








# Lập trình ứng dụng trên iOS

Label, Button, Text Field,  
Slider, Switch, Segment Control

# Nội dung

- Label → **Label** (12)
- Button → 
- Text Field → 
- Sliders → 
- Switches → 
- Segmented Controls → 

# Label

- text
- textColor
- textAlignment
- font
- lineBreakMode
- backgroundColor

**Label** (12)

```

49     UILabel *lbName = [[UILabel alloc] initWithFrame:CGRectMake(10, 10, 200
    , 21)];
50     lbName.text = @"Enter your name:";
51     lbName.textColor = [UIColor redColor];
52     lbName.textAlignment = NSTextAlignmentLeft;
53     lbName.font = [UIFont fontWithName:@"Times New Roman" size:20];
54     [self.view addSubview:lbName];
    
```



# Button



- + (id)buttonWithType:(UIButtonType)buttonType
- – (void)setTitle:(NSString \*)title forState:(UIControlState)state
- - (void)addTarget:(id)target action:(SEL)action forControlEvents:(UIControlEvents)controlEvents

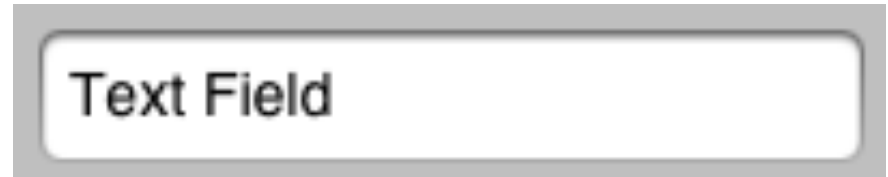
```

49 UIButton *btn = [UIButton buttonWithType:UIButtonTypeRoundedRect];
50 btn.frame = CGRectMake(10, 10, 70, 31);
51 [btn setTitle:@"BTN" forState:UIControlStateNormal];
52 [btn setTitleColor:[UIColor redColor] forState:UIControlStateNormal];
53 [btn addTarget:self action:@selector(clickMe:) forControlEvents:
    UIControlEventTouchUpInside];
54 [self.view addSubview:btn];
    
```



# Text Field

- text
- textAlignment
- textColor
- background
- clearButtonMode
- borderStyle
- delegate
- editing
- font
- placeholder



# Text Field

- text
  - @property(nonatomic, copy) NSString \*text

```
UITextField *tfName = [[UITextField alloc] initWithFrame:CGRectMake(
    10.0, 10.0, 250.0, 31.0)];
[tfName setText:@"Mr David"];
```



- textAlignment: NSTextAlignmentLeft (default), NSTextAlignmentCenter, NSTextAlignmentRight
  - @property(nonatomic) NSTextAlignment textAlignment

```
// Left
[tfName setTextAlignment:NSTextAlignmentLeft];
```



```
// Right
[tfName setTextAlignment:NSTextAlignmentRight];
```



```
// Center
[tfName setTextAlignment:NSTextAlignmentCenter];
```





# Text Field

- `textColor` (default -> nil)
  - `@property(n nonatomic, retain) UIColor *textColor`

```
[tfName setTextColor:(UIColor *)];
```

```
[tfName setTextColor:[UIColor redColor]];
```

Mr David

- `background` (default -> nil)
  - `@property(n nonatomic, retain) UIImage *background`

```
[tfName setBackground:(UIImage *)];
```

```
[tfName setBorderStyle:UITextBorderStyleNone];
```

```
[tfName setBackground:[UIImage imageNamed:@"Image_bg.png"]];
```

UITextField with custom background im...

# Text Field

- clearButtonMode
  - @property(nonatomic) UITextFieldViewMode clearButtonMode

```
[tfName setClearButtonMode:(UITextFieldViewMode)];
[tfName setClearButtonMode:UITextFieldViewModeAlways];
[tfName setClearButtonMode:UITextFieldViewModeNever];
[tfName setClearButtonMode:UITextFieldViewModeUnlessEditing];
[tfName setClearButtonMode:UITextFieldViewModeWhileEditing];
```





# Text Field

- `borderStyle`
  - `@property(nonatomic) UITextBorderStyle borderStyle`

```
[tfName setBorderStyle:(UITextBorderStyle)];
```

```
[tfName setBorderStyle:UITextBorderStyleNone]; // Default
```

Mr David

```
[tfName setBorderStyle:UITextBorderStyleLine];
```

Mr David

```
[tfName setBorderStyle:UITextBorderStyleBezel];
```

Mr David

```
[tfName setBorderStyle:UITextBorderStyleRoundedRect];
```

Mr David



# Text Field

- delegate
  - @property(n nonatomic, assign) id<UITextFieldDelegate> delegate

```
// return YES to allow editing to stop and to resign first responder status
. NO to disallow the editing session to end
- (BOOL)textFieldShouldEndEditing:(UITextField *)textField;
// return NO to disallow editing.
- (BOOL)textFieldShouldBeginEditing:(UITextField *)textField;
// became first responder
- (void)textFieldDidBeginEditing:(UITextField *)textField;
// may be called if forced even if shouldEndEditing returns NO
// (e.g. view removed from window) or endEditing:YES called
- (void)textFieldDidEndEditing:(UITextField *)textField;
// return NO to not change text
- (BOOL)textField:(UITextField *)textField shouldChangeCharactersInRange:
    (NSRange)range replacementString:(NSString *)string;
// called when clear button pressed. return NO to ignore (no notifications)
- (BOOL)textFieldShouldClear:(UITextField *)textField;
// called when 'return' key pressed. return NO to ignore.
- (BOOL)textFieldShouldReturn:(UITextField *)textField;
```

# Text Field

- editing
  - @property(nonatomic, readonly, getter=isEditing) BOOL editing

```
[tfName setEnabled:(BOOL)];
```

```
[tfName setEnabled:YES]; [tfName setEnabled:NO];
```

- font
  - @property(nonatomic, retain) UIFont \*font

```
[tfName setFont:[UIFont fontWithName:(NSString *) size:(CGFloat)]];
[tfName setFont:[UIFont fontWithName:@"Times New Roman" size:20]];
[tfName setAdjustsFontSizeToFitWidth:(BOOL)];
[tfName setAdjustsFontSizeToFitWidth:YES];
[tfName setAdjustsFontSizeToFitWidth:NO];
```

```
[tfName setAdjustsFontSizeToFitWidth:(BOOL)];
```

```
[tfName setAdjustsFontSizeToFitWidth:YES];
```

```
[tfName setAdjustsFontSizeToFitWidth:NO];
```

```
[tfName setAdjustsFontSizeToFitWidth:NO];
```

This is some large te

This is some large text that will s

is some large text that will shrink automatical



# Text Field

- placeholder
  - @property(nonatomic, copy) NSString \*placeholder

```
[tfName setPlaceholder:(NSString *)];
```

```
[tfName setPlaceholder:@"Type text here"];
```



# Sliders



- **value**
  - @property(nontatomic) float value

```
slider.value = 5.5;
```

- **minimumValue**
  - @property(nontatomic) float minimumValue

```
slider.minimumValue = 0.0;
```

- **maximumValue**
  - @property(nontatomic) float maximumValue

```
slider.maximumValue = 100.0;
```

- **continuous**
  - @property(nonatomic, getter=isContinuous) BOOL continuous

```
slider.continuous = YES;
```

# Sliders



- Ví dụ:

```
49  CGRect rect = CGRectMake(10,60, 200, 30);
50  UISlider *mySlider = [[UISlider alloc] initWithFrame:rect];
51  [mySlider addTarget:self action:@selector(sliderValueChanged:)
    forControlEvents:UIControlEventValueChanged];
52  mySlider.backgroundColor = [UIColor clearColor];
53  mySlider.minimumValue = 0.0;
54  mySlider.maximumValue = 10.0;
55  mySlider.continuous = YES;
56  mySlider.value = 5.0;
57  [self.view addSubview: mySlider];
```





# Switches



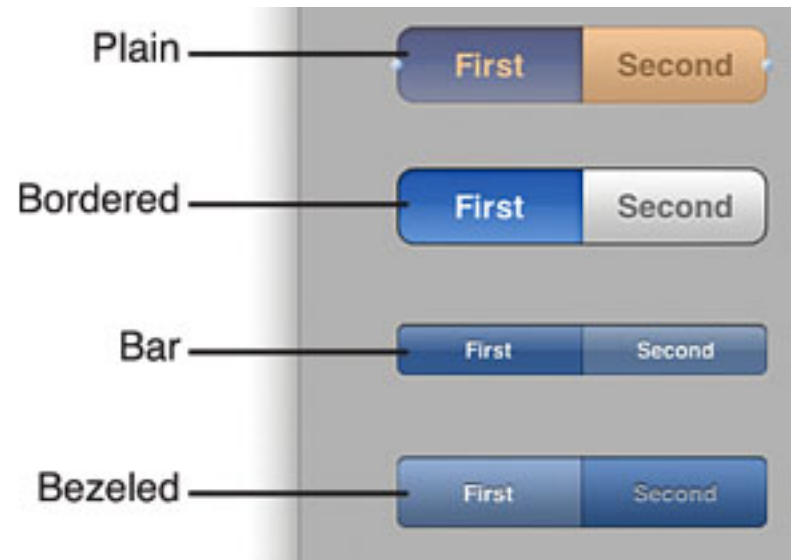
- setOn
  - @property(nonatomic, getter=isOn) BOOL on
- - (void)setOn:(BOOL)on animated:(BOOL)animated

```
CGRect rect = CGRectMake(10, 90, 100, 30);
UISwitch *mySwitch = [[UISwitch alloc] initWithFrame:rect];
[mySwitch addTarget:self action:@selector(switchValueChanged:)
  forControlEvents:UIControlEventValueChanged];
mySwitch.backgroundColor = [UIColor clearColor];
[self.view addSubview: mySwitch];
```

# Segmented Controls



- `selectedSegmentIndex`
  - @property(nonatomic) NSInteger `selectedSegmentIndex`
- `segmentedControlStyle`
  - @property(nonatomic) `UISegmentedControlStyle` `segmentedControlStyle`





# Segmented Controls

First

Second

- Ví dụ:

```

59  NSArray *itemArray = [NSArray arrayWithObjects: @"One", @"Two",
60      @"Three", nil];
61  UISegmentedControl *segmentedControl = [[UISegmentedControl
62      alloc] initWithItems:itemArray];
63  segmentedControl.frame = CGRectMake(35, 200, 250, 50);
64  segmentedControl.segmentedControlStyle =
65      UISegmentedControlStylePlain;
66  segmentedControl.selectedSegmentIndex = 1;
    [segmentedControl addTarget:self action:@selector(pickOne:)
        forControlEvents:UIControlEventValueChanged];
    [self.view addSubview: segmentedControl];
    [segmentedControl release];
    
```

One

Two

Three



# Segmented Controls

First

Second

- - (void)setTitle:(NSString \*)title forSegmentAtIndex:(NSUInteger)segment
- - (void)setImage:(UIImage \*)image forSegmentAtIndex:(NSUInteger)segment
- - (void)insertSegmentWithTitle:(NSString \*)title atIndex:(NSUInteger)segment animated:(BOOL)animated
- - (void)removeSegmentAtIndex:(NSUInteger)segment animated:(BOOL)animated
- removeAllSegments