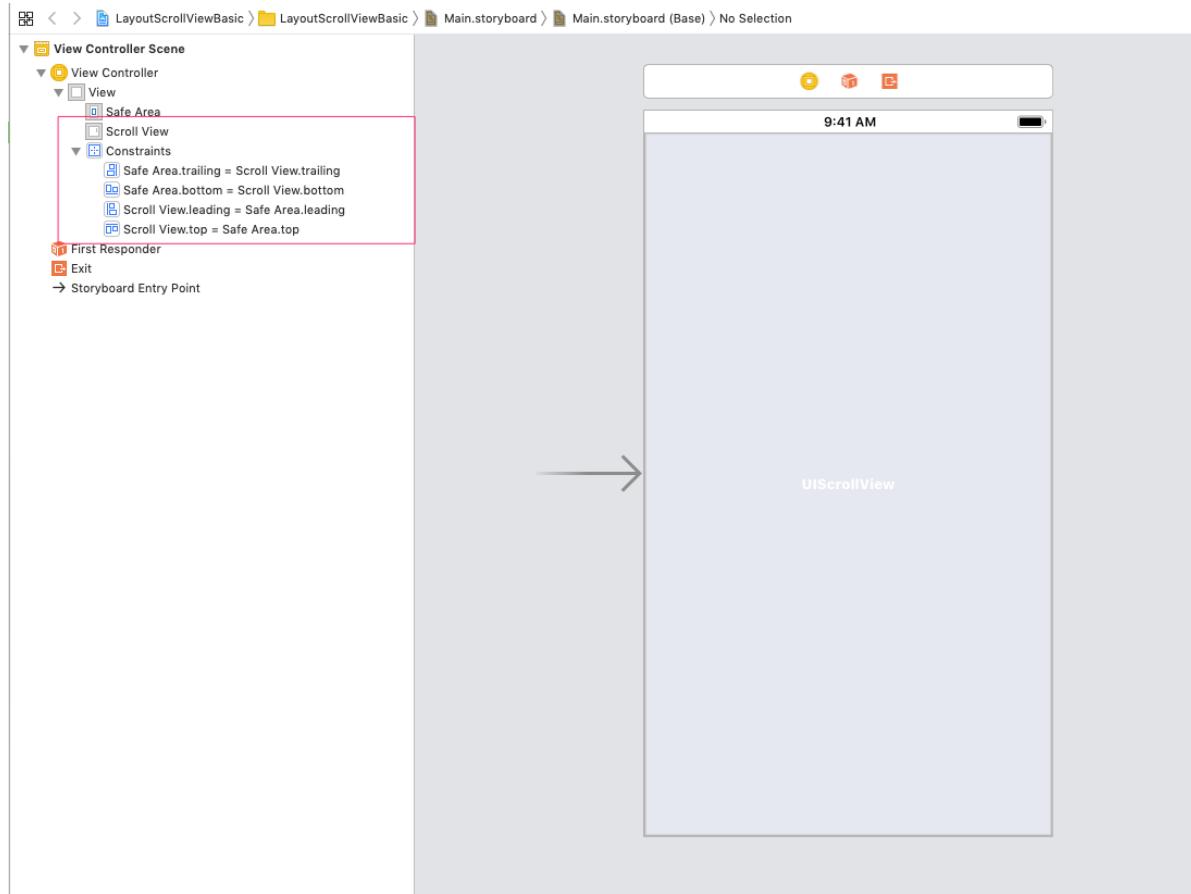


# Quản lý Content Size từ giao diện (Layout)

---

- B1: Thêm một UIScrollView vào ViewController và thêm các constraints: leading, trailing, top, bottom đều set là 0
- B2: Thêm một UIView duy nhất vào UIScrollView, view này sẽ chứa tất cả các phần tử khác. Đặt tên là ContentView  
Thiết lập constraints cho ContentView:
  - + leading, trailing, top, bottom = 0
  - + height, width = height, width của UIScrollView
  - + set priority của height là low (độ ưu tiên thấp hơn)
- B3: Thêm tất cả các constraints cho các phần tử trong ContentView, height – width của ContentView chính là contentSize của scrollView

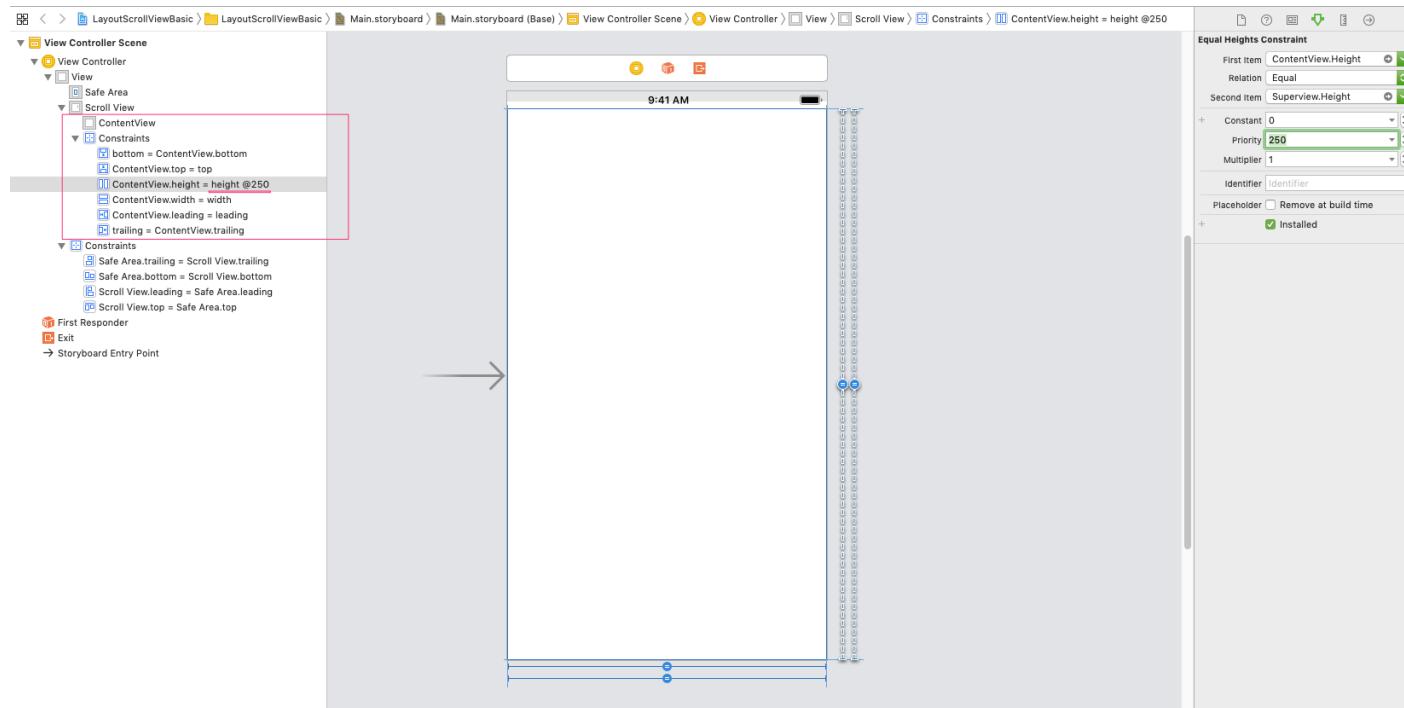
B1: Thêm một UIScrollView vào ViewController và thêm các constraints: leading, trailing, top, bottom đều set là 0



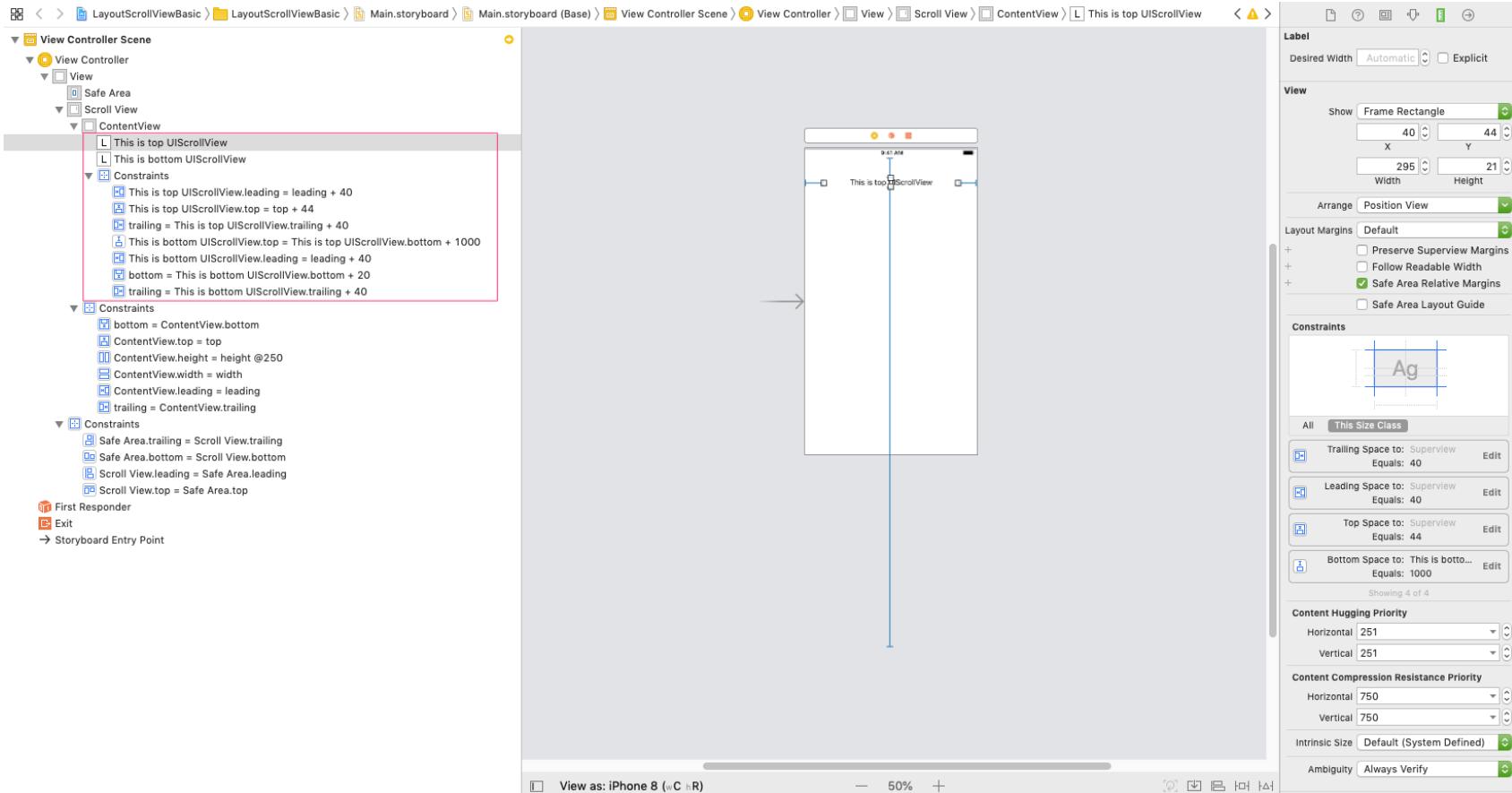
B2: Thêm một UIView duy nhất vào UIScrollView, view này sẽ chứa tất cả các phần tử khác. Đặt tên là ContentView

Thiết lập constraints cho ContentView:

- + leading, trailing, top, bottom = 0
- + height, width = height, width của UIScrollView
- + set priority của height là low (độ ưu tiên thấp hơn)

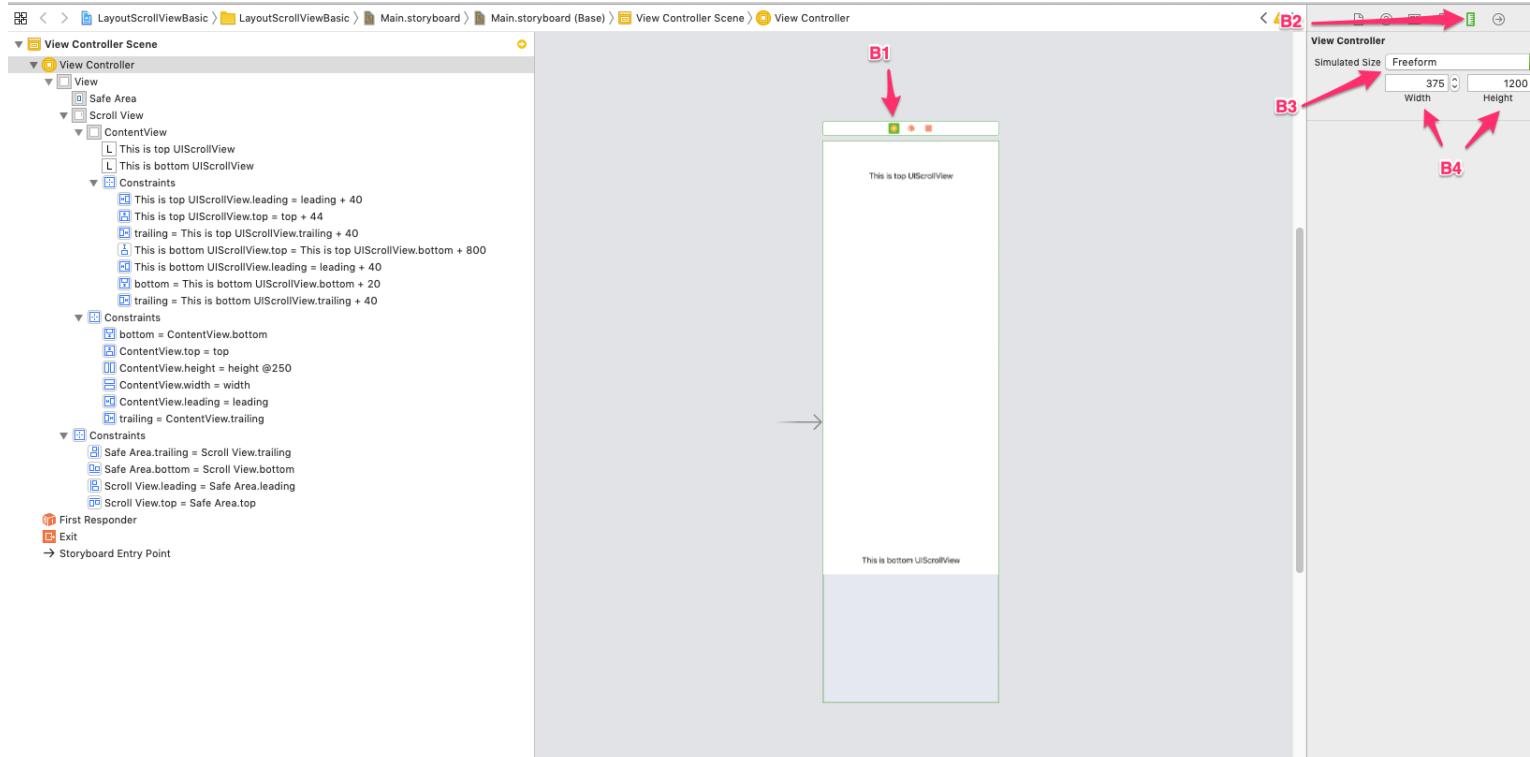


### B3: Thêm tất cả các constraints cho các phần tử trong ContentView, height – width của ContentView chính là contentSize của scrollView



Từ giao diện, để quản lý khung nhìn được tốt hơn, chúng ta có thể đặt kích thước cho thiết bị thành freeform:

*Chọn View Controller -> Show the size inspector -> Simulated Size -> Freeform -> đặt kích thước tùy biến*



# Quản lý Content Size bằng code

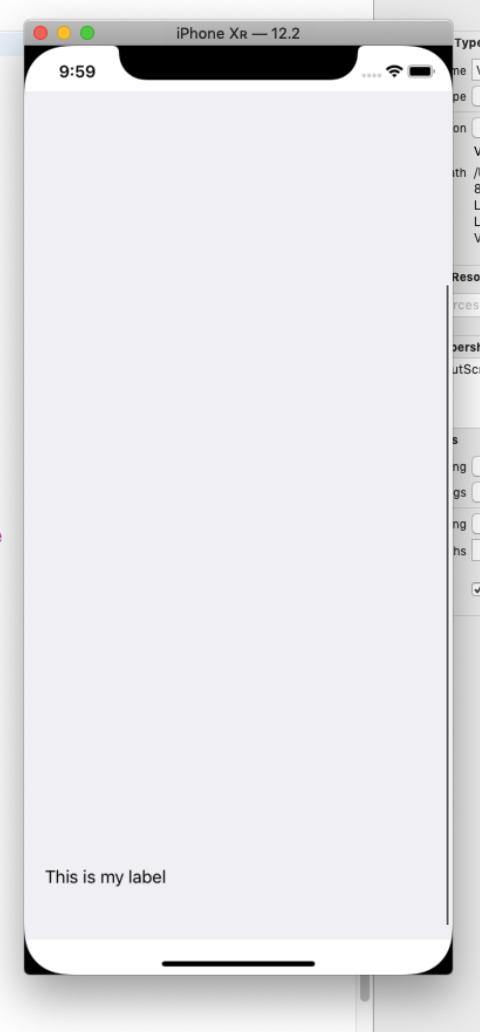
---

```
// set content Size cho UIScrollView  
scrollView.contentSize = CGSize(width: self.view.frame.width, height: 1000)
```

- Để thấy thay đổi, hãy add thử một control nào đó, ví dụ add vào một UILabel

```
let myLabel = UILabel()  
myLabel.text = "This is my label"  
myLabel.translatesAutoresizingMaskIntoConstraints = false  
scrollView.addSubview(myLabel)  
  
myLabel.topAnchor.constraint(equalTo: self.scrollView.topAnchor, constant: 800).isActive = true  
myLabel.leftAnchor.constraint(equalTo: self.scrollView.leftAnchor, constant: 20).isActive = true  
myLabel.rightAnchor.constraint(equalTo: self.scrollView.rightAnchor, constant: -20).isActive = true  
myLabel.bottomAnchor.constraint(equalTo: self.scrollView.bottomAnchor, constant: -50).isActive = true
```

```
1
2
3 import UIKit
4
5 class ViewController: UIViewController {
6
7     @IBOutlet weak var scrollView: UIScrollView!
8
9     override func viewDidLoad() {
10         super.viewDidLoad()
11
12         scrollView.contentSize = CGSize(width: self.view.frame.width, height: 1000)
13         scrollView.showsVerticalScrollIndicator = true
14         let myLabel = UILabel()
15         myLabel.text = "This is my label"
16         myLabel.translatesAutoresizingMaskIntoConstraints = false
17         scrollView.addSubview(myLabel)
18
19         myLabel.topAnchor.constraint(equalTo: self.scrollView.topAnchor, constant: 800).isActive = true
20         myLabel.leftAnchor.constraint(equalTo: self.scrollView.leftAnchor, constant: 20).isActive = true
21         myLabel.rightAnchor.constraint(equalTo: self.scrollView.rightAnchor, constant: -20).isActive = true
22         myLabel.bottomAnchor.constraint(equalTo: self.scrollView.bottomAnchor, constant: -50).isActive = true
23     }
24
25 }
26
27 }
```

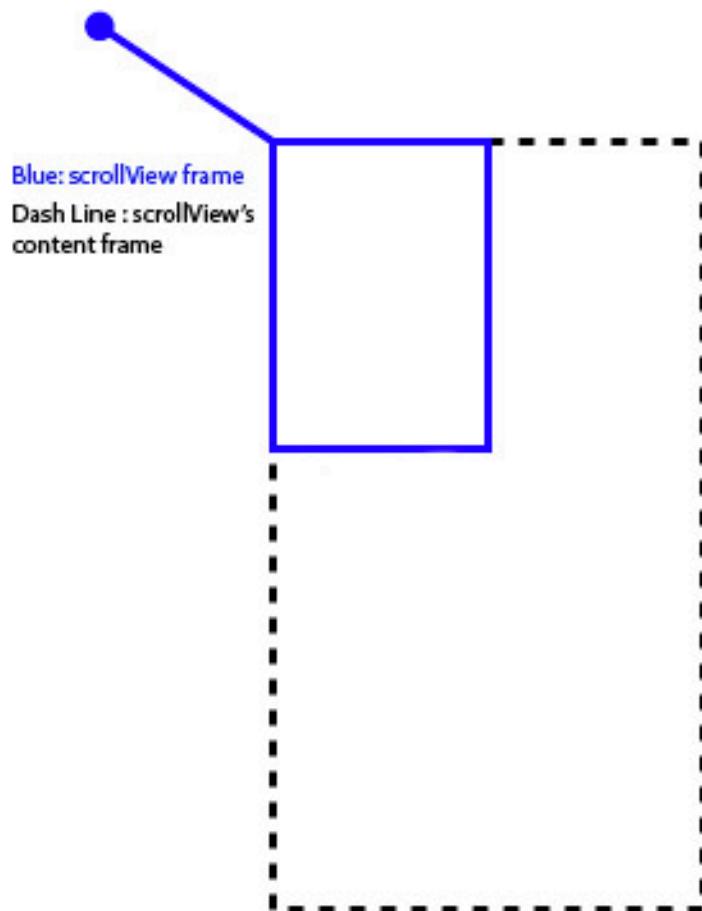


# Content Offset

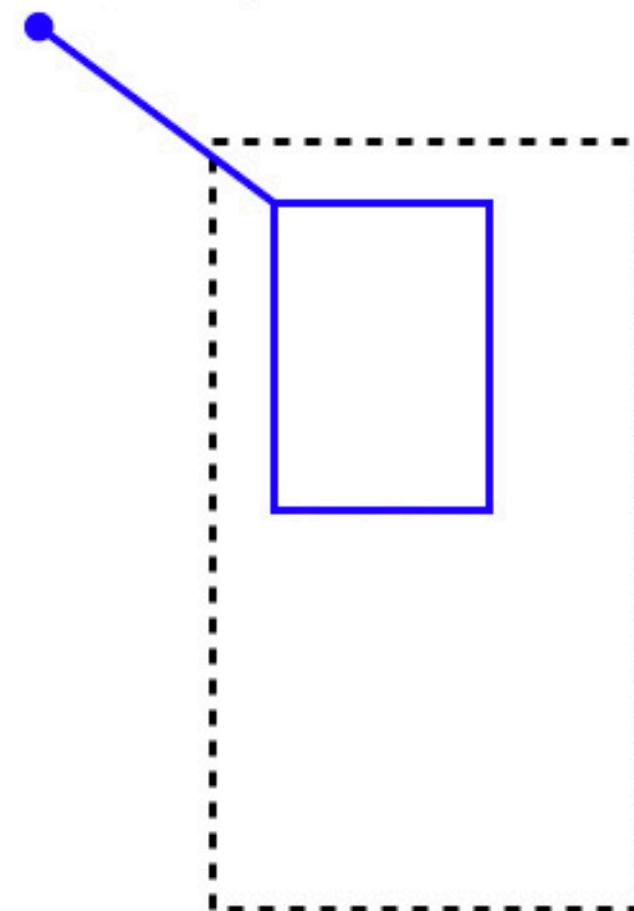
---

- Khi người dùng cuộn chẽ độ xem cuộn, vùng nội dung hiển thị cho người dùng sẽ thay đổi. Vị trí của chẽ độ xem nội dung trong chẽ độ xem cuộn được gọi là phần bù nội dung và được biểu thị bằng thuộc tính contentOffset, giá trị CGPoint.
- Thuộc tính này được xác định bởi khoảng cách với các giá trị x và y từ gốc vùng nhìn thấy được (điểm trên cùng bên trái) đến chẽ độ xem nội dung

**contentOffset= 0, 0**

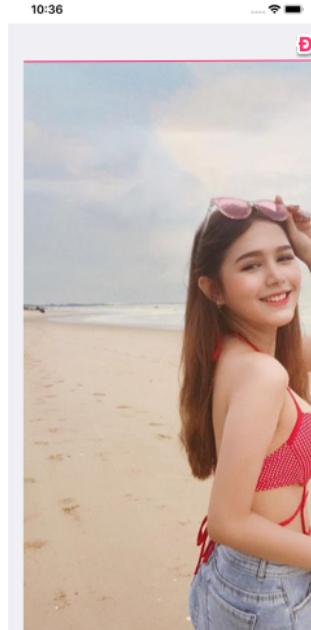


**contentOffset= 10, 10**



- Chế độ contentOffset có thể được thay đổi, ví dụ bạn có thể sử dụng phương thức: **func setContent Offset (CGPoint, animated: Bool)** để tạo hiệu ứng thay đổi. Điều này giúp di chuyển các chế độ xem cuộn, giống như khi người dùng tự di chuyển.

```
scrollView.setContentOffset(CGPoint(x: 20, y: 100), animated: true)
```



Không setContentOffset

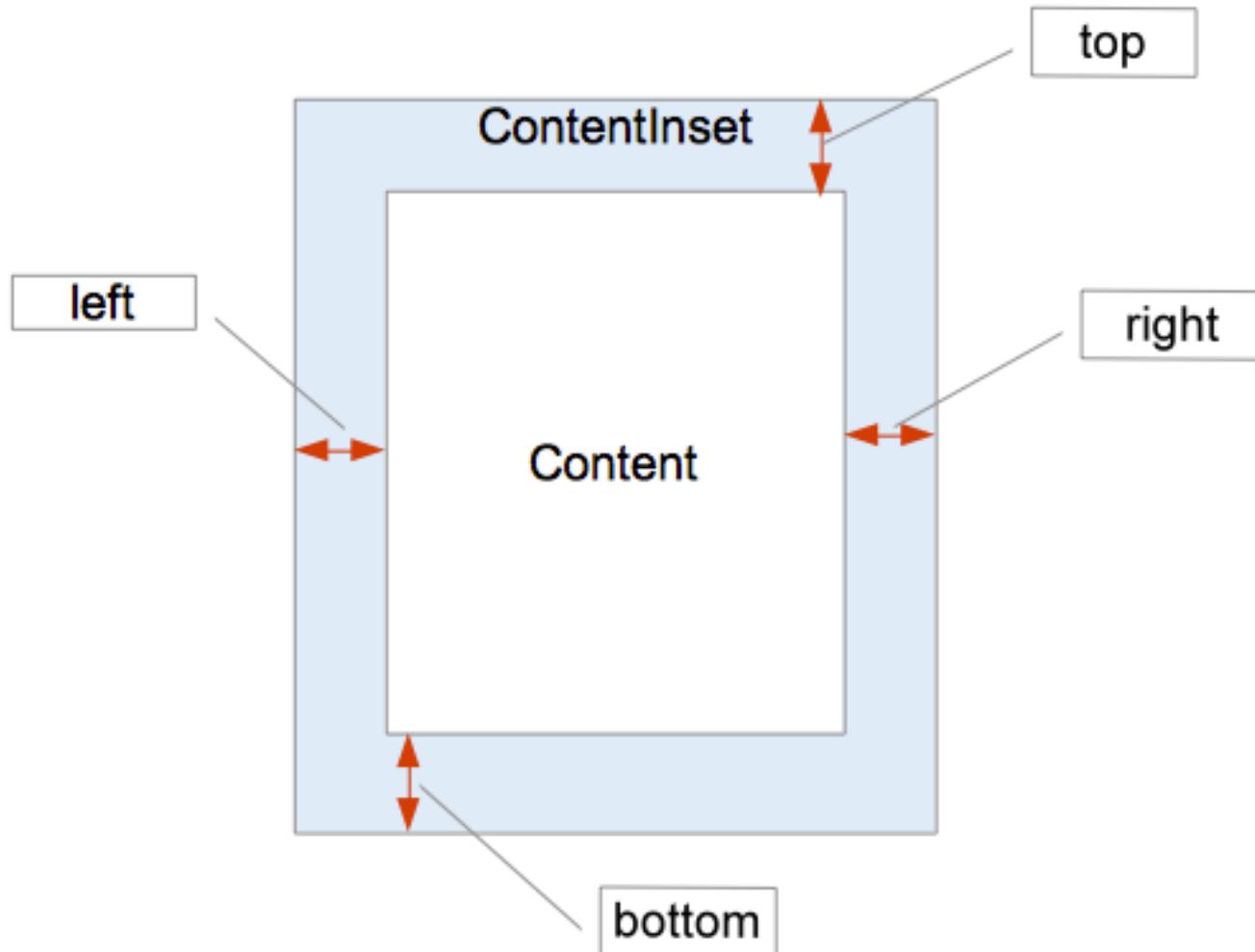


Có setContentOffset

# Content Inset

---

- Content Inset là một giá trị UIEdgeInsets đại diện cho khu vực xung quanh chế độ xem cuộn nội dung.
- Đặt thuộc tính contentInset thành UIEdgeInsetsMake (10, 10, 10, 10) sẽ tạo ra một lề 10 điểm xung quanh nội cuộn. Các giá trị chèn cạnh cũng có thể âm; điều này sẽ đại diện cho khu vực xung quanh nội dung của chế độ xem cuộn mà người dùng có thể nhìn thấy (trừ khi người dùng cuộn qua mép của chế độ xem cuộn).



# Một số thuộc tính của UIScrollView

---

- Để khoá chế độ cuộn của UIScrollView (mặc định là true):

```
scroll.isEnabled = false
```

- Để khi cuộn sẽ thành phân trang trong UIScrollView:

```
scroll.isPagingEnabled = true
```

- Để khi cuộn scroll view không bị nẩy quá kích thước Content Size, chúng ta sử dụng thuộc tính bounces:

```
scrollView.bounces = false
```

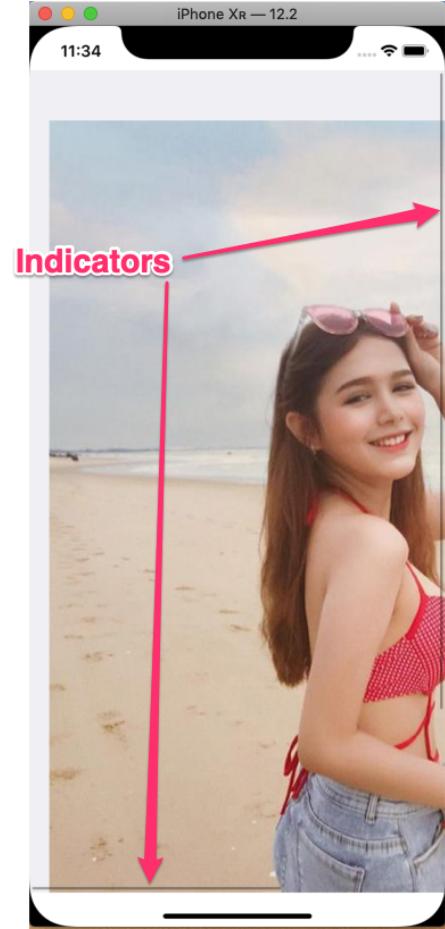
# Scroll Indicator



# Indicator là gì

---

- Indicator của UIScrollView hiển thị khi chế độ xem nội dung của ScrollView lớn hơn ScrollView và người dùng cuộn nó trên Màn hình.
- Để hiển thị Indicator, chúng ta ngoài việc tương tác trực tiếp với UIScrollView thì có thể gọi hàm flashScrollIndicator(), tuy nhiên hàm này sau khi hiển thị sẽ ngay lập tức biến mất



# Bật – tắt hiển thị indicator

---

- UIScrollView cho phép chúng ta cuộn 2 chiều, để không muốn hiển thị thanh cuộn Indicator chúng ta có thể tắt nó:

+ Tắt thanh cuộn chiều dọc:

```
// giá trị false sẽ tắt hiển thị thanh indicator  
scrollView.showsVerticalScrollIndicator = false
```

+ Tắt thanh cuộn chiều ngang:

```
// giá trị false sẽ tắt hiển thị thanh indicator  
scrollView.showsHorizontalScrollIndicator = false
```

# Indicator styles

---

- Chúng ta có thể thay đổi styles của thanh indicator:

```
scrollView.indicatorStyle = .black
```

- indicatorStyle có 3 giá trị là : black, default, white

# **UIScrollView Protocol (UIScrollView Delegate)**



- Các method được khai báo bởi giao thức UIScrollViewDelegate cho phép chấp nhận và phản hồi các tin nhắn từ lớp UIScrollView và do đó tác động đến các hoạt động như cuộn, phóng to, giảm tốc độ nội dung cuộn và cuộn hình ảnh động.
- Cách sử dụng UIScrollViewDelegate:
  - 1, Phân kế thừa class ViewController, thêm UIScrollViewDelegate
  - 2, Gán delegate cho UIScrollView trong viewDidLoad():  
    scrollView.delegate = self
  - 3, Gọi hàm trong delegate mà chúng ta cần sử dụng

```
8
9 import UIKit
10
11 class ViewController: UIViewController, UIScrollViewDelegate {
12
13     // Khởi tạo một scroll view
14     lazy var scrollView: UIScrollView = {
15         let scrollView = UIScrollView()
16         scrollView.translatesAutoresizingMaskIntoConstraints = false
17         scrollView.contentSize.height = 2000
18         scrollView.backgroundColor = UIColor.brown
19         return scrollView
20     }()
21
22     override func viewDidLoad() {
23         super.viewDidLoad()
24
25         view.addSubview(scrollView)
26         setupScrollView()
27
28         scrollView.delegate = self
29     }
30
```

# Cuộn (scroll) và kéo (drag) trong UIScrollView

---

## **func scrollViewDidScroll(\_ scrollView: UIScrollView):**

Báo cho delegate khi người dùng bắt đầu cuộn, và được gọi khi độ lệch nội dung của chế độ xem thay đổi

## **func scrollViewWillBeginDragging(\_ scrollView: UIScrollView):**

Được gọi bất cứ khi nào scrollView sắp được kéo bởi người dùng

## **func scrollViewWillEndDragging(\_ scrollView: UIScrollView, withVelocity velocity: CGPoint, targetContentOffset: UnsafeMutablePointer<CGPoint>):**

Được gọi bất cứ khi nào người dùng nhấc ngón tay khỏi chế độ xem cuộn sau khi kéo.

**func scrollViewDidEndDragging(\_ scrollView: UIScrollView, willDecelerate decelerate: Bool):**

**func scrollViewShouldScrollToTop(\_ scrollView: UIScrollView) -> Bool:**

Chế độ xem cuộn **sẽ** cuộn lên đầu nội dung

**func scrollViewDidScrollToTop(\_ scrollView: UIScrollView):**

Chế độ xem cuộn **đã** cuộn lên trên cùng của nội dung.

**func scrollViewWillBeginDecelerating(\_ scrollView: UIScrollView):**

Chế độ xem cuộn đang bắt đầu giảm tốc độ di chuyển. Được gọi bất cứ khi nào chế độ xem cuộn sắp bắt đầu animation giảm tốc

**func scrollViewDidEndDecelerating(\_ scrollView: UIScrollView)**

Chế độ xem cuộn **đã** kết thúc giảm tốc độ di chuyển

# Scrolling animation

---

**func scrollViewDidEndScrollingAnimation(\_ scrollView:  
UIScrollView):**

Được gọi khi chế độ xem cuộn kết thúc

# **Zooming**

---

**func viewForZooming(in scrollView: UIScrollView) -> UIView?:**

Chế độ xem theo tỷ lệ khi thu phóng sắp xảy ra trong chế độ xem cuộn.

**func scrollViewWillBeginZooming(\_ scrollView: UIScrollView, with view: UIView?):**

Việc phóng to nội dung trong chế độ xem cuộn sắp bắt đầu.

**func scrollViewDidZoom(\_ scrollView: UIScrollView):**

Thông báo khi hệ số thu phóng của chế độ xem cuộn thay đổi.

**func scrollViewDidEndZooming(\_ scrollView: UIScrollView,  
with view: UIView?, atScale scale: CGFloat):**

Thông báo khi phóng to nội dung trong chế độ xem cuộn hoàn thành.

# Zoom ảnh sử dụng UIScrollView



## B1: Thêm scrollView và imageView từ giao diện (layout) và kéo outlet

The screenshot shows the Xcode interface with two main panes: the storyboard on the left and the code editor on the right.

**Storyboard (Left Pane):**

- Project navigation bar: ZoomImage, ZoomImage, Main.storyboard, Main.storyboard (Base), No Selection.
- File list: ZoomImage (group), AppDelegate.swift, ViewController.swift, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, Info.plist, Products.
- View Controller Scene: View Controller (selected). It contains a View which has a Safe Area and a Scroll View. The Scroll View contains an Image View. The constraints for the Image View are highlighted with a red box:
  - height = 259
  - centerX = centerX
  - leading = leading
  - top = top + 150
  - bottom = Image View.bottom + 144
  - trailing = Image View.trailing
- Other components: First Responder, Exit, Storyboard Entry Point.

**Code Editor (Right Pane):**

```
1 // ViewController.swift
2 // ZoomImage
3 // Created by Taof on 8/14/19.
4 //
5 // Copyright © 2019 Taof. All rights reserved.
6 //
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController{
12
13     @IBOutlet weak var scrollView: UIScrollView!
14     @IBOutlet weak var imageView: UIImageView!
15
16     override func viewDidLoad() {
17         super.viewDidLoad()
18
19     }
20
21 }
22
23
24 }
```

A red box highlights the outlets `scrollView` and `imageView` in the code editor.

## B2: Class ViewController cần kế thừa UIScrollViewDelegate

```
9 import UIKit
10
11 class ViewController: UIViewController, UIScrollViewDelegate {
12
13     @IBOutlet weak var scrollView: UIScrollView!
14     @IBOutlet weak var imageView: UIImageView!
15
16     override func viewDidLoad() {
17         super.viewDidLoad()
18
19     }
20
21
22 }
23
24
```

Gán thuộc tính delegate của scrollView cho ViewController

```
10
11 class ViewController: UIViewController, UIScrollViewDelegate {
12
13     @IBOutlet weak var scrollView: UIScrollView!
14     @IBOutlet weak var imageView: UIImageView!
15
16     override func viewDidLoad() {
17         super.viewDidLoad()
18
19         scrollView.delegate = self
20     }
21
22
23 }
24
```

### B3: Gọi hàm hứng sự kiện zooming của scrollView

```
func viewForZooming(in scrollView: UIScrollView) -> UIView? {  
    return imageView  
}
```

```
11 class ViewController: UIViewController, UIScrollViewDelegate {  
12  
13     @IBOutlet weak var scrollView: UIScrollView!  
14     @IBOutlet weak var imageView: UIImageView!  
15  
16     override func viewDidLoad() {  
17         super.viewDidLoad()  
18  
19         scrollView.delegate = self  
20     }  
21  
22     func viewForZooming(in scrollView: UIScrollView) -> UIView? {  
23         return imageView  
24     }  
25     |  
26 }  
27  
28
```

11:07



11:07



11:08



# **UIScrollView và UIPageControl**



# UIPageControl là gì?

---

**UIPageControl** là lớp cung cấp cho bạn một control dùng để quản lý các trang giao diện trong ứng dụng. Hình ảnh của control này là một dãy các dấu “●”, mỗi dấu “●” đại diện cho một trang trong ứng dụng.



# Các thuộc tính của UIPageControl

---

- Trang hiện tại:

```
pageControl.currentPage = 0
```

- Tổng số lượng trang:

```
pageControl.numberOfPages = 5
```

- Ẩn dấu chấm nếu chỉ có một trang duy nhất:

```
pageControl.hidesForSinglePage = true
```



- Màu sắc của các dấu chấm trên control:

```
pageControl.pageIndicatorTintColor = UIColor.lightGray
```



- Màu sắc dấu chấm của trang hiện tại:

```
pageControl.currentPageIndicatorTintColor = UIColor.purple
```



# Ví dụ kết hợp UIScrollView và UIPageControl

---

- Thay vì dùng UIPageViewController, chúng ta có thể sử dụng kết hợp UIScrollView và UIPageControl để tạo ra màn hình quản lý, giới thiệu các nội dung
- Sự kết hợp này được áp dụng nhiều trong thiết kế giới thiệu các chức năng của App cho người mới bắt đầu

B1: Thêm UIScrollView và UIPageControl từ giao diện (layout), kéo outlet cho 2 control này:

The image shows the Xcode interface with two main panes. On the left is the storyboard editor for 'Main.storyboard' showing a single view controller scene. Inside the scene, there is a scroll view labeled 'UIScrollView' and a page control labeled 'UIPageControl'. A red arrow points from the text 'UIScrollView' to the scroll view in the storyboard. Another red arrow points from the text 'UIPageControl' to the page control in the storyboard. A red box highlights the 'Layout' section in the storyboard's object library sidebar, with the text 'Layout' written above it in pink. On the right is the code editor for 'ViewController.swift'. A red box highlights the outlets 'scrollView' and 'pageControl' in the code, with the text 'Kéo outlet' written above it in pink. The code is as follows:

```
// ViewController.swift
// Created by Taof on 8/14/19.
// Copyright © 2019 Taof. All rights reserved.

import UIKit

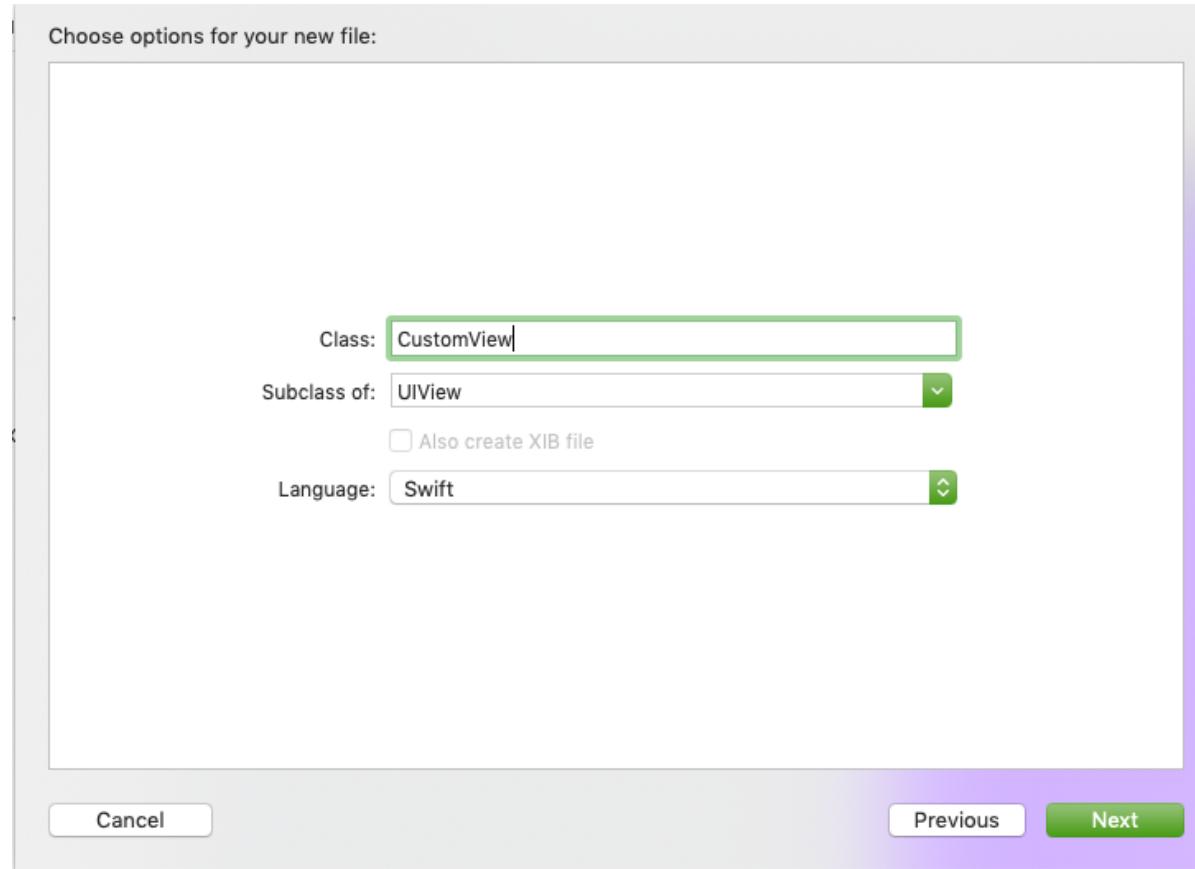
class ViewController: UIViewController {

    @IBOutlet weak var scrollView: UIScrollView!
    @IBOutlet weak var pageControl: UIPageControl!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
    }
}
```

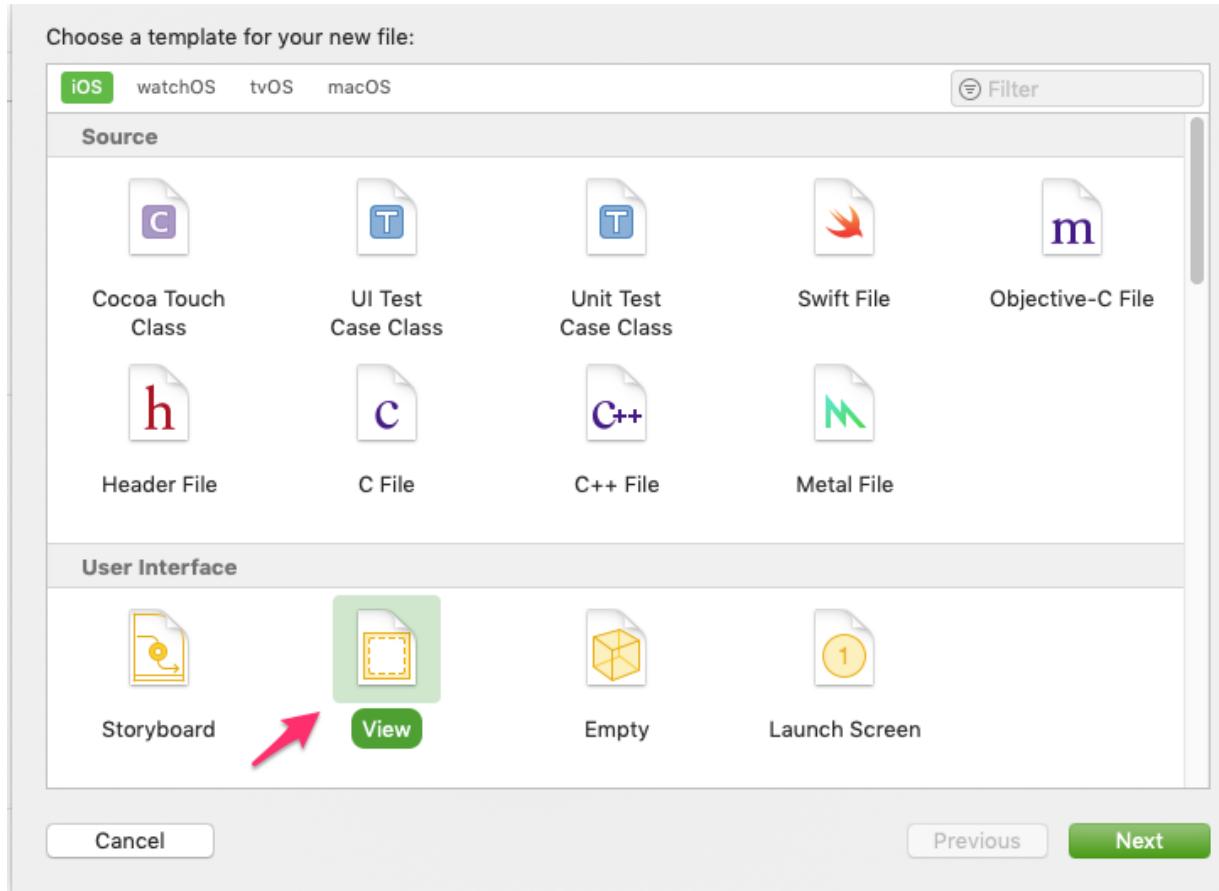
## B2: Custom một UIView:

- Tạo mới một file nguồn CocoaTouch Class, Subclass là UIView

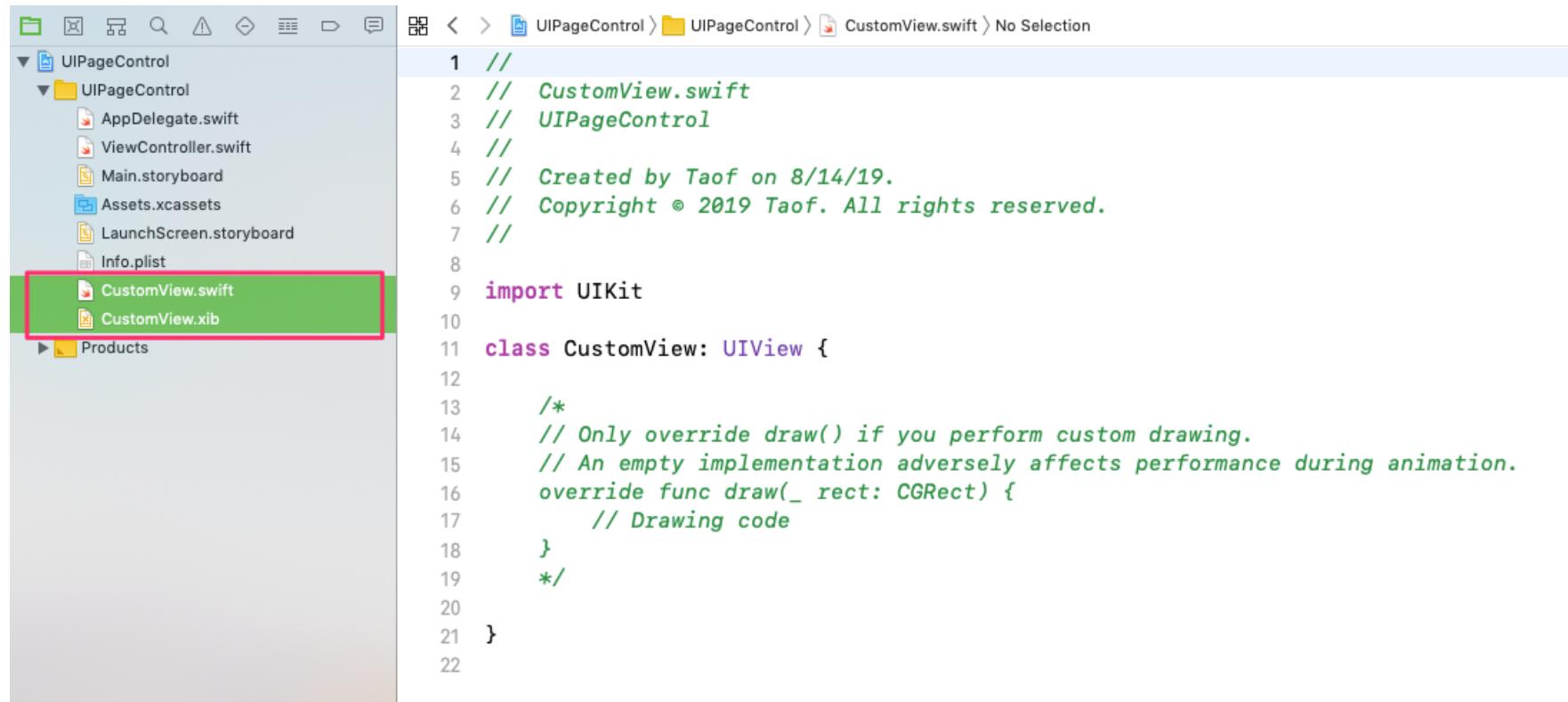


## Tiếp B2 (2.1)

- Tạo mới một file xib, đặt tên là CustomView



Sau khi tạo chúng ta sẽ được 2 file này:



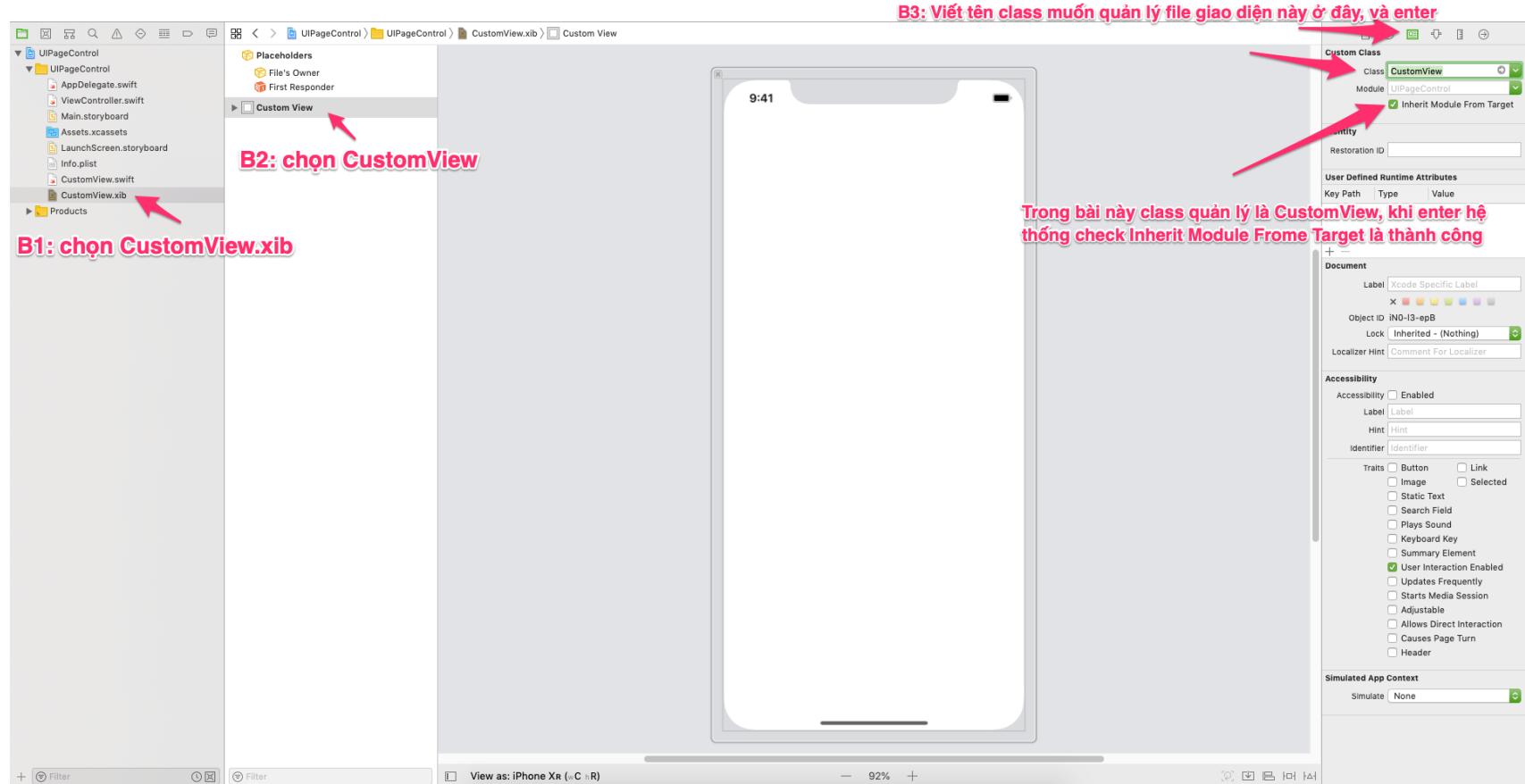
The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure under "UIPageControl". A red box highlights the "CustomView.swift" and "CustomView.xib" files, which are located in a folder named "UIPageControl".
- Editor:** Displays the content of "CustomView.swift". The code is as follows:

```
1 //  
2 //  CustomView.swift  
3 //  UIPageControl  
4 //  
5 //  Created by Taof on 8/14/19.  
6 //  Copyright © 2019 Taof. All rights reserved.  
7 //  
8  
9 import UIKit  
10  
11 class CustomView: UIView {  
12  
13     /*  
14     // Only override draw() if you perform custom drawing.  
15     // An empty implementation adversely affects performance during animation.  
16     override func draw(_ rect: CGRect) {  
17         // Drawing code  
18     }  
19     */  
20  
21 }
```

## Tiếp B2 (2.2)

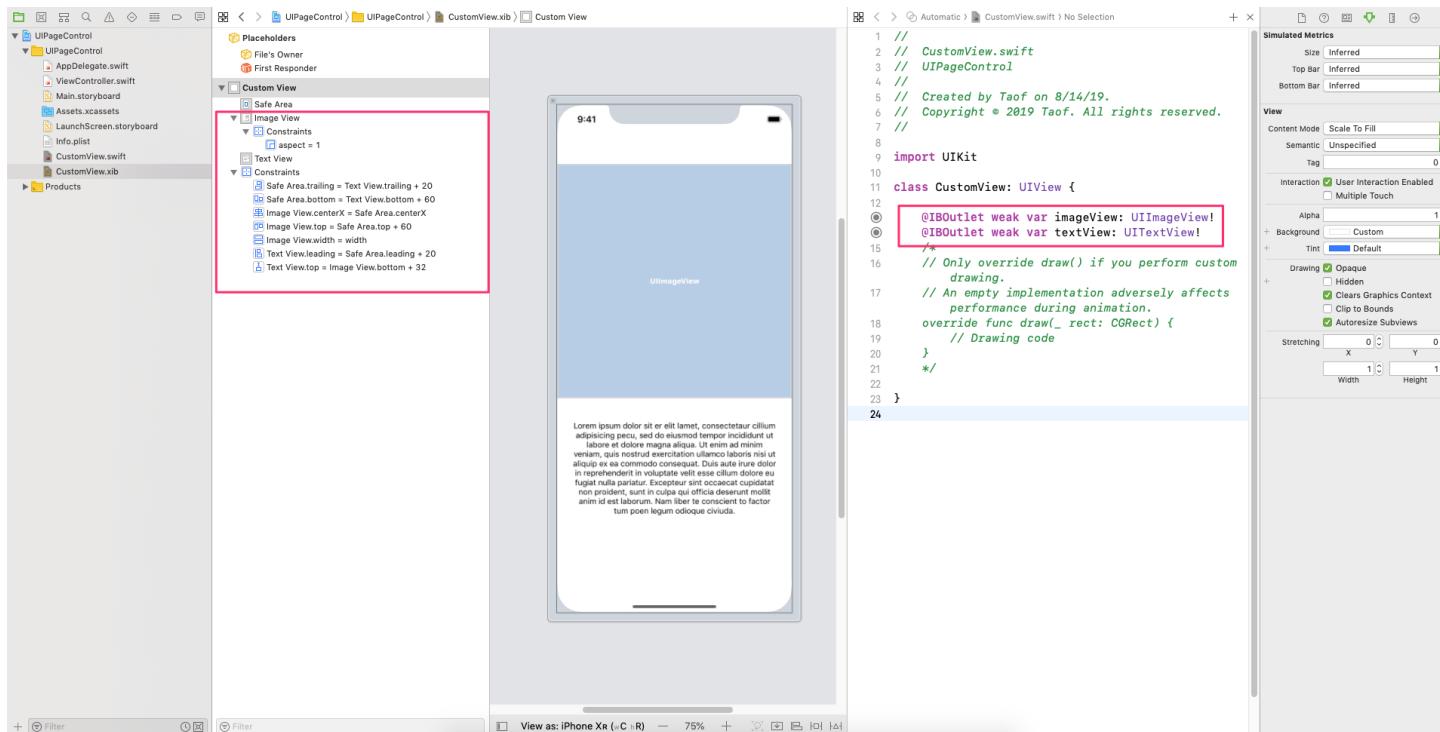
- Liên kết file quản lý code và file quản lý giao diện



## Tiếp B2 (2.3)

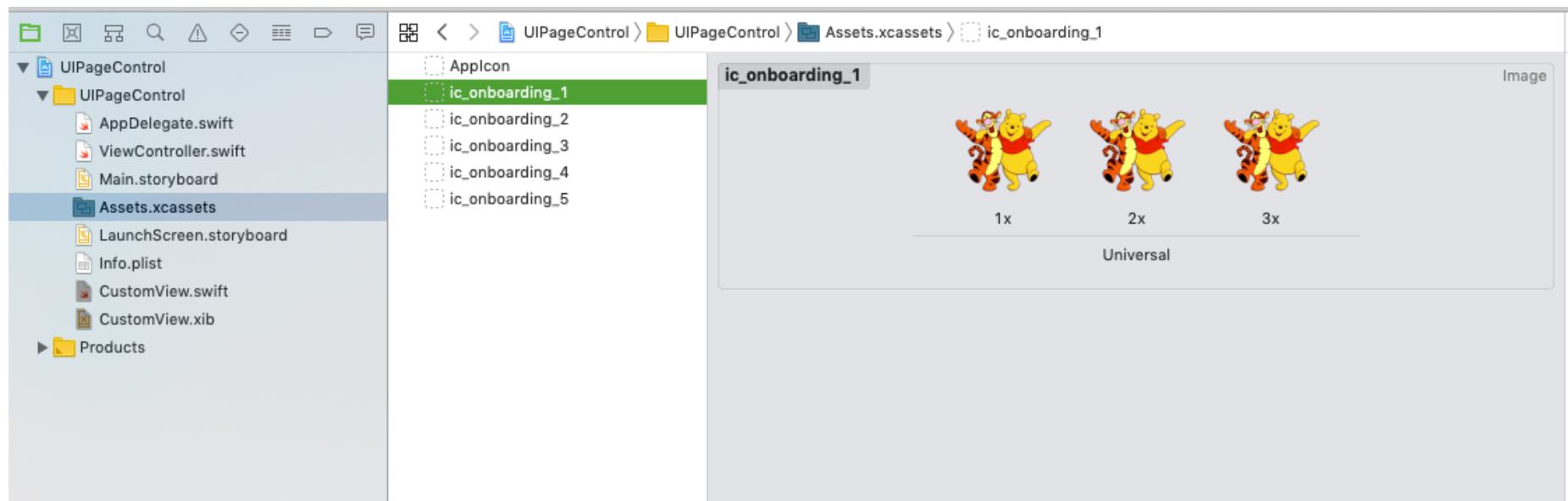
- Thêm các thành phần con vào giao diện CustomView và kéo outlet
- Layout các thành phần con:

UIImageView cách top 60, width = width màn hình, ratio = 1, center Horizontal  
UITextView cách đáy ImageView 32, cách trái – phải = 20, cách bottom = 60,



### B3: Tạo dữ liệu

- Thêm ảnh vào Assets



## Tiếp B3 (3.1)

- Vào file ViewController:

+ Khởi tạo một mảng chứa các customView

```
var views: [CustomView] = []
```

+ Viết hàm tạo dữ liệu:

```
func createData(){

    let data1: CustomView = Bundle.main.loadNibNamed("CustomView", owner: self, options: nil)?.first as! CustomView
    data1.imageView.image = UIImage(named: "ic_onboarding_1")
    data1.textView.text = "Did you know that Winnie the chubby little cubby was based on a real, young bear in London"

    let data2: CustomView = Bundle.main.loadNibNamed("CustomView", owner: self, options: nil)?.first as! CustomView
    data1.imageView.image = UIImage(named: "ic_onboarding_2")
    data1.textView.text = "Did you know that Winnie the chubby little cubby was based on a real, young bear in London"

    let data3: CustomView = Bundle.main.loadNibNamed("CustomView", owner: self, options: nil)?.first as! CustomView
    data1.imageView.image = UIImage(named: "ic_onboarding_3")
    data1.textView.text = "Did you know that Winnie the chubby little cubby was based on a real, young bear in London"

    let data4: CustomView = Bundle.main.loadNibNamed("CustomView", owner: self, options: nil)?.first as! CustomView
    data1.imageView.image = UIImage(named: "ic_onboarding_4")
    data1.textView.text = "Did you know that Winnie the chubby little cubby was based on a real, young bear in London"

    let data5: CustomView = Bundle.main.loadNibNamed("CustomView", owner: self, options: nil)?.first as! CustomView
    data1.imageView.image = UIImage(named: "ic_onboarding_5")
    data1.textView.text = "Did you know that Winnie the chubby little cubby was based on a real, young bear in London"

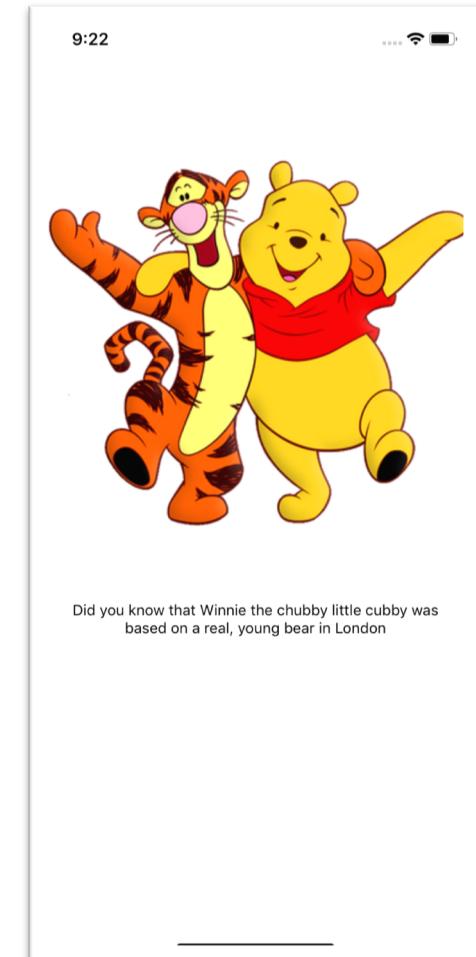
    views = [data1, data2, data3, data4, data5]
}
```

+ Gọi hàm createData() lên viewDidLoad:

B4: Viết hàm thêm customView vào UIScrollView, và gọi hàm này lên viewDidLoad

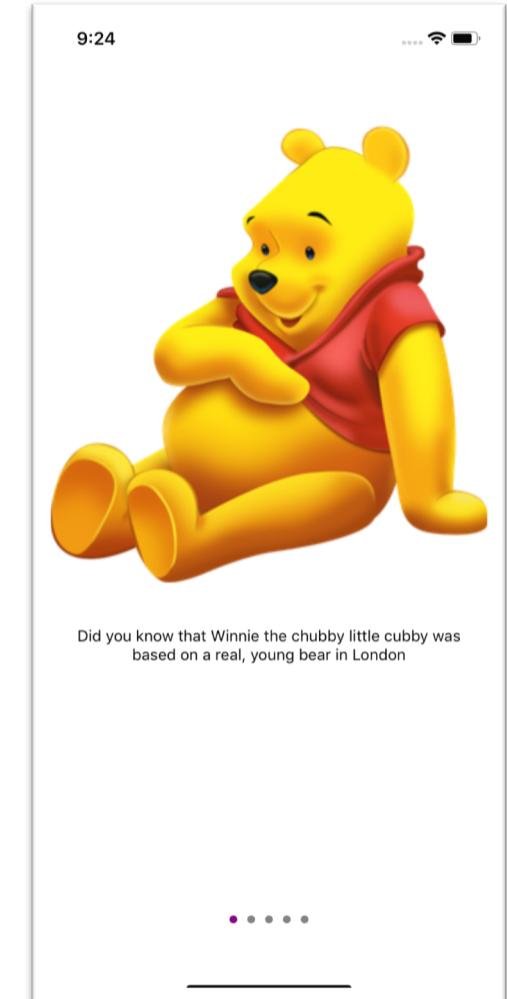
```
// add Slide vào scrollView
func setupSlideScrollView(slides: [Slide]){
    scrollView.frame = CGRect(x: 0, y: 0, width: screenWidth,
height: screenHeight - 100)
    scrollView.contentSize = CGSize(width: screenWidth *
CGFloat(slides.count), height: screenHeight - 100)
    scrollView.isPagingEnabled = true

    for i in 0 ..< slides.count {
        slides[i].frame = CGRect(x: screenWidth * CGFloat(i), y:
0, width: screenWidth, height: screenHeight - 100)
        scrollView.addSubview(slides[i])
    }
}
```



B5: Cấu hình ban đầu cho UIPageControl (viết ở viewDidLoad):

```
// cấu hình pageControl  
pageControl.currentPage = 0  
pageControl.numberOfPages = views.count  
pageControl.pageIndicatorTintColor = UIColor.gray  
pageControl.currentPageIndicatorTintColor = UIColor.purple
```



B6: Để dẫu chấm của PageControl thay đổi theo các trang thì chúng ta viết gọi delegate của UIScrollView (chúng ta có thể viết tách ra thành extension)

Trong viewDidLoad:

```
scrollView.delegate = self
```

Ngoài class ViewController, chúng ta viết extension

```
extension ViewController: UIScrollViewDelegate {
    func scrollViewDidScroll(_ scrollView: UIScrollView) {
        let pageIndex = round(scrollView.contentOffset.x/screenWidth)
        pageControl.currentPage = Int(pageIndex)
    }
}
```

```
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var scrollView: UIScrollView!
14     @IBOutlet weak var pageControl: UIPageControl!
15
16     var views: [CustomView] = []
17     let screenWidth = UIScreen.main.bounds.size.width
18     let screenHeight = UIScreen.main.bounds.size.height
19
20     override func viewDidLoad() {
21         super.viewDidLoad()
22
23         // gọi hàm tạo dữ liệu
24         createData()
25
26         // gọi hàm thêm customView
27         setupSlideScrollView(customView: views)
28
29         // cấu hình pageControl
30         pageControl.currentPage = 0
31         pageControl.numberOfPages = views.count
32         pageControl.pageIndicatorTintColor = UIColor.gray
33         pageControl.currentPageIndicatorTintColor = UIColor.purple
34
35         scrollView.delegate = self
36     }
37
38     func createData(){...}
39
40     | func setupSlideScrollView(customView: [CustomView]){...}
41
42 }
43
44
45 extension ViewController: UIScrollViewDelegate {
46     func scrollViewDidScroll(_ scrollView: UIScrollView) {
47         let pageIndex = round(scrollView.contentOffset.x/screenWidth)
48         pageControl.currentPage = Int(pageIndex)
49     }
50
51 }
```



Did you know that Winnie the chubby little cubby was based on a real, young bear in London