Level 2

Subviews

CODE SCHOOL

Try iOS

• PRESENTS •

In this level we’ll cover

01 Creating a Button

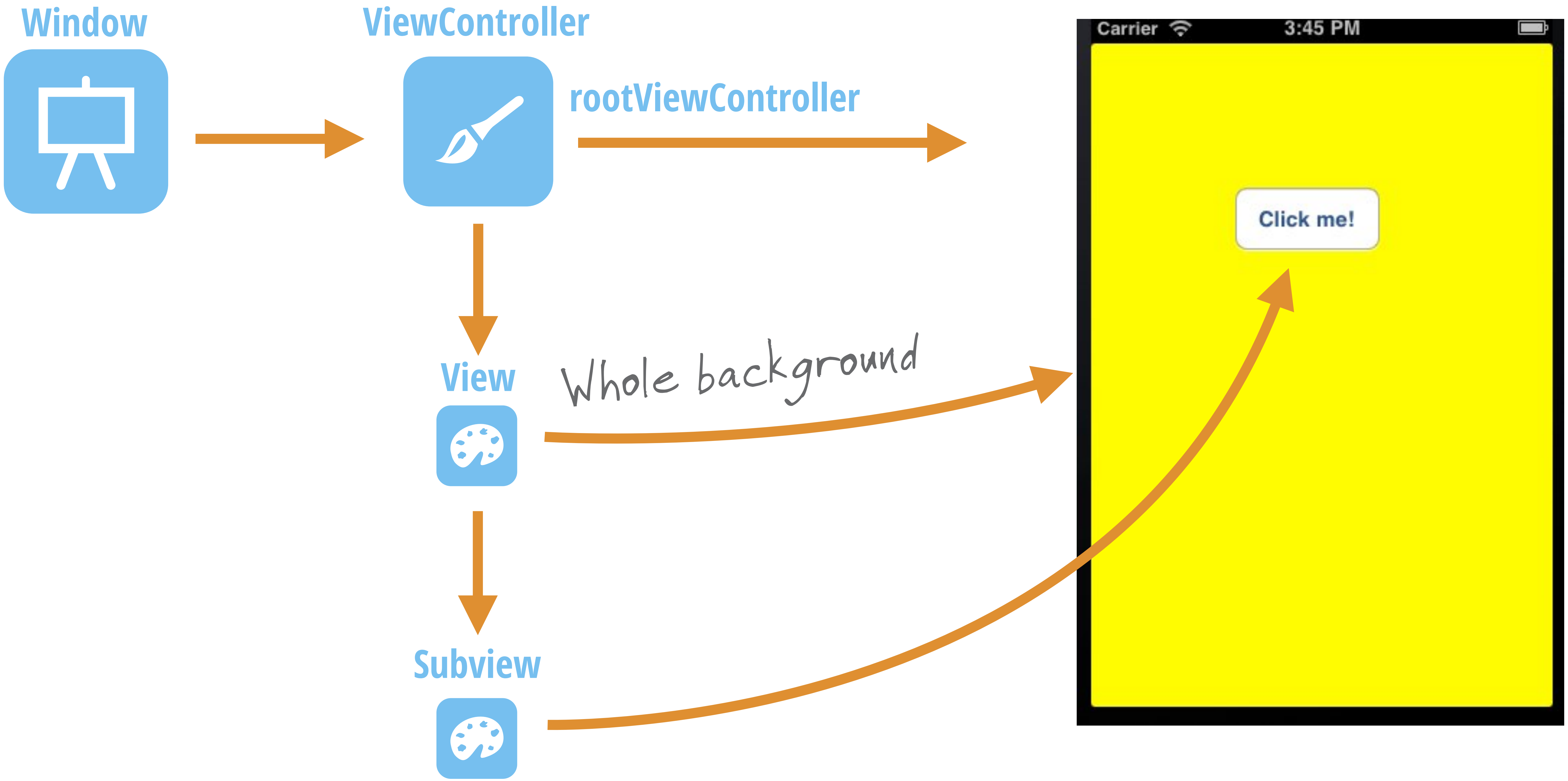
02 Creating a Label

03 Making the Button Work

04 Creating Multiple Buttons

05 Refactoring Our Code

Remember this?



Window

ViewController

rootViewController

View

Whole background

Subview

Create a click me button

ViewController.m

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.view.backgroundColor = [UIColor yellowColor];

    UIButton *firstButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
```

Create a new UIButton of type UIButtonTypeRoundedRect

```
    firstButton.frame = CGRectMake(100, 100, 100, 44);
```

Located at x = 100pts, y = 100pts, 100pts width, 44pts height

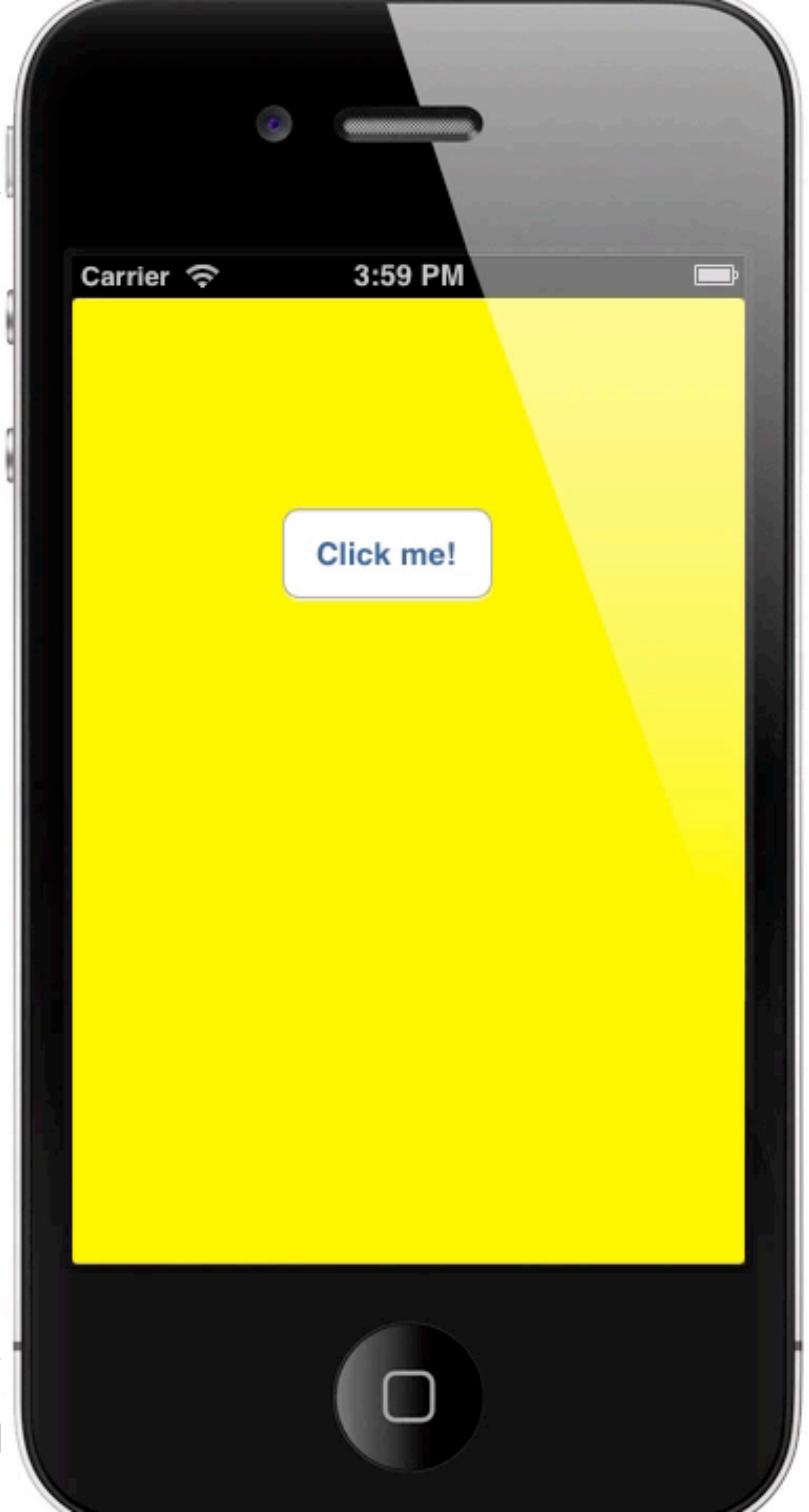
```
    [firstButton setTitle:@"Click me!" forState:UIControlStateNormal];
```

Set the button title, what it shows normally

```
    [self.view addSubview:firstButton];
}
```

Add button on top of our view

Yay, a button!



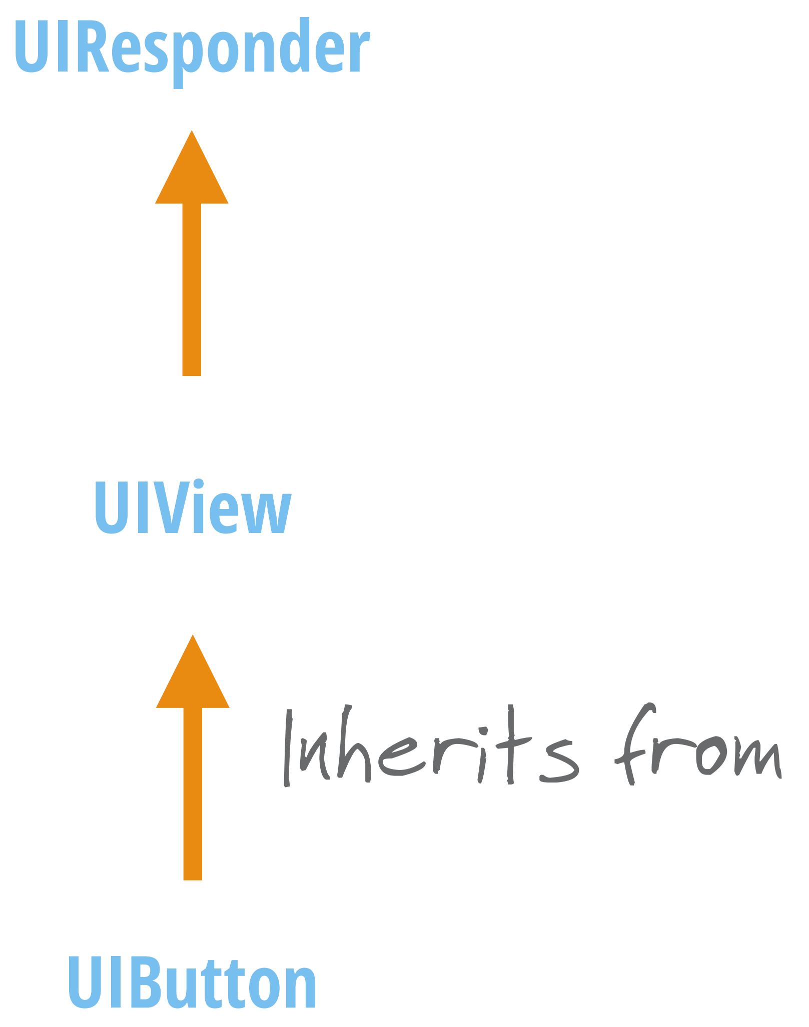
Create a click me button

ViewController.m

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.view.backgroundColor = [UIColor yellowColor];

    UIButton *firstButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
     Create a new UIButton
    firstButton.frame = CGRectMake(100, 100, 100, 44);
     Located at x = 100, y = 100, 100 width, 44 height
    [firstButton setTitle:@"Click me!" forState:UIControlStateNormal];
     Set the button title, what it shows normally
    [self.view addSubview:firstButton];
}    Add button on top of our view
```

Create a UIButton

```
UIButton *firstButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
```

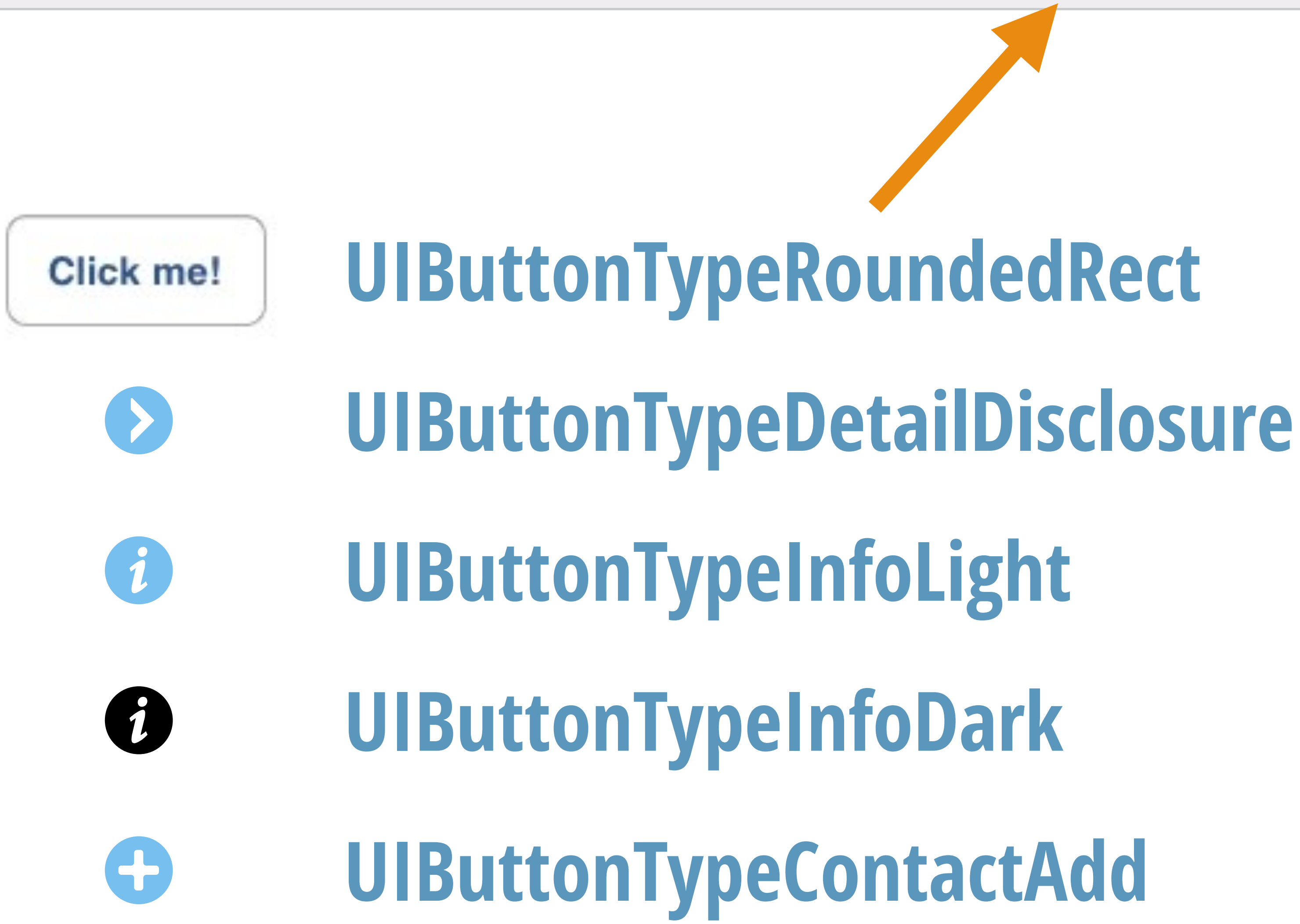


UIResponder

UIView

UIButton

Inherits from



UIButtonTypeRoundedRect

UIButtonTypeDetailDisclosure

UIButtonTypeInfoLight

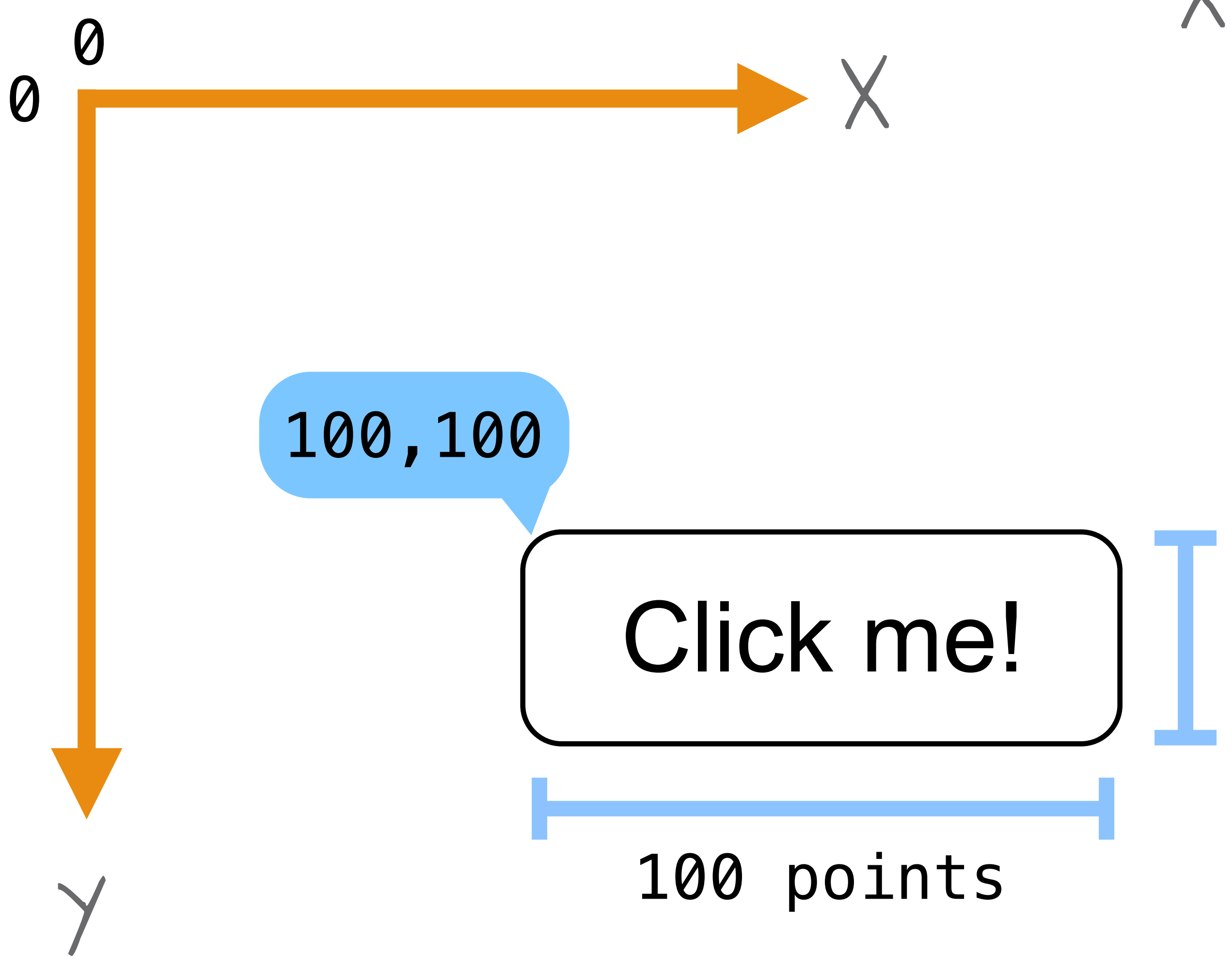
UIButtonTypeInfoDark

UIButtonTypeContactAdd

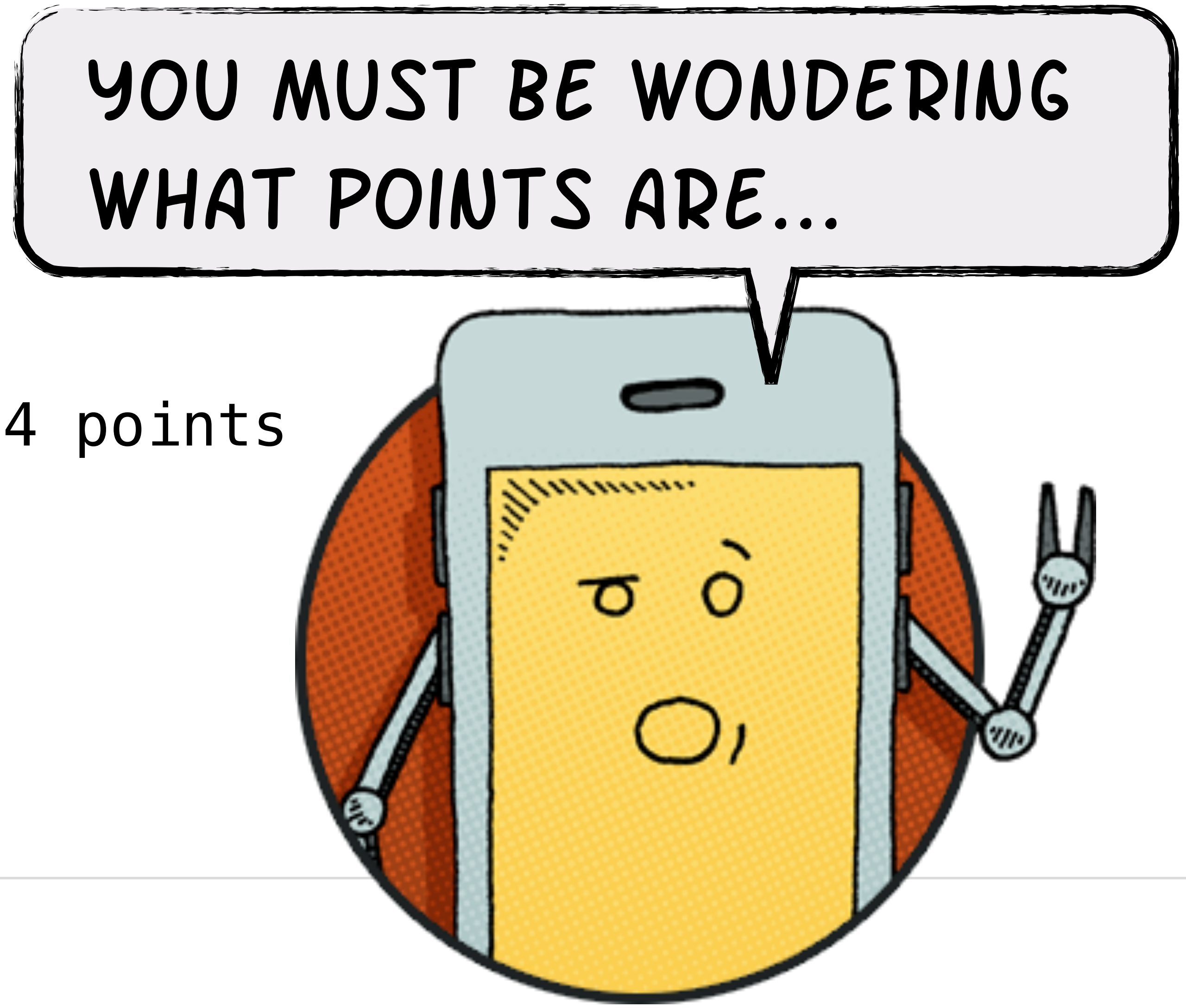
Set the position & size of the button

```
firstButton.frame = CGRectMake(100, 100, 100, 44);
```

X, Y, width, height



44 points



Understanding Points

	Points		Pixels
Non Retina iPhone 1 point = 1 pixel	□	320x480	□	320x480
iPhone 4 (Retina Screen) 1 point = 4 pixels	□	320x480	⊞	640×960
iPhone 5 (Retina Screen) 1 point = 4 pixels	□	320x568	⊞	640×1136

Set the button title

```
[firstButton setTitle:@"Click me!" forState:UIControlStateNormal];
```

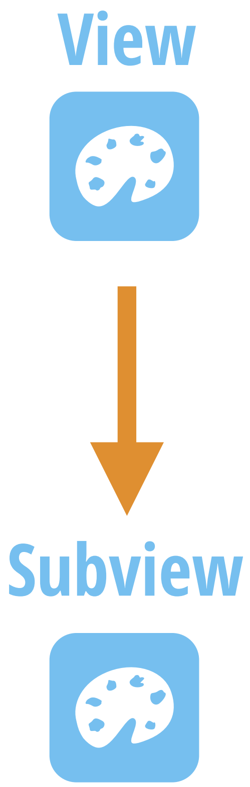
Sets the title shown on button

```
forState:UIControlStateHighlighted
```

Sets the title when pressed

Add the Button view as a subview

```
[self.view addSubview:firstButton];
```



View

Subview

Can be multiple subviews

Adding a UILabel (also inherits from UIView)

```
UILabel *firstLabel = [[UILabel alloc] initWithFrame:CGRectMake(50, 30, 200, 44)];
```

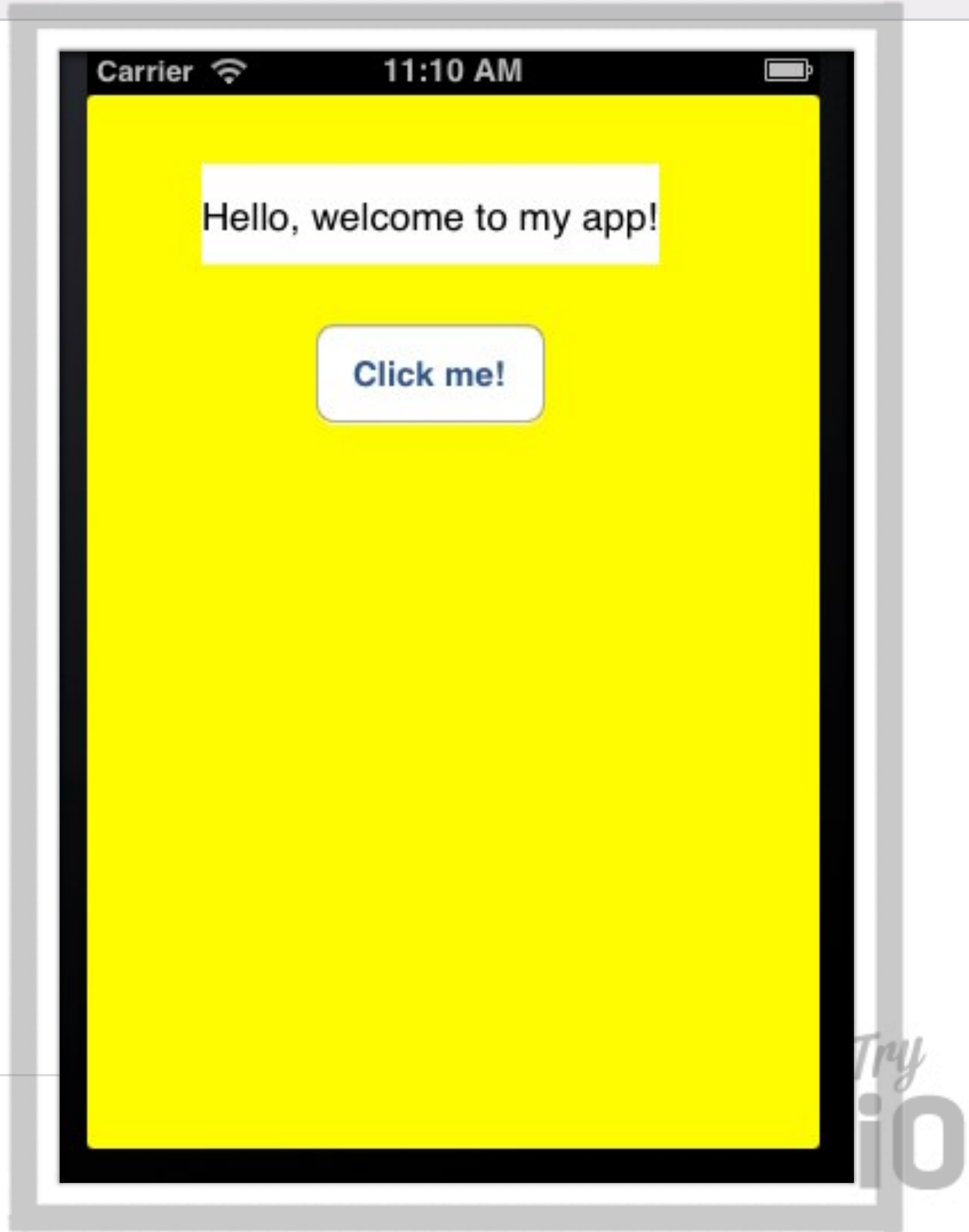
Create a new label with frame

```
firstLabel.text = @"Hello, welcome to my app!";
```

Set the label text

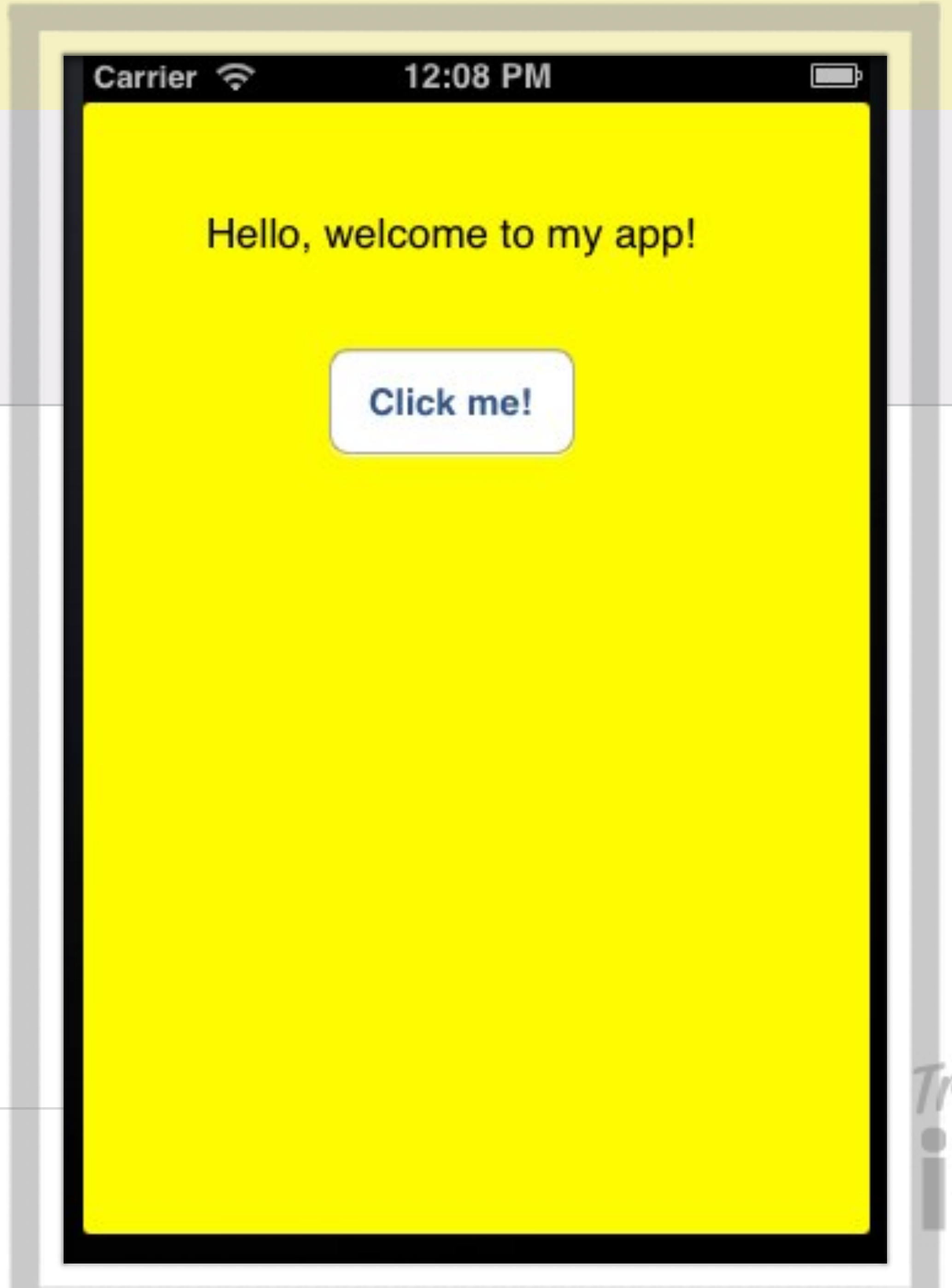
```
[self.view addSubview:firstLabel];
```

Add the label as a subview

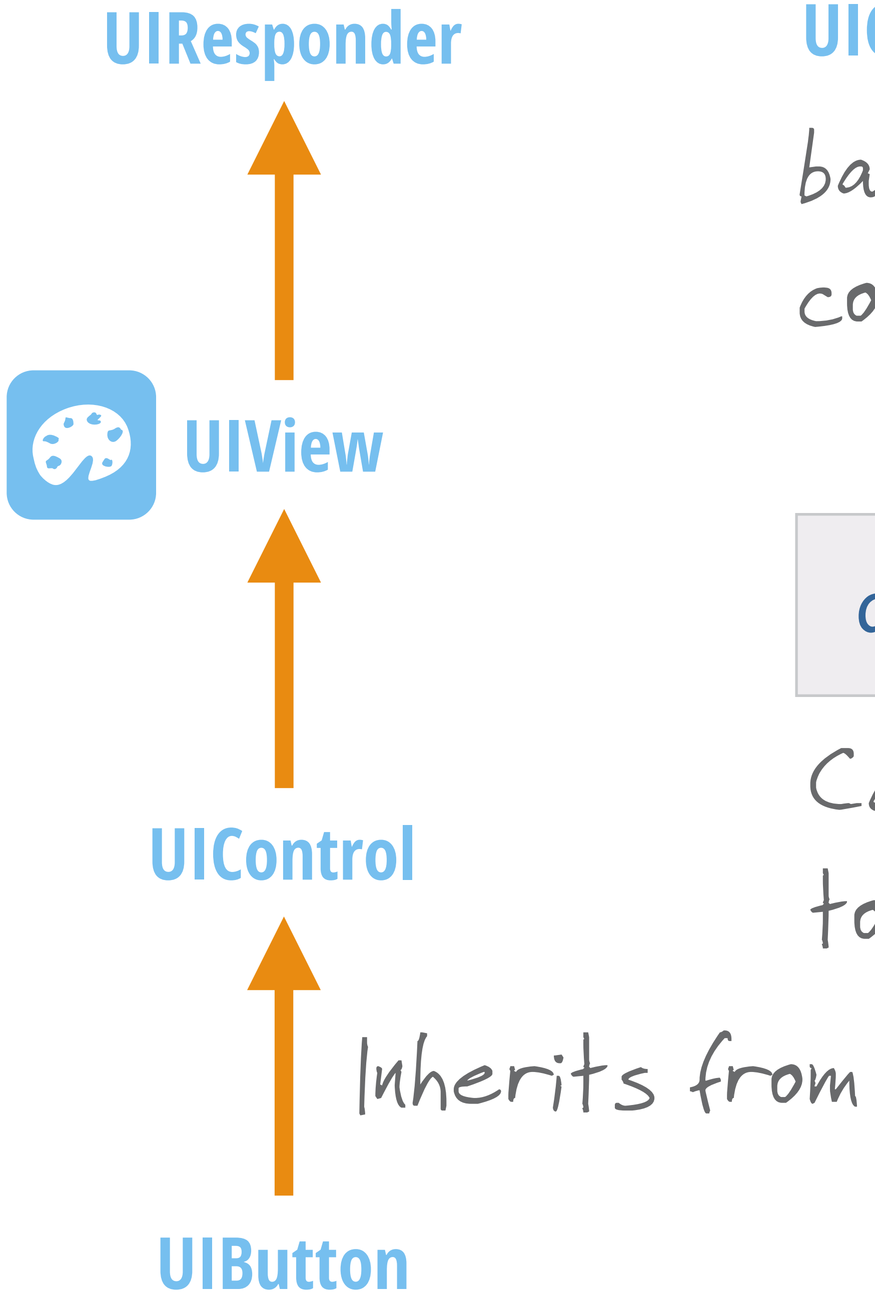


Coloring in the background

```
UILabel *firstLabel = [[UILabel alloc] initWithFrame:CGRectMake(50, 30, 200, 44)];

firstLabel.backgroundColor = [UIColor clearColor];

firstLabel.text = @"Hello, welcome to my app!";
[self.view addSubview:firstLabel];
```



How do we make the button work?



UIControl

base class for control objects which convey user intent

```
addTarget:action:forControlEvents:
```

Can send this to the UIButton object to listen for events

Adding Control Events

addTarget:action:forControlEvents:

```
- (void)addTarget:(id)target
            action:(SEL)action
 forControlEvents:(UIControlEvents)controlEvents;
```

instance method

target	the target object to which the message will be sent
action	the message to send to the target
controlEvents	the event to listen for

Adding Control Events

ViewController.m

```
- (void)viewDidLoad
{
    ...
    [firstButton addTarget:self
                    action:@selector(buttonPressed:)
          forControlEvents:UIControlEventTouchUpInside];
}

- (void)buttonPressed:(UIButton *)sender
{
    NSLog(@"Button pressed, sender: %@", sender);
    self.view.alpha = ((double)arc4random() / 0x100000000);
}
```

→ ViewController

Call `[self buttonPressed:firstButton];` when event triggered

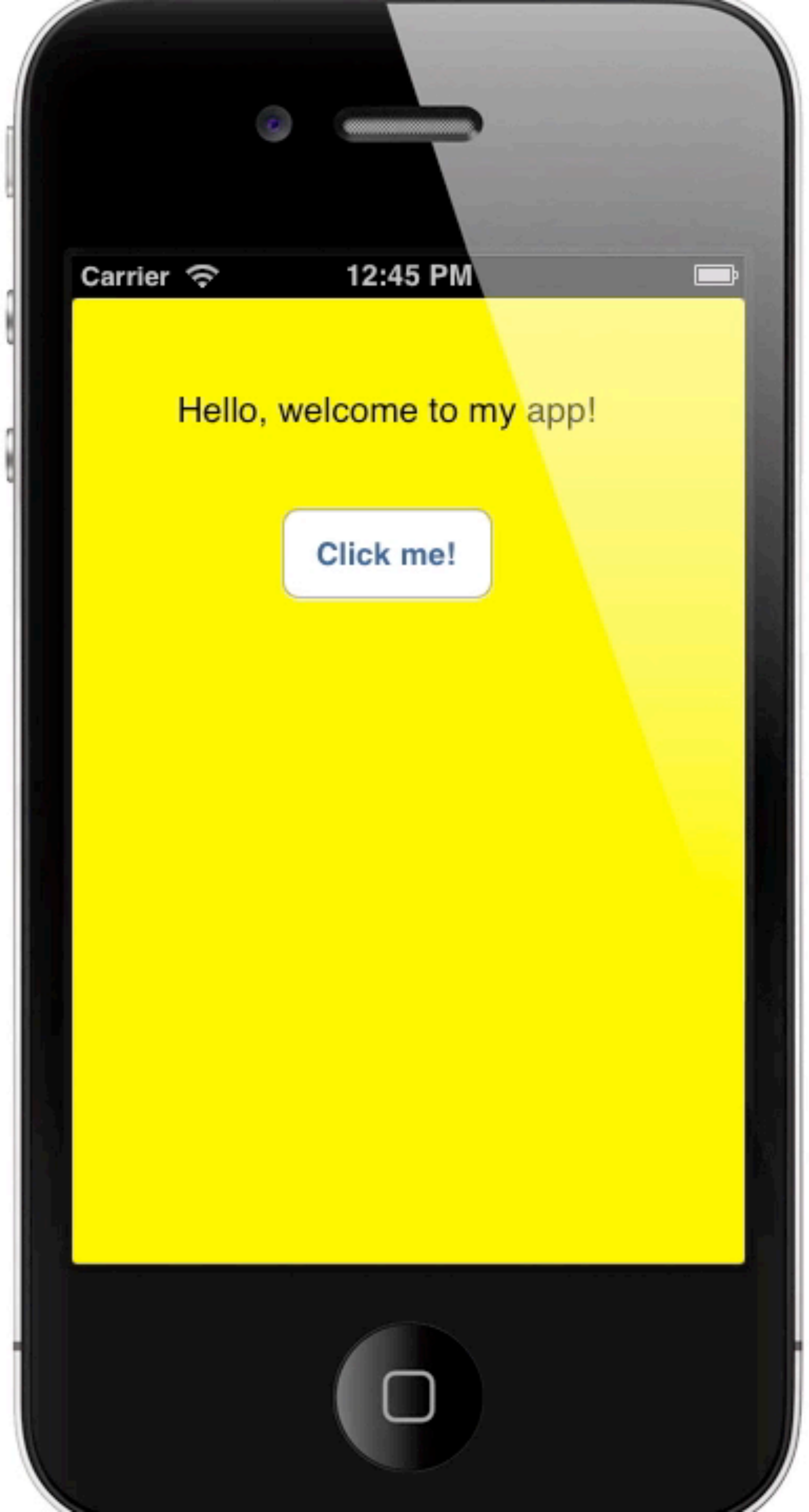
Adding Method to Header

ViewController.m

```
- (void)buttonPressed:(UIButton *)sender
{
    NSLog(@"Button pressed, sender: %@", sender);
    self.view.alpha = ((double)arc4random() / 0x100000000);
}
```

ViewController.h

```
#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

- (void)buttonPressed:(UIButton *)sender;

@end
```



Our button works!

```
Button pressed, sender: <UIRoundedRectButton:
0x7195870; frame = (100 100; 100 44); opaque = NO;
layer = <CALayer: 0x71959e0>>
```

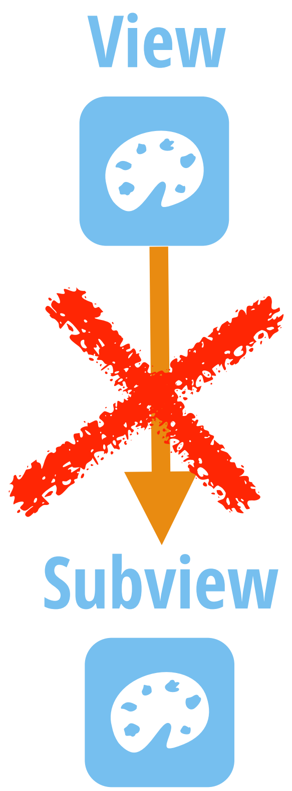
UIControl Events

```
UIControlEventTouchDown
UIControlEventTouchDownRepeat
UIControlEventTouchDragInside
UIControlEventTouchDragOutside
UIControlEventTouchDragEnter
UIControlEventTouchDragExit
UIControlEventTouchUpInside
UIControlEventTouchUpOutside
UIControlEventTouchCancel
UIControlEventValueChanged
UIControlEventEditingDidBegin
UIControlEventEditingChanged
UIControlEventEditingDidEnd
UIControlEventEditingDidEndOnExit
UIControlEventAllTouchEvents
UIControlEventAllEditingEvents
UIControlEventAllEvents
```

Removing a View

ViewController.m

```
- (void)buttonPressed:(UIButton *)sender
{
    NSLog(@"Button pressed, sender: %@", sender);
    self.view.alpha = ((double)arc4random() / 0x100000000);
    [sender removeFromSuperview];
}
```

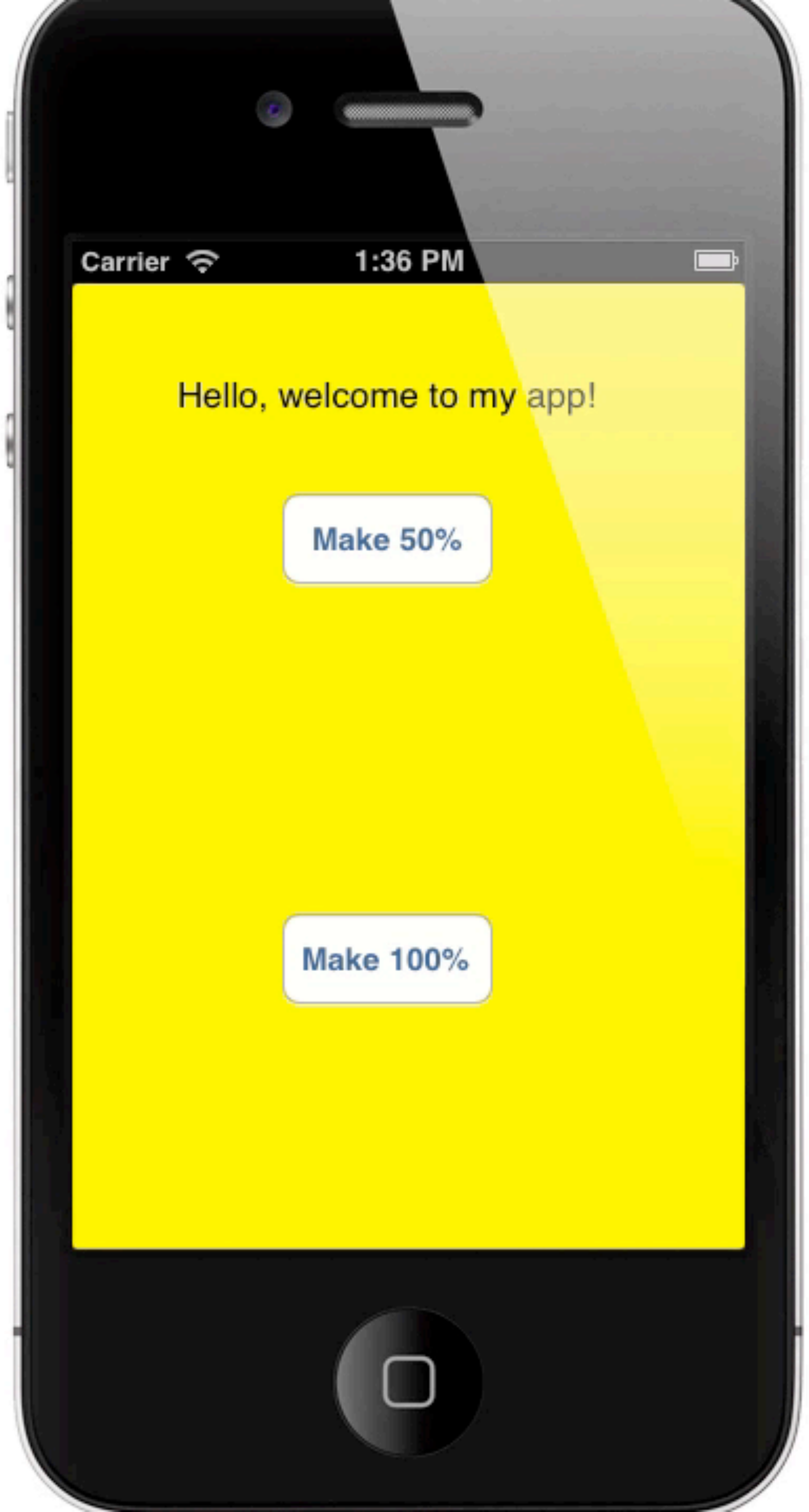


A Second Button

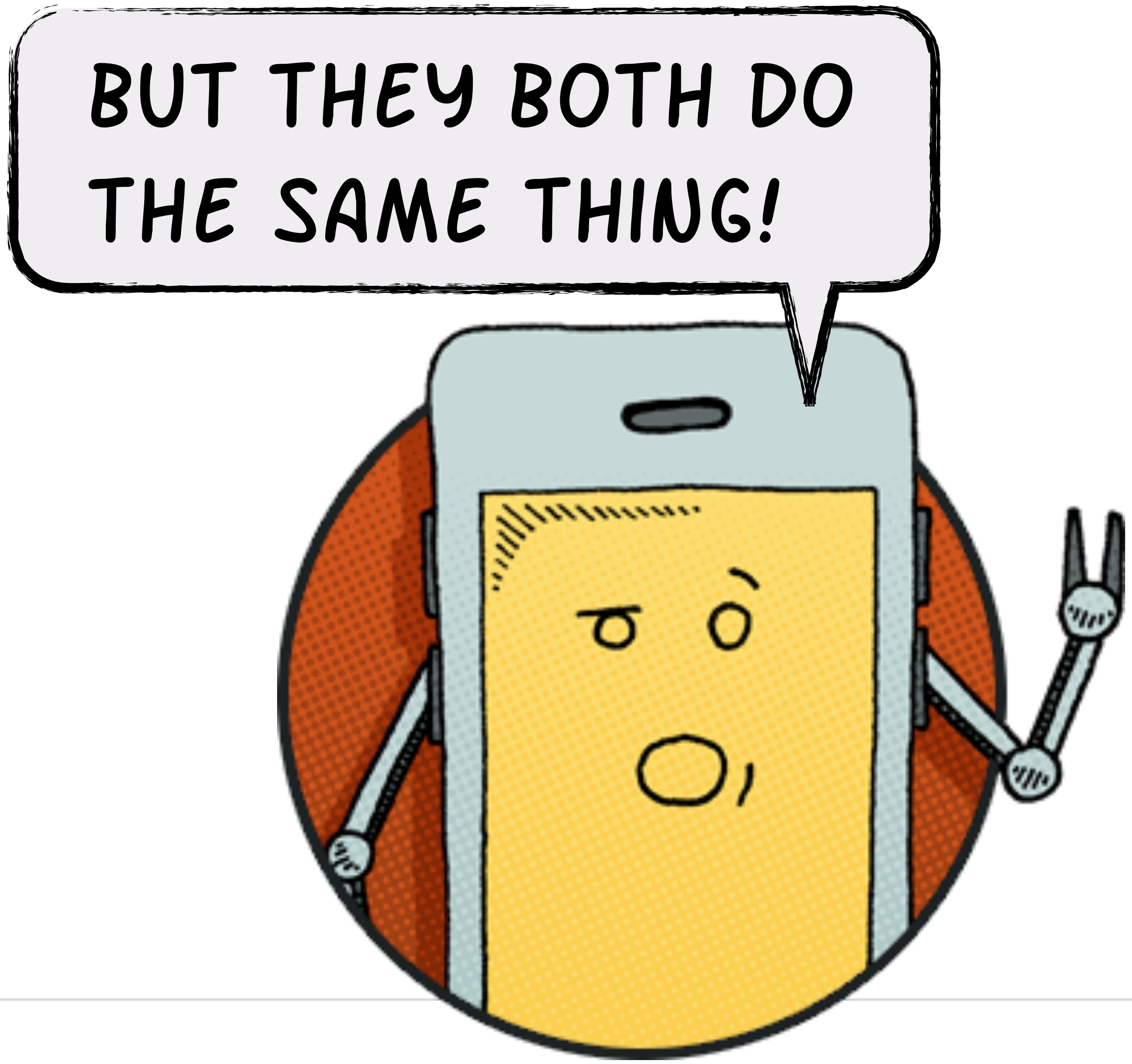
ViewController.m

```
UIButton *firstButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
firstButton.frame = CGRectMake(100, 100, 100, 44);
[firstButton setTitle:@"Make 50%" forState:UIControlStateNormal];
[self.view addSubview:firstButton];

[firstButton addTarget:self
                action:@selector(buttonPressed:)
      forControlEvents:UIControlEventTouchUpInside];

UIButton *secondButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
secondButton.frame = CGRectMake(100, 300, 100, 44);
[secondButton setTitle:@"Make 100%" forState:UIControlStateNormal];
[self.view addSubview:secondButton];

[secondButton addTarget:self
                 action:@selector(buttonPressed:)
       forControlEvents:UIControlEventTouchUpInside];
```

Now with two Buttons



Hello, welcome to my app!



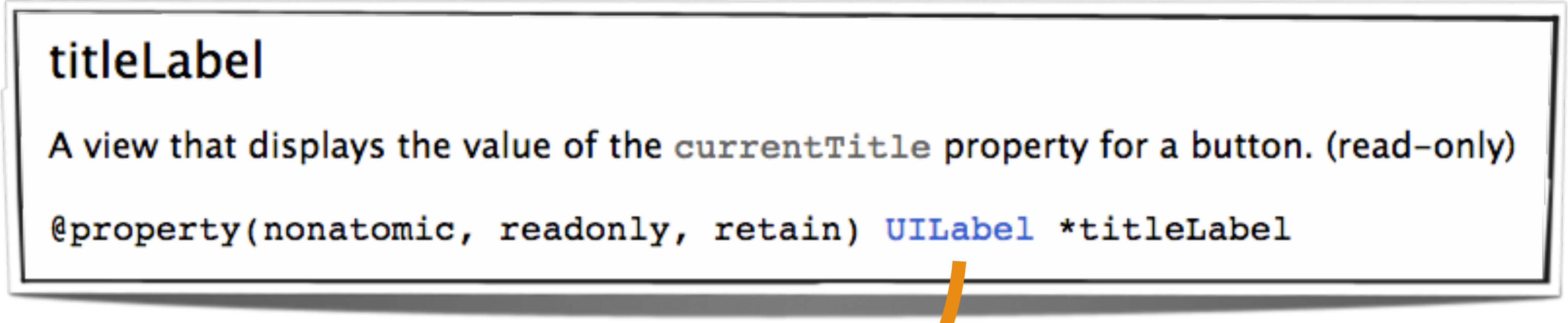
Completing alpha Functionality

ViewController.m

```
- (void)buttonPressed:(UIButton *)sender
{
    NSLog(@"Button pressed, sender: %@", sender);

    if (<is this the 50% button?>) {
        self.view.alpha = .5;
    } else {
        self.view.alpha = 1;
    }
}
```

Checking for the right button

Documentation for UIButton

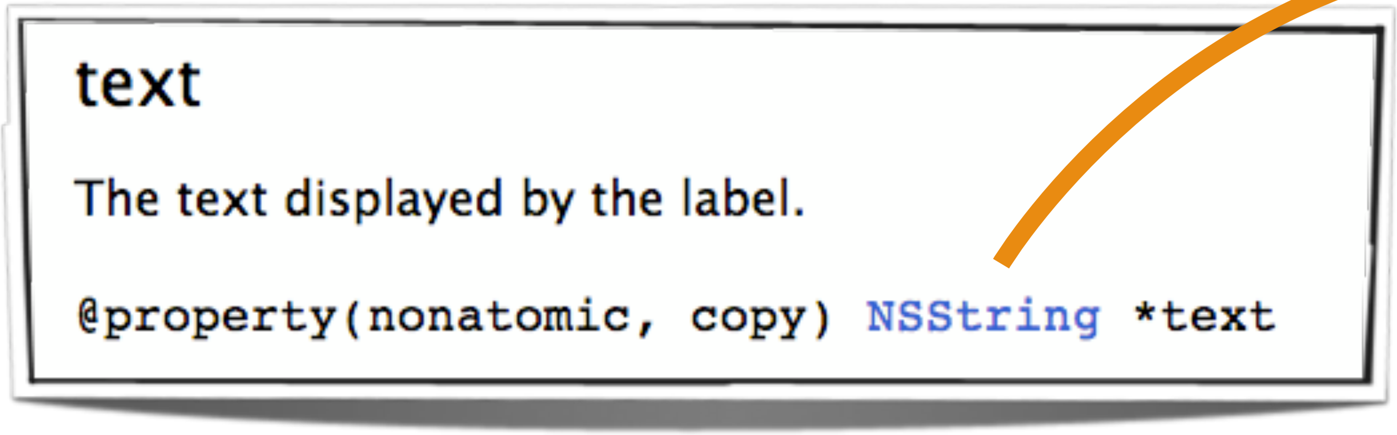


titleLabel

A view that displays the value of the `currentTitle` property for a button. (read-only)

```
@property(nonatomic, readonly, retain) UILabel *titleLabel
```

Documentation for UILabel

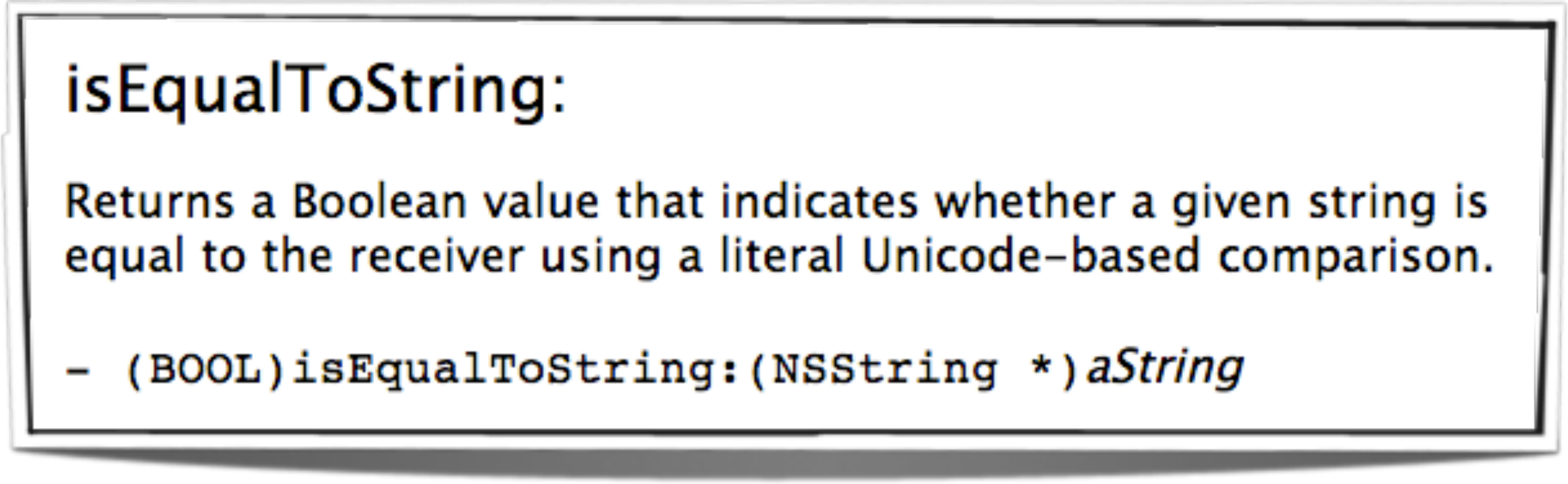


text

The text displayed by the label.

```
@property(nonatomic, copy) NSString *text
```

Documentation for NSString



isEqualToString:

Returns a Boolean value that indicates whether a given string is equal to the receiver using a literal Unicode-based comparison.

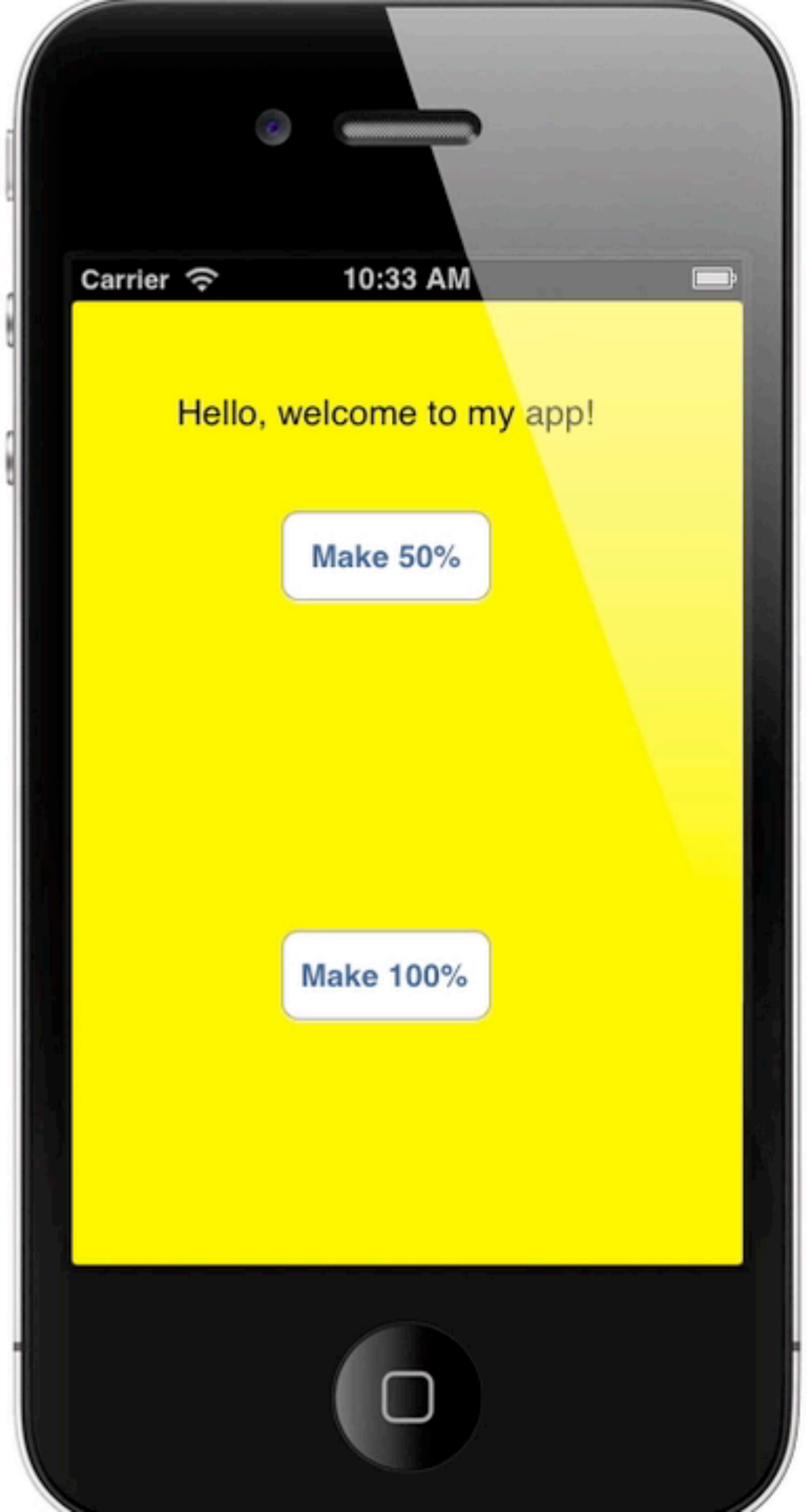
```
- (BOOL)isEqualToString:(NSString *)aString
```

```
if ([sender.titleLabel.text isEqualToString:@"Make 50%"]) {
```

Completing alpha Functionality

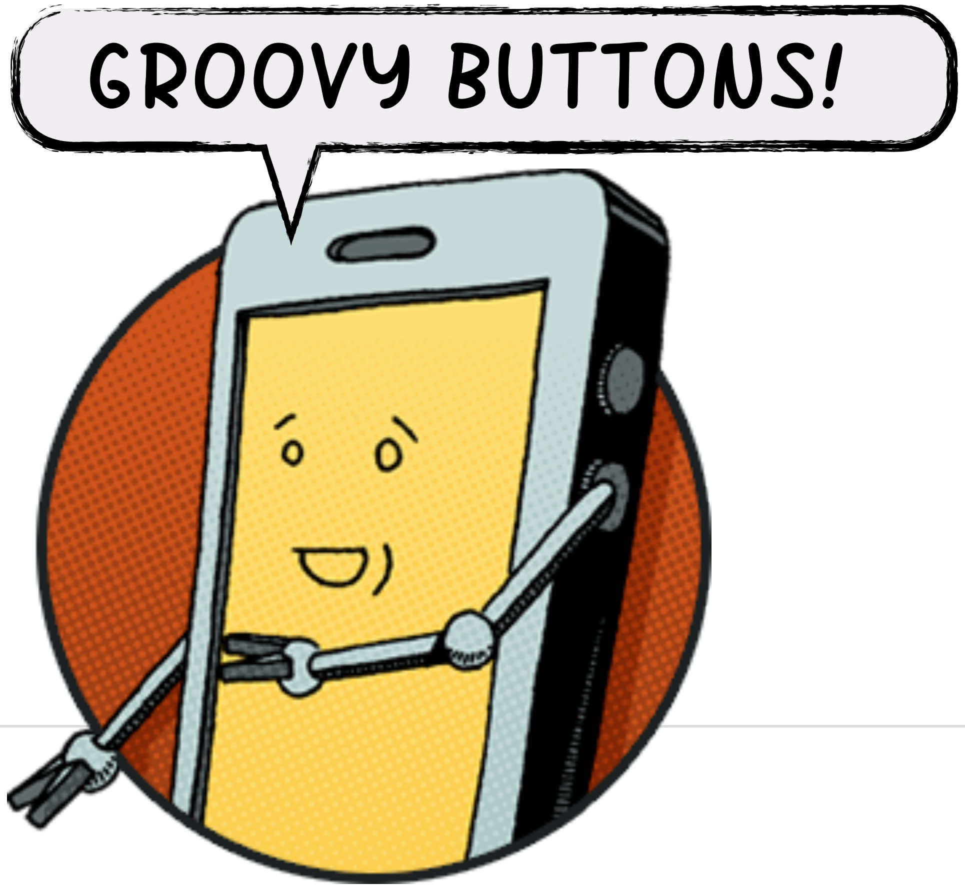
```
- (void)buttonPressed:(UIButton *)sender
{
    NSLog(@"Button pressed, sender: %@", sender);

    if ([sender.titleLabel.text isEqualToString:@"Make 50%"]) {
        self.view.alpha = .5;
    } else {
        self.view.alpha = 1;
    }
}
```

Now they work!



Carrier 10:33 AM

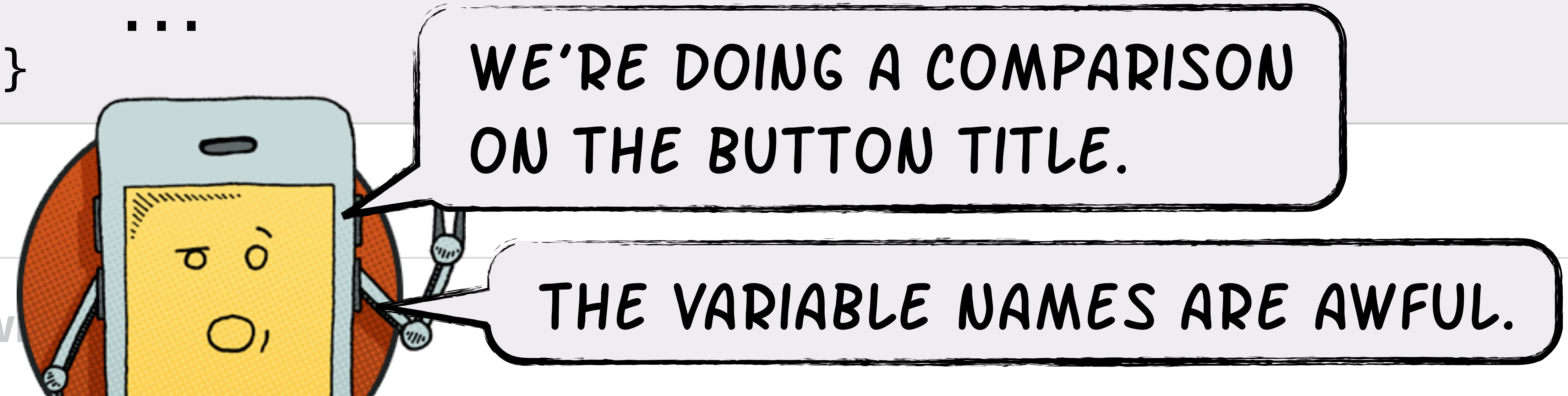
Hello, welcome to my app!



Improving this Code

ViewController.m

```
- (void)viewDidLoad
{
    UIButton *firstButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
    ...
    UIButton *secondButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
    ...
}
- (void)buttonPressed:(UIButton *)sender
{
    if ([sender.titleLabel.text isEqualToString:@"Make 50%"]) {
    ...
}
```


WE'RE DOING A COMPARISON ON THE BUTTON TITLE.
THE VARIABLE NAMES ARE AWFUL.

Creating new properties

ViewController.h

```
@interface ViewController : UIViewController

@property (weak, nonatomic) UIButton *fiftyPercentButton;

@property (weak, nonatomic) UIButton *hundredPercentButton;
```

ViewController.m

old code

```
- (void)viewDidLoad
{
    UIButton* firstButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
    firstButton.frame = CGRectMake(100, 100, 100, 44);
    [firstButton setTitle:@"Make 50%" forState:UIControlStateNormal];
    [self.view addSubview:firstButton];

    [firstButton addTarget:self
                    action:@selector(buttonPressed:)
          forControlEvents:UIControlEventTouchUpInside];
```

Updating the controller to use new property

ViewController.h

```
@interface ViewController : UIViewController

@property (weak, nonatomic) UIButton *fiftyPercentButton;

@property (weak, nonatomic) UIButton *hundredPercentButton;
```

ViewController.m

```
- (void)viewDidLoad
{
     self.fiftyPercentButton= [UIButton buttonWithType:UIButtonTypeRoundedRect];
     self.fiftyPercentButton.frame = CGRectMake(100, 100, 100, 44);
    [self.fiftyPercentButton  setTitle:@"Make 50%" forState:UIControlStateNormal];
    [self.view addSubview:self.fiftyPercentButton];

    [self.fiftyPercentButton addTarget:self
                               action:@selector(buttonPressed:)
                     forControlEvents:UIControlEventTouchUpInside];
```

Updating the buttonPressed conditional

ViewController.h

```
@interface ViewController : UIViewController

@property (weak, nonatomic) UIButton *fiftyPercentButton;

@property (weak, nonatomic) UIButton *hundredPercentButton;
```

ViewController.m

```
- (void)buttonPressed:(UIButton *)sender
{
    if ([sender.titleLabel.text isEqualToString:@"Make 50%"]) {
    ...
}
```

Updating the buttonPressed method

ViewController.h

```
@interface ViewController : UIViewController

@property (weak, nonatomic) UIButton *fiftyPercentButton;

@property (weak, nonatomic) UIButton *hundredPercentButton;
```

ViewController.m

```
- (void)buttonPressed:(UIButton *)sender
{
    if ([sender isEqual:self.fiftyPercentButton]) {
    ...
}
```