



# Lập trình iOS

## Bài 6. *Window – View* *và các đối tượng view cơ bản*

Ngành Mạng & Thiết bị di động





# Nội dung

---

## 1. Window

- Giới thiệu window
- Vai trò của window

## 2. View

## 3. View Controller

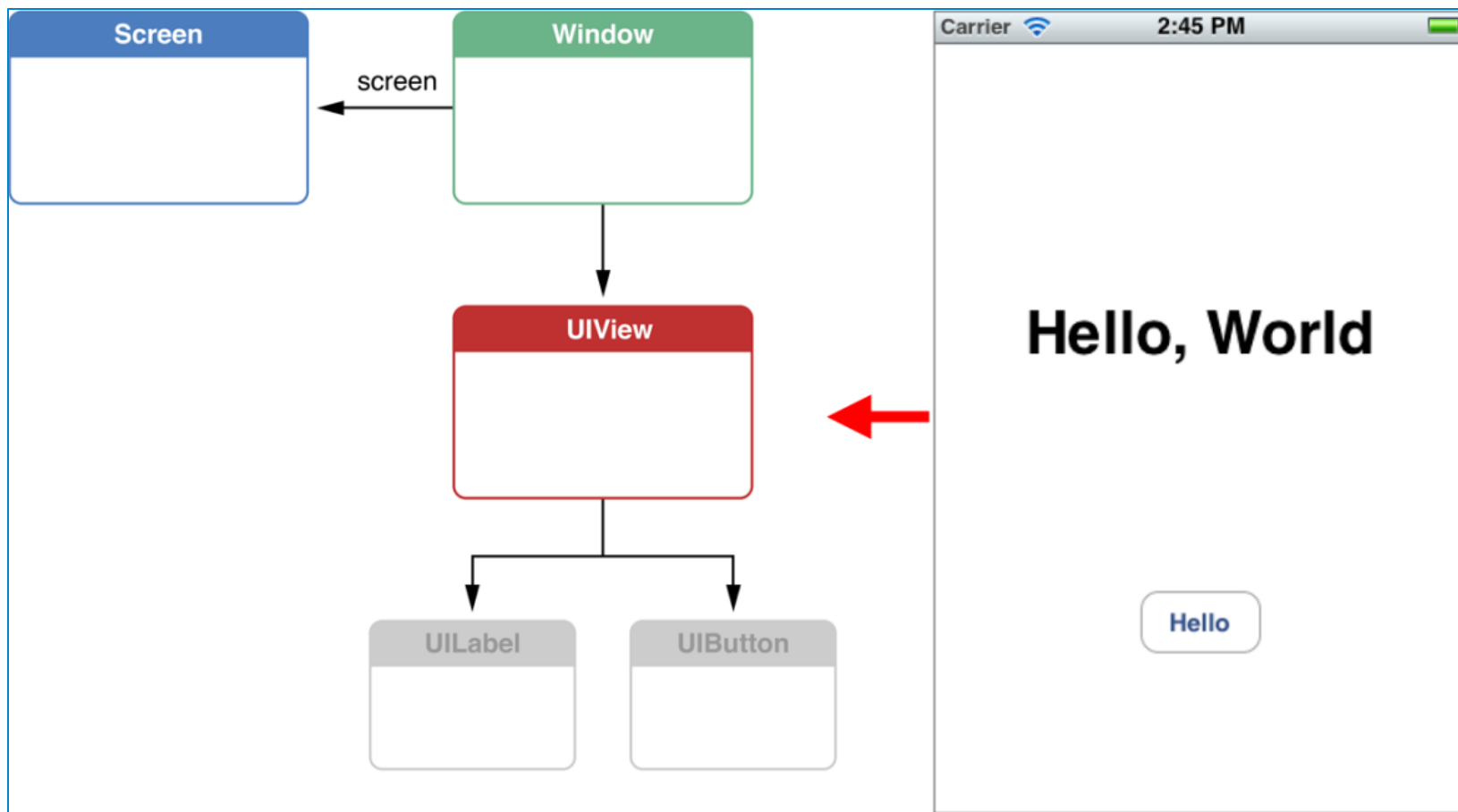
## 4. Điều khiển hiển thị nội dung văn bản

## 5. Điều khiển tương tác



# 1.1 Giới thiệu window

- ❑ Mỗi ứng dụng iOS cần ít nhất một đối tượng window thuộc lớp UIWindow.





## 1.2 Vai trò của window

---

### ❑ Một đối tượng window có các vai trò sau:

- Chứa nội dung hiển thị cho ứng dụng.
- Đóng một vai trò quan trọng trong việc cung cấp các sự kiện chạm cho view và các đối tượng khác trong ứng dụng.
- Làm việc với ViewController trong ứng dụng để tạo điều kiện thay đổi định hướng.



# Nội dung

---

## 1. Window

## 2. View

- Giới thiệu View
- Khảo sát lớp UIView

## 3. View Controller

## 4. Điều khiển hiển thị nội dung văn bản

## 5. Điều khiển tương tác



## 2.1 Giới thiệu View

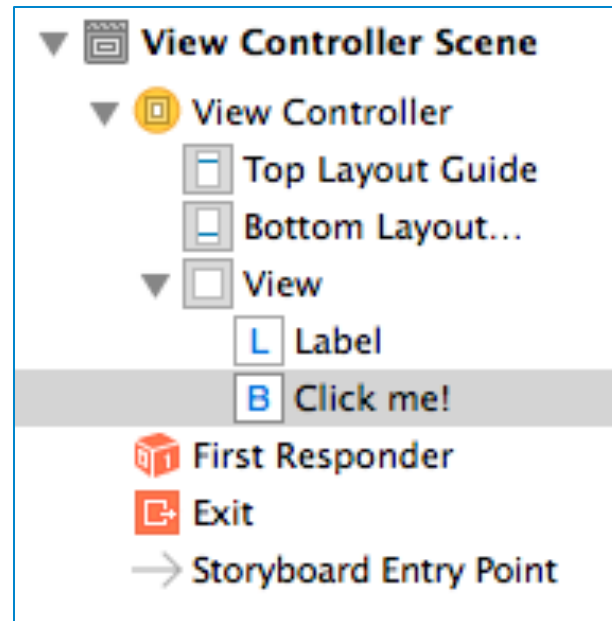
---

- ❑ Một view tượng trưng cho một giao diện người dùng. Mỗi view bao gồm một khu vực rõ ràng. Trong khu vực này nó hiển thị nội dung của nó hoặc trả lời sự kiện của người dùng.
- ❑ Các view có thể được đặt vào một hệ thống view. Khi đó các view này được gọi là subview và view bọc lấy chúng được gọi là superview. Các subview có vị trí hiển thị phụ thuộc vào superview. Vì vậy, khi superview di chuyển thì các subview của nó cũng được di chuyển theo.
- ❑ Khi thay đổi các giá trị thuộc tính của View có thể sử dụng chế độ animate.



## 2.1 Giới thiệu View

- ❑ Khi các view bị nằm chồng lên nhau thì việc ưu tiên hiển thị sẽ dựa vào thứ tự sắp xếp của nó, càng nằm ở vị trí ngoài thì càng được ưu tiên





## 2.2 Khảo sát lớp UIView

### ❑ Thuộc tính thường sử dụng

- **backgroundColor**: thể hiện màu nền của view.
- **hidden**: xét ẩn/hiện cho view. Giá trị bằng True là ẩn, bằng False là hiện.
- **alpha**: độ mờ của view, giá trị của alpha là số thực, từ 0 cho đến 1. Độ alpha càng thấp thì view càng mờ.
- **frame**: là khung hình chữ nhật bao gồm thông tin về kích thước và vị trí của view. Giá trị của frame là kiểu cấu trúc CGRect
- **center**: là vị trí trung tâm của view. Giá trị của center là kiểu cấu trúc CGPoint
- **subviews**: lưu danh sách các đối tượng view con bên trong view
- **superview**: lưu view chứa nó
- **tag**: là một số nguyên dùng để nhận dạng các đối tượng view trong ứng dụng





## 2.2 Khảo sát lớp UIView

---

### ❑ Phương thức thường sử dụng:

- - **initWithFrame**: khởi tạo với tham số truyền vào là một giá trị kiểu CGRect.
- - **addSubview**: thêm một view con vào trong view.
- - **bringSubviewToFront**: đưa subview ra ngoài cùng của chuỗi sắp xếp, có nghĩa là được ưu tiên hiển thị đầu tiên.
- - **sendSubviewToBack**: đưa subview vào trong cùng của chuỗi sắp xếp, có nghĩa là được ưu tiên hiển thị cuối cùng.
- - **addGestureRecognizer**: thêm xử lý các cử chỉ của người dùng tác động lên view.



## 2.2 Khảo sát lớp UIView

---

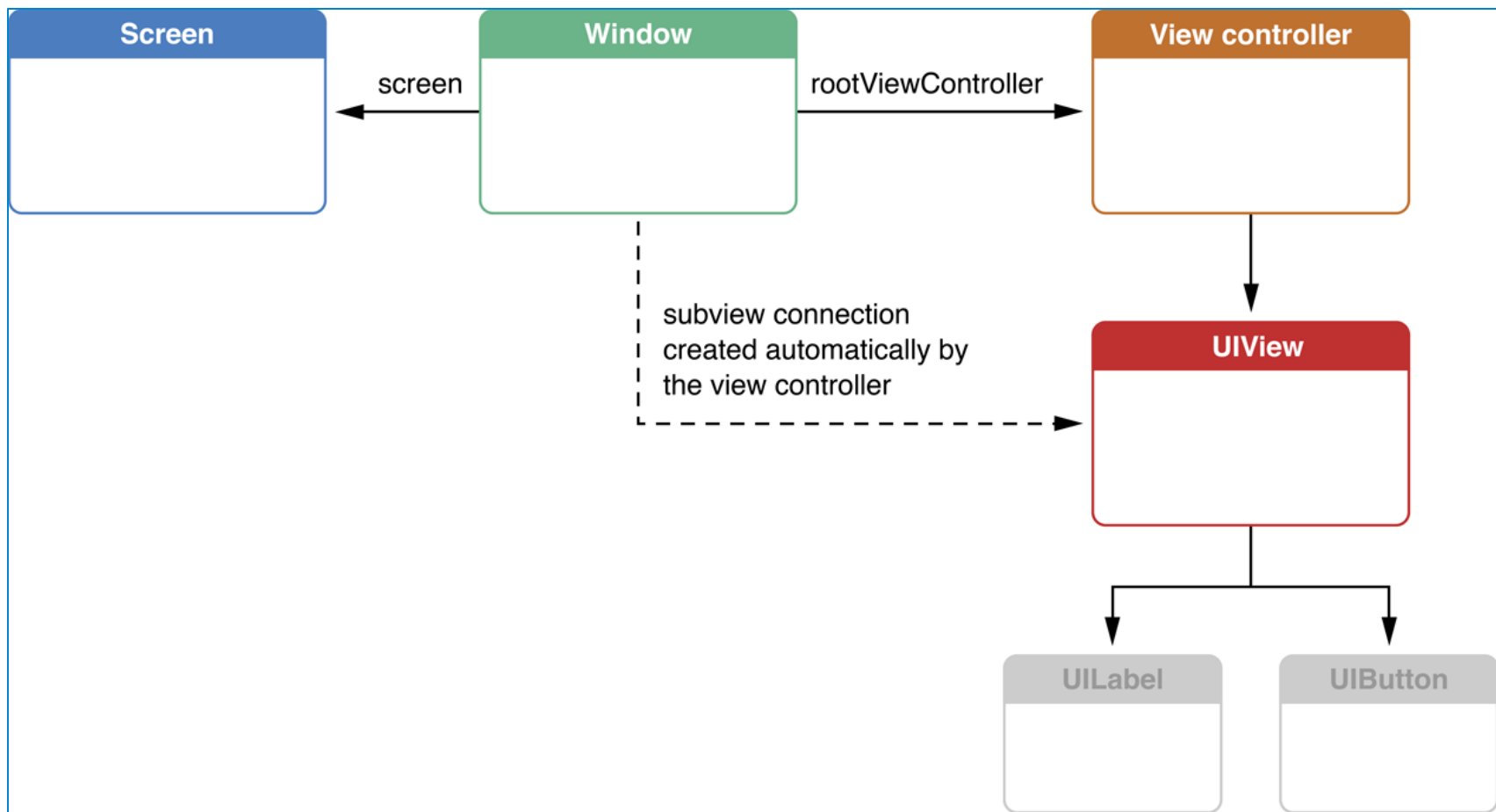
### ❑ Phương thức thường sử dụng:

- - **removeGestureRecognizer**: bỏ tính năng xử lý khi người dùng tác động một cử chỉ xác định lên view.
- + **animateWithDuration:delay:options:animations:completion**: xây dựng hiệu ứng cho các thay đổi của các đối tượng view
- - **viewWithTag**: là phương thức lấy ra đối tượng subview với tham số truyền vào là số tag mà ta đã quy định cho đối tượng đó



### 3. View Controller

- ❑ Mỗi view controller quản lý và điều khiển 1 view. View này thường là view gốc của một hệ thống view





## 3. View Controller

---

- ❑ Ngoài nhiệm vụ mặc định là quản lý view và tài nguyên thì tùy thuộc vào nhu cầu sử dụng, view controller còn có thể thực hiện thêm nhiều nhiệm vụ khác như xử lý các sự kiện người dùng thao tác lên control, gọi lệnh chuyển hướng, lấy dữ liệu và gọi lệnh hiển thị dữ liệu,...
- ❑ View controller chỉ tải đối tượng view của nó khi đối tượng view được cần để hiển thị. Nó cũng có thể giải phóng đối tượng view khi không cần thiết nữa. Vì thế nên view controller được xem là một phần quan trọng trong quản lý tài nguyên trong ứng dụng
- ❑ Mỗi view controller sẽ được quản lý bởi một lớp kế thừa từ UIViewController hoặc bởi chính lớp UIViewController (đây là lớp quản lý mặc định)
- ❑ Lớp quản lý view controller có thể kết nối đến các subview bên trong view bằng các kết nối IBOutlet, IBOutletCollection và IBAction



# Nội dung

---

## 1. Window

## 2. View

## 3. View Controller

## 4. Điều khiển hiển thị nội dung văn bản

- Label
- TextField
- TextView

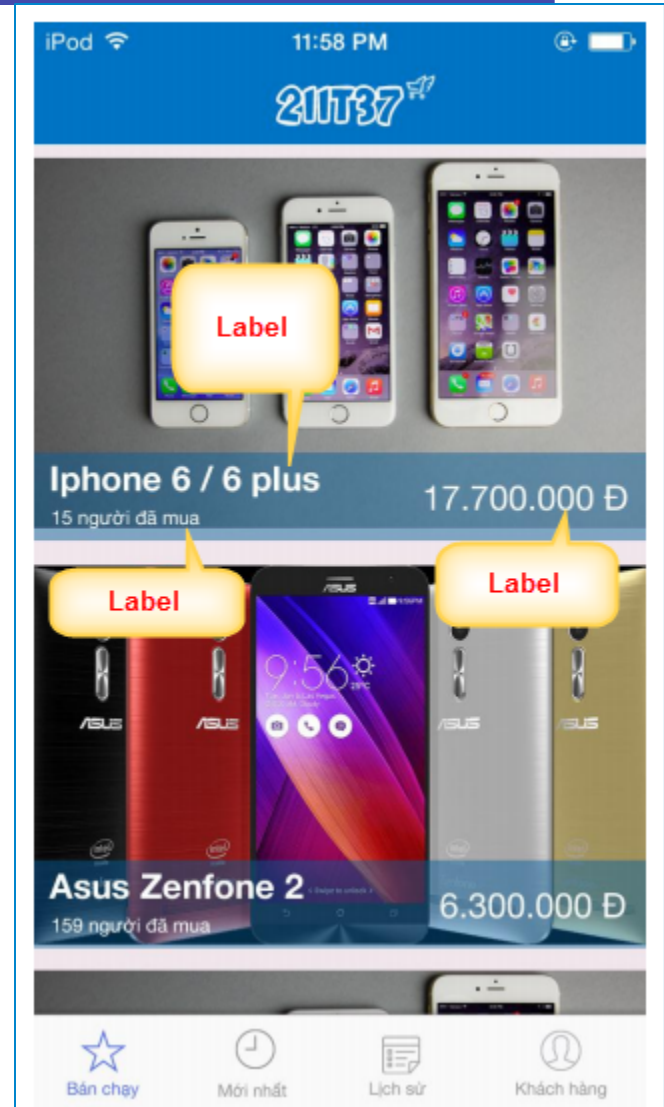
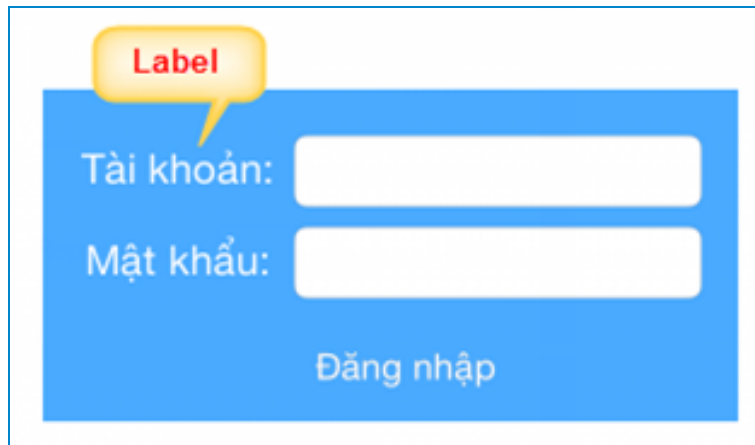
## 5. Điều khiển tương tác

## 4.1 Label

❑ Label là một đối tượng thuộc lớp UILabel. Dùng để hiển thị văn bản tĩnh có thuộc tính chỉ đọc.

- Mục đích sử dụng chính
  - Diễn giải chức năng của các điều khiển khác trong ứng dụng
  - Diễn giải ngữ cảnh hoặc chỉ dẫn trong ứng dụng

- Ví dụ





## 4.1 Label

- ❑ Tùy chỉnh hiển thị đối tượng UILabel trong Interface Builder

The screenshot shows the 'Label' settings panel in Interface Builder. It contains the following controls:

- Text:** A dropdown menu set to 'Plain' and a text field containing 'Label'.
- Color:** A color picker showing black and a dropdown menu set to 'Default'.
- Font:** A dropdown menu set to 'System 17.0' with a font icon and a size spinner.
- Alignment:** Three buttons for left, center, and right alignment, with the left button selected.
- Lines:** A text field set to '1' with a spinner.
- Behavior:** Two checkboxes: 'Enabled' (checked) and 'Highlighted' (unchecked).
- Baseline:** A dropdown menu set to 'Align Baselines'.
- Line Breaks:** A dropdown menu set to 'Truncate Tail'.
- Autoshrink:** A dropdown menu set to 'Fixed Font Size' and an unchecked checkbox for 'Tighten Letter Spacing'.
- Highlighted:** A color picker showing black and a dropdown menu set to 'Default'.
- Shadow:** A color picker showing red and a dropdown menu set to 'Default'.
- Shadow Offset:** Two spinners: 'Horizontal' set to '0' and 'Vertical' set to '-1'.



## 4.1 Label

### ❑ Các thuộc tính và phương thức thường dùng :

UILabel là một lớp kế thừa từ lớp UIView, do đó nó có thể sử dụng các thuộc tính và phương thức UIView. Ngoài ra nó còn có thêm một số thuộc tính riêng như:

- **text**: Lưu thông tin sẽ hiển thị trên label. Giá trị là một chuỗi thuộc kiểu NSString.
- **font**: là font chữ sẽ hiển thị trên label. Giá trị thuộc kiểu UIFont
- **textColor**: là màu chữ sẽ hiển thị trên label. Giá trị thuộc kiểu UIColor
- **textAlignment**: Canh chuỗi, có giá trị là một kiểu enum NSTextAlignment bao gồm canh trái, canh phải, canh giữa và canh đều
- **numberOfLines**: Số dòng có trong Label, mặc định là 1 dòng. Khi muốn Label hiển thị text ở nhiều dòng thì ta có thể tăng số dòng lên





## 4.2 TextField

❑ TextField là một đối tượng thuộc lớp UITextField. Dùng để nhập dữ liệu văn bản trên một dòng.

- Mục đích sử dụng chính
  - Hiển thị những thông tin dạng văn bản ngắn cho phép chỉnh sửa.
  - Lấy thông tin từ người dùng nhập vào

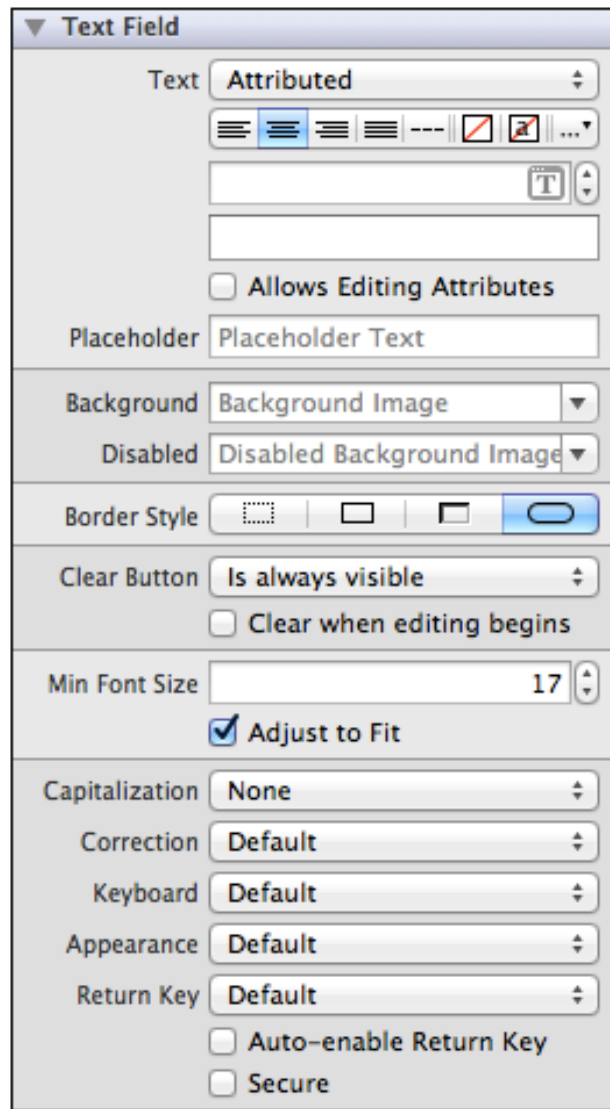
- Ví dụ





## 4.2 TextField

- ❑ Tùy chỉnh hiển thị đối tượng UITextField trong Interface Builder.





## 4.2 TextField

### ❑ Các thuộc tính và phương thức thường dùng :

Cũng là một đối tượng hiển thị văn bản như Label nên TextField cũng có các thuộc tính giống như Label. Ví dụ như các thuộc tính: text, font, textColor, textAlignment,... Ngoài ra TextField còn có thêm nhiều thuộc tính như:

- **placeholder**: là thuộc tính cho phép khi TextField chưa có giá trị thì nó sẽ hiển thị nội dung có trong placeholder.
- **delegate**: thuộc tính cho phép View controller nhận ủy thác từ TextField
- **keyboardType**: Loại bàn phím sẽ hỗ trợ nhập liệu cho TextField.
- **returnKeyType**: tùy chỉnh tiêu đề của phím return sao cho phù hợp với ngữ cảnh nhập liệu.
- **inputView**: Lưu trữ đối tượng bàn phím hỗ trợ nhập liệu cho TextField.
- **secureTextEntry**: là khai báo để che các ký tự nhập vào thành cái dấu “\*”. Thuộc tính này thường sử dụng cho nhập mật khẩu.



## 4.2 TextField

- ❑ Để có thể nhận ủy thác từ TextField, đối tượng delegate cần phải áp dụng protocol UITextFieldDelegate.
- ❑ Protocol UITextFieldDelegate cung cấp phương thức sau:
  - - **textFieldShouldBeginEditing**: Phương thức này sẽ hỏi delegate khi textField được bắt đầu chỉnh sửa. Phương thức không bắt buộc.
  - - **textFieldDidBeginEditing**: Phương thức này thông báo cho delegate rằng textField đã bắt đầu được chỉnh sửa. Phương thức không bắt buộc.
  - - **textFieldShouldEndEditing**: Phương thức này sẽ hỏi delegate khi textField kết thúc quá trình chỉnh sửa. Phương thức không bắt buộc.
  - - **textFieldDidEndEditing**: Phương thức này sẽ được gọi khi textField kết thúc chỉnh sửa. Phương thức không bắt buộc.



## 4.2 TextField

- - **textField:shouldChangeCharactersInRange:replacementString:**  
textField sẽ gọi phương thức này khi nào người dùng gõ một ký tự mới hoặc xóa một ký tự trong textField. Phương thức không bắt buộc
- - **textFieldShouldClear:** textField sẽ gọi phương thức này khi người dùng bấm vào phím “Clear”. Phím này mặc định không hiển thị, nếu muốn hiển thị ta có thể sử dụng thuộc tính **clearButtonMode** để ẩn hoặc hiện tùy ý. Phương thức không bắt buộc
- - **textFieldShouldReturn:** đây là phương thức sẽ được gọi khi người dùng bấm vào phím “Return”. Thường thì người ta hay sử dụng phương thức này để giúp người dùng thông báo rằng họ đã nhập liệu xong và giúp họ ẩn bàn phím.



## 4.3 TextView

❑ UITextView: đối tượng cho phép nhập và hiển thị dữ liệu văn bản nhiều dòng, hỗ trợ thao tác cuộn.

- Mục đích sử dụng chính
  - Hiển thị và cho phép tùy chỉnh dữ liệu nhập và xử lý khi có yêu cầu.
- Ví dụ

The screenshot shows a mobile application interface for adding a movie. At the top is a green header bar with three buttons: 'Hủy' (Cancel), 'Thêm Phim' (Add Movie), and 'Xong' (Done). Below the header, there are several input fields for movie details: 'Tên Phim' (Movie Name) with the value 'Hobbit 2', 'Thời lượng' (Duration) with '161' and 'phút' (minutes), 'Đạo diễn' (Director) with 'Peter Jackson', 'Ngày chiếu' (Release Date) with '01.01.2014', 'Đánh giá' (Rating) with '8.1', and 'Giá vé' (Ticket Price) with '135000'. The 'Nội dung' (Content) field is a multi-line text area containing a paragraph about the second part of a movie. An arrow labeled 'TextView' points to this text area.

Hủy Thêm Phim Xong		
Tên Phim	Hobbit 2	
Thời lượng	161	phút
Đạo diễn	Peter Jackson	
Ngày chiếu	01.01.2014	
Đánh giá	8.1	
Giá vé	135000	
Nội dung	<p>Phần 2 tiếp tục kể về cuộc phiêu lưu của nhân vật chính Bilbo Baggins và cuộc hành trình với phù thủy Gandalf cùng mười ba người lùn dẫn đầu bởi Thorin Oakenshield, họ mang trên mình một nhiệm vụ sử thi để đòi lại Lonely Mountain và Vương quốc người lùn đã mất của Erebor.</p>	



## 4.3 TextView

- ❑ Tùy chỉnh hiển thị đối tượng UITextView trong Interface Builder.

The screenshot shows the 'Text View' settings panel in Xcode's Interface Builder. The panel is organized into several sections:

- Text:** A dropdown menu set to 'Plain'. Below it is a text area containing Vietnamese text: 'Phần 2 tiếp tục kể về cuộc phiêu lưu của nhân vật chính Bilbo Baggins và cuộc hành trình với phù thủy Gandalf cùng mười ba người lùn dẫn đầu bởi Thorin Oakenshield, họ mang trên mình một nhiệm vụ sử thi để đòi lại Lonely Mountain và Vương quốc người lùn đã mất của Erebor.'
- Color:** A color picker set to 'Default'.
- Font:** A dropdown menu set to 'System 14.0'.
- Alignment:** Three buttons for text alignment: Left (selected), Center, and Right.
- Behavior:** Two checked checkboxes: 'Editable' and 'Selectable'.
- Detection:** Four unchecked checkboxes: 'Links', 'Addresses', 'Phone Numbers', and 'Events'.
- Capitalization:** A dropdown menu set to 'Sentences'.
- Correction:** A dropdown menu set to 'Default'.
- Keyboard:** A dropdown menu set to 'Default'.
- Appearance:** A dropdown menu set to 'Default'.
- Return Key:** A dropdown menu set to 'Default'.
- Auto-enable Return Key:** An unchecked checkbox.



## 4.3 TextView

### ❑ Các thuộc tính và phương thức thường dùng:

TextView cũng có thể chỉnh sửa văn bản như TextField nên nó cũng có các thuộc tính hỗ trợ nhập liệu như: **keyboardType**, **returnKeyType**, **inputView**,... Và một số thuộc tính hỗ trợ hiển thị văn bản như **text**, **font**, **textColor**, **textAlignment**,... Ngoài ra TextView còn có thêm nhiều thuộc tính như:

- **editable**: là thuộc tính cho biết TextView có thể chỉnh sửa nội dung hay không.
- **delegate**: đối tượng nhận ủy thác của TextView.
- **scrollEnabled**: bật/tắt chế độ cuộn nội dung bên trong TextView.
- **selectable**: bật/tắt cho phép chọn nội dung trong TextView
- **selectedRange**: lưu trữ vị trí đoạn nội dung mà người dùng đã chọn. Giá trị là kiểu NSRange.





## 4.3 TextView

---

- ❑ Để có thể nhận ủy thác từ TextView, đối tượng delegate cần phải áp dụng protocol UITextViewDelegate.
- ❑ Protocol UITextViewDelegate cung cấp phương thức không bắt buộc sau :
  - - **textViewShouldBeginEditing**: Phương thức này sẽ hỏi delegate khi textView được bắt đầu chỉnh sửa.
  - - **textViewDidBeginEditing**: Phương thức này thông báo cho delegate biết rằng textView đã bắt đầu được chỉnh sửa.
  - - **textViewShouldEndEditing**: Phương thức này sẽ hỏi delegate khi textView kết thúc quá trình chỉnh sửa.
  - - **textViewDidEndEditing**: Phương thức này sẽ được gọi khi textView kết thúc chỉnh sửa.



## 4.3 TextView

---

- - **textView:shouldChangeTextInRange:replacementText:** textView sẽ gọi phương thức này khi nào người dùng gõ một ký tự mới hoặc xóa một ký tự trong textView.
- - **textViewDidChange:** textView sẽ gọi phương thức này khi người dùng đã chỉnh sửa nội dung trong textView.
- - **textViewDidChangeSelection:** Phương thức này sẽ báo với delegate rằng đoạn nội dung được chọn đã được thay đổi.
- - **textView:shouldInteractWithTextAttachment:inRange:** Phương thức này sẽ hỏi delegate có nên cho phép người dùng tương tác với đoạn văn bản đính kèm hay không.
- - **textView:shouldInteractWithURL:inRange:** Phương thức này sẽ hỏi delegate có nên cho phép người dùng tương tác với đoạn URL hay không.



# Nội dung

---

1. Window
2. View
3. View Controller
4. Điều khiển hiển thị nội dung văn bản
5. Điều khiển tương tác
  - Button
  - Switch
  - Segment
  - Slider
  - Stepper

## 5.1 Button

❑ Button là một đối tượng thuộc lớp UIButton, thường được sử dụng để chuyển hướng người dùng. Button nhận các tương tác chạm, nhấn trên màn hình thiết bị thuộc khu vực hiển thị và thực thi hành động đã thiết lập sẵn.

- Mục đích sử dụng chính
  - Thực thi các xử lý được thiết lập dựa trên tương tác người dùng.
- Ví dụ





## 5.1 Button

- ❑ Tùy chỉnh hiển thị đối tượng UIButton trong Interface Builder.

The image shows the 'Button' settings panel in Interface Builder. It contains various configuration options for a UIButton, including Type, State Config, Title, Font, Text Color, Shadow Color, Image, Background, Shadow Offset, Drawing options, Line Break, Edge, and Inset values.

**Button**

Type: System

State Config: Default

Title: Plain

Đặt Vé

Font: System 15.0

Text Color: Default

Shadow Color: Default

Image: Default Image

Background: Default Background Image

Shadow Offset: 0.0 (Width) 0.0 (Height)

☐ Reverses On Highlight

Drawing ☐ Shows Touch On Highlight

☒ Highlighted Adjusts Image

☒ Disabled Adjusts Image

Line Break: Truncate Middle

Edge: Content

Inset: 0 (Top) 0 (Bottom) 0 (Left) 0 (Right)



## 5.1 Button

---

### ❑ Các thuộc tính và phương thức thường dùng :

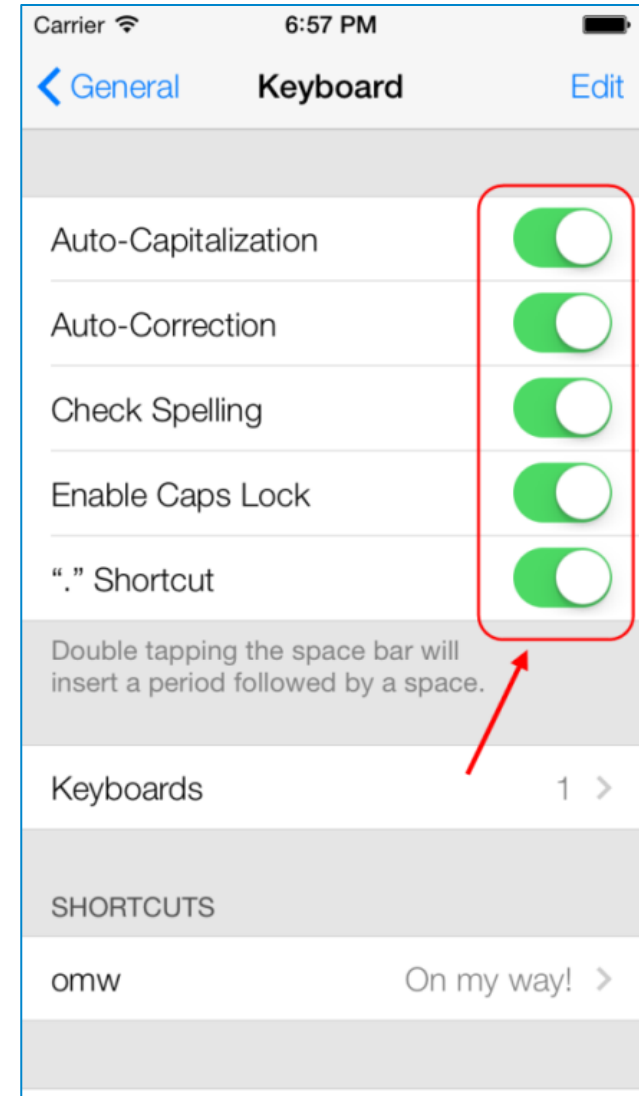
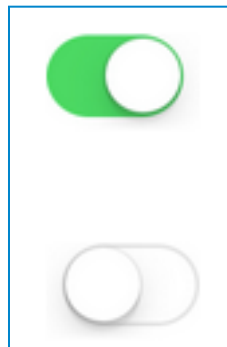
- **titleLabel**: là tiêu đề của button, đây là một đối tượng thuộc lớp UILabel do đó ta có thể thao tác với nó như thao tác với một Label.
- **tintColor**: là màu để áp dụng cho hình tiêu đề của button.
- **buttonType**: cho biết loại của button. Thuộc tính này chỉ cho phép đọc.
- **currentTitle**: cho biết tiêu đề hiện đang hiển thị trên button. Thuộc tính này chỉ cho phép đọc.
- - **setTitle:forState**: dùng để xét tiêu đề sẽ hiển thị cho trạng thái truyền vào.
- - **setTitleColor:forState**: xét màu tiêu đề cho trạng thái truyền vào.
- - **setBackgroundImage:forState**: hình nền tương ứng với trạng thái truyền vào.
- - **setImage:forState**: hình hiển thị tương ứng với trạng thái truyền vào.



## 5.2 Switch

❑ UISwitch: đối tượng cho phép người dùng thực hiện chuyển đổi giữa 2 trạng thái.

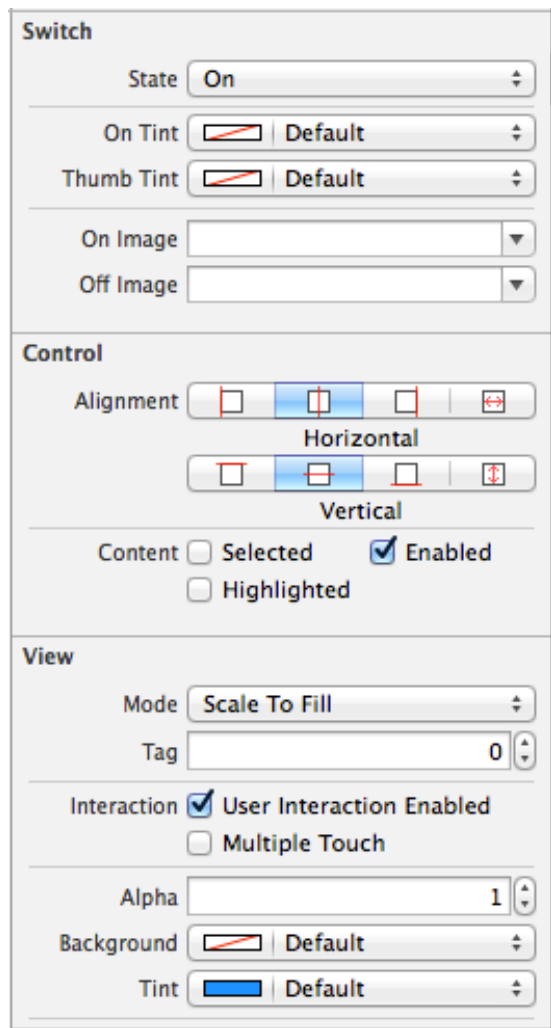
- Mục đích sử dụng chính
  - Thực thi chuyển đổi giữa hai trạng thái và xử lý theo từng trạng thái riêng biệt.
  - Bật hoặc tắt một tính năng nào đó.
- Ví dụ





## 5.2 Switch

- ❑ Tùy chỉnh hiển thị đối tượng UISwitch trong Interface Builder.







## 5.2 Switch

---

### ❑ Các thuộc tính và phương thức thường dùng :

- **on**: là thuộc tính thể hiện trạng thái đang bật của Switch. Nếu kết quả bằng NO nghĩa là đang trạng thái tắt.
- **onTintColor**: là màu của Switch khi đang ở trạng thái bật.
- **tintColor**: là màu viền của Switch khi đang ở trạng thái tắt.
- - **setOn:animated**: là phương thức hỗ trợ bật Switch với hiệu ứng đi kèm nếu muốn.

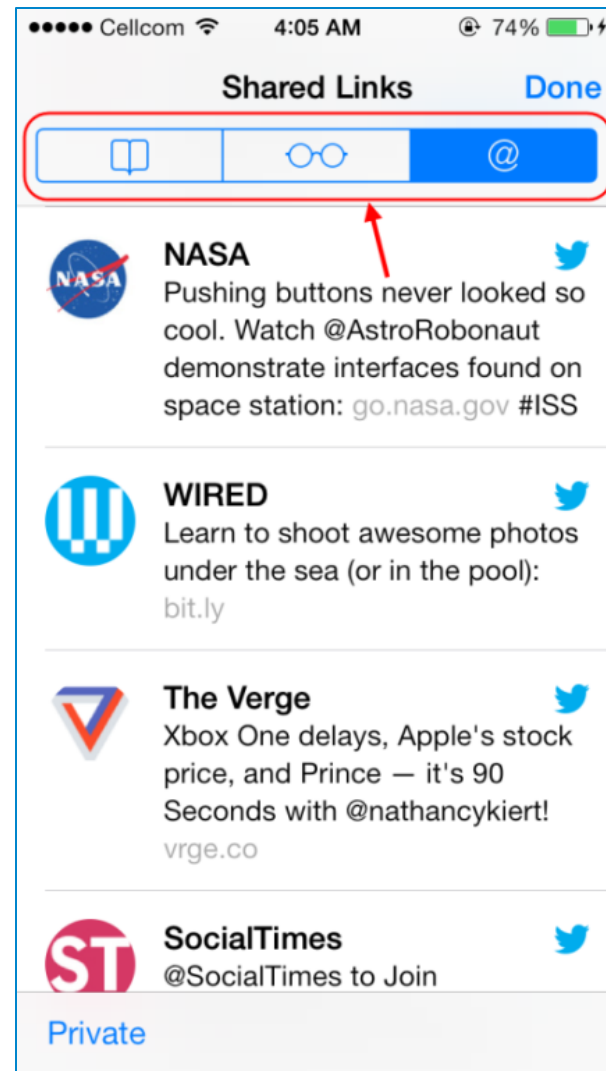
## 5.3 Segment

- ❑ **UISegmentedControl:** dạng điều khiển tổ hợp các nút theo chiều ngang, với mỗi nút được thực thi xử lý riêng biệt, tại một thời điểm chỉ một nút được thực thi.

- Mục đích sử dụng chính

- Tổ hợp thực thi các nút điều khiển và xử lý theo từng nút riêng biệt khi có tương tác.

- Ví dụ





## 5.3 Segment

- ❑ Tùy chỉnh hiển thị đối tượng **UISegmentedControl** trong Interface Builder.

The screenshot shows the properties panel for a Segmented Control in Interface Builder, organized into three sections: Segmented Control, Control, and View.

**Segmented Control**

- Style: Plain
- State: ☐ Momentary
- Segments: 3
- Segment: Segment 2 - Yêu Thích
- Title: Yêu Thích
- Image: Image
- Behavior: ☒ Enabled, ☐ Selected
- Content Offset: X: 0, Y: 0

**Control**

- Alignment: Horizontal (Left, Center, Right, Fill) and Vertical (Top, Center, Bottom, Fill)
- Content: ☐ Selected, ☒ Enabled, ☐ Highlighted

**View**

- Mode: Scale To Fill
- Tag: 0
- Interaction: ☒ User Interaction Enabled, ☐ Multiple Touch



## 5.3 Segment

### ❑ Các thuộc tính và phương thức thường dùng:

- **numberOfSegments**: là thuộc tính lưu số button có trong Segment.
- **selectedSegmentIndex**: lưu trữ vị trí button được người dùng bấm vào sau cùng.
- - **setImage:forSegmentAtIndex**: xét hình ảnh cho button ở vị trí truyền vào.
- - **imageForSegmentAtIndex**: lấy ra hình ảnh của button ở vị trí truyền vào.
- - **setTitle:forSegmentAtIndex**: xét tiêu đề cho button ở vị trí truyền vào.
- - **titleForSegmentAtIndex**: lấy ra tiêu đề của button tại vị trí truyền vào
- - **insertSegmentWithImage:atIndex:animated**: thêm một button với hình ảnh và vị trí của button được truyền vào.
- - **insertSegmentWithTitle:atIndex:animated**: thêm một button với tiêu đề và vị trí của button được truyền vào.

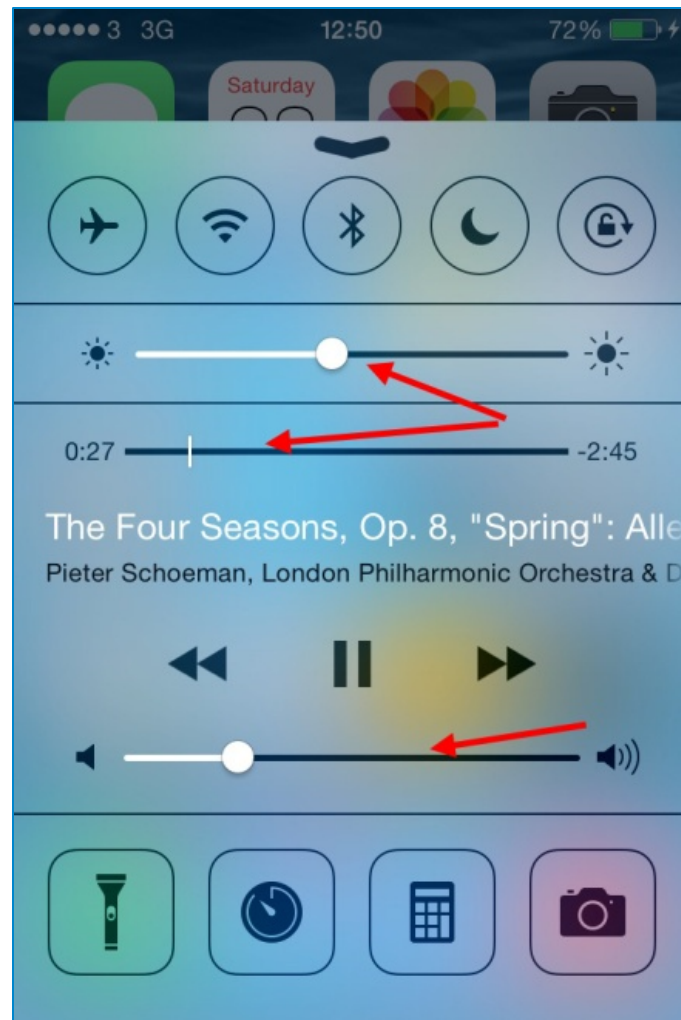
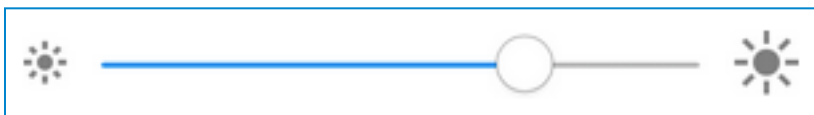
## 5.4 Slider



❑ UISlider: dạng điều khiển cho phép người dùng tùy chỉnh thông số dựa trên tương tác kéo trượt.

- Mục đích sử dụng chính
  - Thực hiện tăng giảm giá trị một cách tự động hoặc dựa trên tương tác.
  - Thiết lập giá trị cho điều khiển khác.
  - Thiết lập giá trị nhanh chóng bằng tương tác kéo trượt đơn giản.

● Ví dụ





## 5.4 Slider

- ❑ Tùy chỉnh hiển thị đối tượng UISlider trong Interface Builder.

The screenshot shows the properties panel for a UISlider in Interface Builder, organized into three sections: Slider, Control, and View.

**Slider**

- Value:** A numeric field showing 0, with a Minimum value of 0 and a Maximum value of 1. Below it, a **Current** value of 0.5 is displayed.
- Min Image:** A dropdown menu set to "Image".
- Max Image:** A dropdown menu set to "Image".
- Min Track Tint:** A color picker and a dropdown menu set to "Default".
- Max Track Tint:** A color picker and a dropdown menu set to "Default".
- Thumb Tint:** A color picker and a dropdown menu set to "Default".
- Update Events:** A checkbox for "Continuous" is checked.

**Control**

- Alignment:** Two rows of icons. The first row is labeled "Horizontal" and the second row is labeled "Vertical". The "Horizontal" row has four icons, with the second one (center) selected. The "Vertical" row has four icons, with the second one (center) selected.
- Content:** Three checkboxes: "Selected" (unchecked), "Enabled" (checked), and "Highlighted" (unchecked).

**View**

- Mode:** A dropdown menu set to "Scale To Fill".
- Tag:** A numeric field showing 0.
- Interaction:** A checkbox for "User Interaction Enabled" is checked.



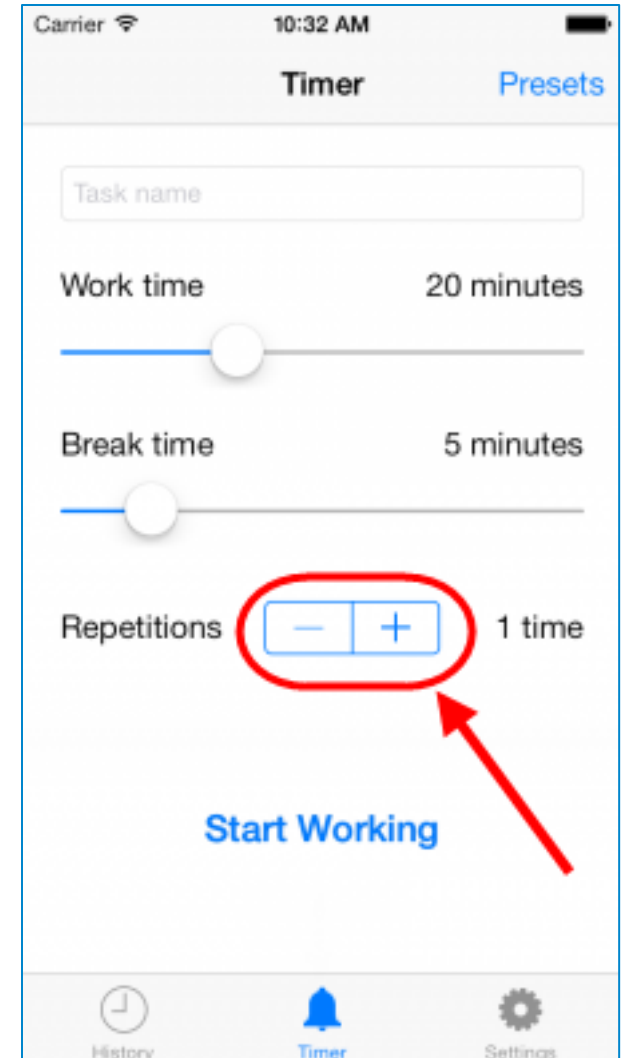
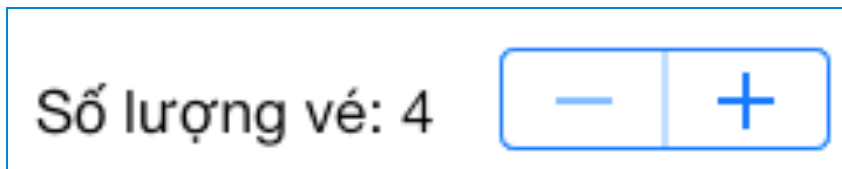
## 5.4 Slider

### ❑ Các thuộc tính và phương thức thường dùng:

- **value**: giá trị hiện tại của Slider.
- **minimumValue**: giá trị nhỏ nhất của Slider.
- **maximumValue**: giá trị lớn nhất của Slider.
- - **setValue:animated**: Phương thức cho phép xét giá trị hiện tại của Slider .
- - **thumbImageForState**: phương thức trả về hình đại diện cho giá trị hiện tại của Slider dựa vào trạng thái truyền vào.
- - **setThumbImage:forState**: phương thức cho phép xét hình đại diện tương ứng với trạng thái của control

## 5.5 Stepper

- ❑ **UIStepper:** dạng điều khiển cho phép tăng giảm với lượng giá trị nhất định.
  - Mục đích sử dụng chính
    - Tăng giảm giá trị có thông số rời rạc.
    - Thiết lập giá trị cho điều khiển khác.
  - Ví dụ







## 5.5 Stepper

- ❑ Tùy chỉnh hiển thị đối tượng UIStepper trong Interface Builder.

The image shows the configuration panel for a UIStepper in Interface Builder. It is divided into three main sections: Stepper, Control, and View.

**Stepper**

- Value: 0 (Minimum), 100 (Maximum)
- Current: 0, Step: 1
- Behavior: ☒ Autorepeat, ☒ Continuous, ☐ Wrap

**Control**

- Alignment: Horizontal (selected), Vertical
- Content: ☐ Selected, ☒ Enabled, ☐ Highlighted

**View**

- Mode: Scale To Fill
- Tag: 0
- Interaction: ☒ User Interaction Enabled, ☐ Multiple Touch
- Alpha: 1
- Background: Default
- Tint: Default



## 5.5 Stepper

---

### ❑ Các thuộc tính và phương thức thường dùng

- **minimumValue**: giá trị nhỏ nhất của Stepper.
- **maximumValue**: giá trị lớn nhất của Stepper.
- **stepValue**: đơn vị cho mỗi lần bấm tăng/giảm.
- **value**: giá trị hiện tại của Stepper.
- - **backgroundImageForState**: phương thức trả về hình background trả về tương ứng với trạng thái của control truyền vào.
- - **setBackgroundImage:forState**: phương thức cho phép xét hình ảnh background tương ứng với trạng thái của control truyền vào.

