iPad
iPhone
iPod touch

iOS Development

By Dani Arnaout

[Twitter: dani_arnaout]
[Skype: dani_arnaout]
[Email: dani.arnaout@live.com]
[Youtube channel: programmingtutor0]

Version 1.0

Table of Contents:

I. UILabel:

(The following examples apply to a UILabel called myLabel)

- **text:**

  The text displayed by the label.

  ```
  myLabel.text = @”Text”;
  ```

- **textAlignment**

  The technique to use for aligning the text.

  ```
  [label setTextAlignment:UITextAlignmentLeft];
  ```

  also: UITextAlignmentRight
        UITextAlignmentCenter

- **textColor**

  The color of the text.

  ```
  myLabel.textColor=[UIColor blueColor];
  ```

- **backgroundColor**

  The color of the text.

  ```
  myLabel.backgroundColor=[UIColor redColor];
  ```

- **font**

  The font of the text.

  ```
  [myLabel setFont:[UIFont fontWithName:@"Arial" size:18]];
  ```

- **alfa**

  Set the alfa of the text.

  ```
  myLabel.alfa = 0.5;
  ```
  (This applies to all other UI elements too)

- **hidden**

  Hide or show the text.

  ```
  Mylabel.hidden = YES;
  ```
  (This applies to all other UI elements too)

II. UIButton:

(The following examples apply to a UIButton called myButton)

- **setTitle: forState:**

  The title used for specified state.

  ```
  [myButton setTitle:@”hello” forState:UIControlStateNormal];
  ```

  ```
  also: UIControlStateHighlighted
         UIControlStateDisabled
         UIControlStateSelected
  ```

- **setTitleColor: forState:**

  The color of the button’s title.

  ```
  [myButton setTitleColor:[UIColor BlueColor]
  forState:UIControlStateSelected];
  ```

- **setBackgroundImage: forState:**

  The image of the button’s background.

  ```
  myLabel.backgroundColor=[UIColor redColor];
  ```

- **titleLabel.font**

  The font of the button’s title.

  ```
  myButton.titleLabel.font = [UIFont fontWithName:@"Arial" size:18];
  ```

set images for states:

```
[button setImage:[UIImage imageNamed:@"normal.png"]
forState:UIControlStateNormal];
[button setImage:[UIImage imageNamed:@"pressed.png"]
forState:UIControlStateHighlighted];
```

III. UISegmentedControl:

(The following examples apply to a UISegmentedControl called mySegmentedControl)

```
if (mySegmentedControl.selectedSegmentIndex ==0)  NSLog(@"HI");
```

IV. UITextField:

(The following examples apply to a UITextField called myTextField)

- **Using a text field:**

  Connect an instance of UITextField to .h file then implement using the following code .

```
myTextField.text=@"hi";
myTextField.placeholder=@"Enter your name";
myTextField.clearButtonMode = UITextFieldViewModeAlways;

also:
UITextFieldViewModeNever
UITextFieldViewModeWhileEditing
UITextFieldViewModeUnlessEditing

Bool Attributes:
     myTextField.hasText;
     myTextField.highlighted;
     myTextField.hidden;
     myTextField.isEditing;
```

V. UISlider:

(The following examples apply to a UISlider called mySlider)

- **Creating a slider:**

  Connect an instance of UISlider to .h file then implement using the following code .

```
mySlider.minimumValue = 0.0;
mySlider.maximumValue = 100.0;

NSString *sliderValue = [NSString stringWithFormat:@"%f",mySlider.value];
```

VI. UISwitch:

(The following examples apply to a UISwitch called mySwitch)

```
if (mySwitch.on == YES) NSLog(@”Switch is ON”);
```

VII. UIActivityIndicator:

(The following examples apply to a UIActivityIndicator called myActivityIndicator)

start animating:

```
[myActivityIndicator startAnimating];
```

Stop animating:

```
[myActivityIndicator stopAnimating];
```

Check if it’s animating:

```
if (myActivityIndicator.isAnimating) NSLog(@"It is animating");
```

Change its style:

```
myActivityIndicator.activityIndicatorViewStyle =
UIActivityIndicatorViewStyleWhite;
```

```
also available: UIActivityIndicatorViewStyleWhiteLarge;
                 UIActivityIndicatorViewStyleGrey;
```

VIII. UIProgressView:

(The following examples apply to a UIProgressView called myProgressView)

set progress value:

```
[myProgressView setProgress:1.0 animated:YES];
```

get the value of the progressView and save it in a

```
float a= myProgressView.progress;
```

VIII. UIPageControl:

Add this to your .h

```
@interface ScrollingViewController : UIViewController
<UIScrollViewDelegate>
{
    BOOL pageControlIsChangingPage;
}
@property (retain, nonatomic) IBOutlet UIScrollView *scrollView;

@property (retain, nonatomic) IBOutlet UIPageControl *pageControl;

/* for pageControl */
- (IBAction)changePage:(id)sender;

/* internal */
- (void)setupPage;

@end
```

Add this to your .m

```
- (void)setupPage
{
scrollView.delegate = self;

[self.scrollView setBackgroundColor:[UIColor blackColor]];
[scrollView setCanCancelContentTouches:NO];

scrollView.indicatorStyle = UIScrollViewIndicatorStyleWhite;
scrollView.clipsToBounds = YES;
scrollView.scrollEnabled = YES;
scrollView.pagingEnabled = YES;

NSUInteger nimages = 0;
CGFloat cx = 0;
for (; ; nimages++) {
NSString *imageName = [NSString stringWithFormat:@"image%d.jpg", (nimages
+ 1)];
UIImage *image = [UIImage imageNamed:imageName];
if (image == nil) {
           break;}
UIImageView *imageView = [[UIImageView alloc] initWithImage:image];
CGRect rect = imageView.frame;
rect.size.height = image.size.height;
rect.size.width = image.size.width;
rect.origin.x = ((scrollView.frame.size.width - image.size.width) / 2) +
cx;
rect.origin.y = ((scrollView.frame.size.height - image.size.height) / 2);
imageView.frame = rect;
[scrollView addSubview:imageView];
[imageView release];
cx += scrollView.frame.size.width;
     }
self.pageControl.numberOfPages = nimages;
[scrollView setContentSize:CGSizeMake(cx, [scrollView
bounds].size.height)];
}

- (void)scrollViewDidScroll:(UIScrollView *)_scrollView
{
    if (pageControlIsChangingPage) {
        return;
    }

     /*
      *    We switch page at 50% across
      */
CGFloat pageWidth = _scrollView.frame.size.width;
int page = floor((_scrollView.contentOffset.x - pageWidth / 2) /
pageWidth) + 1;
    pageControl.currentPage = page;
}

- (void)scrollViewDidEndDecelerating:(UIScrollView *)_scrollView
{
    pageControlIsChangingPage = NO;
}

#pragma mark -
#pragma mark PageControl stuff
- (IBAction)changePage:(id)sender
{
     /*    Change the scroll view*/
    CGRect frame = scrollView.frame;
    frame.origin.x = frame.size.width * pageControl.currentPage;
    frame.origin.y = 0;
    [scrollView scrollRectToVisible:frame animated:YES];
     /*
      *    When the animated scrolling finishings,
scrollViewDidEndDecelerating will turn this off
      */
    pageControlIsChangingPage = YES;
}
@end
```

IX. UIStepper:

(The following examples apply to a UIStepper called myStepper)

- **Using a stepper:**

  Connect an instance of UIStepper to .h file then implement using the following code .

  You can set Current, Min, Max, & Step values for your stepper from the properties menu.

  To get the value of myStepper:

  ```
  myStepper.value (return type : DOUBLE)
  ```

  To get the value in the form of a string:

  ```
  NSString *hi = [NSString stringWithFormat:@"%i",(int)myStepper.value];
  ```

  ```
  // Set min and max
    [myStepper setMinimumValue:0];
    [myStepper setMaximumValue:99];

  // To change the increment value for each step
    // (default is 1)
    [myStepper setStepValue:10];

  // Set current value
  [myStepper Value:0];
  ```

X. UITableViews:

How to create a static table:

1) Create a view that sublasses from UITableView (you will get extra methods ready for you to implement)

2)Create a global Array in .h file:

```
@interface tables : UITableViewController
  { NSArray *myList;}
  @end
```

3)Initialize the array in the your viewDidLoad:

```
myList = [NSArray arrayWithObjects:@"one",@"two",@"three",nil];
```

4)We have to implement three methods

a)number of sections (usually 1)

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
// Return the number of sections.
    return 1;
}
```

b) number of rows in each section (we calculate the elements of the array by using a ready method).

```
- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section
{
// Return the number of rows in the section.
    return [myList count];
}
```

c)Assigning the array elements as the row cells titles

```
- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
static NSString *CellIdentifier = @"Cell";

UITableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
 cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:CellIdentifier];
    }
 cell.textLabel.text = [myList objectAtIndex:[indexPath row]];
    return cell;
```

```
      }
```

TABLE VIEWS: EXTRA HELPFUL METHODS:

```
/*
// Override to support conditional editing of the table view.
- (BOOL)tableView:(UITableView *)tableView
canEditRowAtIndexPath:(NSIndexPath *)indexPath
{
    // Return NO if you do not want the specified item to be editable.
    return YES;
}
*/

/*
// Override to support editing the table view.
- (void)tableView:(UITableView *)tableView
commitEditingStyle:(UITableViewCellEditingStyle)editingStyle
forRowAtIndexPath:(NSIndexPath *)indexPath
{
    if (editingStyle == UITableViewCellEditingStyleDelete) {
        // Delete the row from the data source
        [tableView deleteRowsAtIndexPaths:[NSArray
arrayWithObject:indexPath] withRowAnimation:UITableViewRowAnimationFade];
    }
    else if (editingStyle == UITableViewCellEditingStyleInsert) {
        // Create a new instance of the appropriate class, insert it into
the array, and add a new row to the table view
    }
}
*/

/*
// Override to support rearranging the table view.
- (void)tableView:(UITableView *)tableView moveRowAtIndexPath:(NSIndexPath
*)fromIndexPath toIndexPath:(NSIndexPath *)toIndexPath
{
}
*/

/*
// Override to support conditional rearranging of the table view.
- (BOOL)tableView:(UITableView *)tableView
canMoveRowAtIndexPath:(NSIndexPath *)indexPath
{
    // Return NO if you do not want the item to be re-orderable.
    return YES;
}
*/
```

```
#pragma mark - Table view delegate

- (void)tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    // Navigation logic may go here. Create and push another view
controller.
    /*
     <#DetailViewController#> *detailViewController =
[[<#DetailViewController#> alloc] initWithNibName:@"<#Nib name#>"
bundle:nil];
     // ...
     // Pass the selected object to the new view controller.
     [self.navigationController pushViewController:detailViewController
animated:YES];
     */
}
```

XI. UIImageView:

(The following examples apply to a UIImageView called myImage)

make sure you use .png images

```
myImage.image = [UIImage imageNamed:@"image1"];
```

XII. UITextView:

(The following examples apply to a UITextView called myTextView)

```
   myTextView.editable = NO;
   myTextView.text = @"HI";
```

(This is used for multiple lines of text with scrollable ability. So if the text is too large for the screen, you can scroll down to get some more space for a new text)

XIII. UIWebView:

(The following examples apply to a UIWebView called myWebView)

- **Displaying a web page:**

  Connect an instance of UIWebView to .h file then implement using the following code .

```
NSURL *url = [NSURL URLWithString:@"http://www.bau.edu.lb"];
NSURLRequest *req = [NSURLRequest requestWithURL:url];
[webView loadRequest:req];


//GET URL FROM WEBVIEW:
NSString *currentURL= myWebView.request.URL.absoluteString;

//Get source code of the page:
NSString *source = [webView
stringByEvaluatingJavaScriptFromString:

@"document.getElementsByTagName('html')[0].outerHTML"];
```

XIV. MKMapView:

(The following examples apply to a MKMapView called mapView)

- **Displaying a map:**

  Import the Map framework.
  Connect an instance of MKMapView to .h file then implement using the following code .
  Add the following line to your .h file:

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

//Create region (Get coordinated from Google Maps)
    mapView.mapType= MKMapTypeSatellite;
    MKCoordinateRegion newRegion;
    newRegion.center.latitude = 33.675435;
    newRegion.center.longitude = 35.466281;
    newRegion.span.latitudeDelta = 0.002411;
    newRegion.span.longitudeDelta = 0.004576;

//Display the create ‘newRegion’ on the view
   [self.mapView setRegion:newRegion animated:YES];
```

- **Placing a pin on the map:**

```
//Create coordinate then assign it to a pin
CLLocationCoordinate2D coordinate;
    coordinate.latitude = 33.675435;
    coordinate.longitude = 35.466281;
```

```
//Create Pin
MKPointAnnotation *annotation = [[MKPointAnnotation alloc] init];

//Assign the coordinates we’ve created to the pin
    [annotation setCoordinate:coordinate];
//Give the pin a title
  [annotation setTitle:@"My Pin"];
//Place pin on the map
[self.mapView addAnnotation:annotation];
```

- **Disclosure button beside the pin’s title:**

```
- (MKAnnotationView *)mapView:(MKMapView *)mapView viewForAnnotation:(id
<MKAnnotation>)annotation {
    MKPinAnnotationView *pinView = (MKPinAnnotationView *)[mapView
dequeueReusableAnnotationViewWithIdentifier:@"pinView"];
    if (!pinView) {
        pinView = [[MKPinAnnotationView alloc] initWithAnnotation:annotation
reuseIdentifier:@"pinView"];
        pinView.pinColor = MKPinAnnotationColorRed;
        pinView.animatesDrop = YES;
        pinView.canShowCallout = YES;

        UIButton *rightButton = [UIButton
buttonWithType:UIButtonTypeDetailDisclosure];
        pinView.rightCalloutAccessoryView = rightButton;
    } else {
        pinView.annotation = annotation;
    }
    return pinView;
}


//IMPLEMENT THIS ONE (WHEN DISCLOSURE BUTTON IS PRESSED):
- (void)mapView:(MKMapView *)mapView annotationView:(MKAnnotationView *)view
calloutAccessoryControlTapped:(UIControl *)control{
      //  myLabel.text=@"hi";
 [[UIApplication sharedApplication] openURL:[NSURL URLWithString:
@"http://www.google.se"]];
}
```

XV. UIDatePicker:

(The following examples apply to a UIDatePicker called myDatePicker)

```
myDatePicker.datePickerMode = UIDatePickerModeDate;

   UIDatePickerModeTime,
   UIDatePickerModeDate,
   UIDatePickerModeDateAndTime,
   UIDatePickerModeCountDownTimer
```

```
//To set a specific format for a date
NSDateFormatter *format = [[NSDateFormatter alloc] init];
[format setDateFormat:@"MM-dd-yyyy"];
//   To create a date from string with specified format:
NSDate *date = [format dateFromString:@"1-1-2011"];
//   To get date with specified format
NSString *dateString = [format stringFromDate:date];
// To set the datePicker to a specific date
[myDatePicker setDate:date animated:YES];
```

To find the difference between two dates:

```
NSTimeInterval interval = [endDay timeIntervalSinceDate:startDay];
```

XVI. GestureRecognizer:

Types of gestures:

UITapGestureRecognizer
UIRotationGestureRecognizer
UIPanGestureRecognizer
UISwipeGestureRecognizer
UIPinchGestureRecognizer

To create any kind of gestures, you have to follow the following steps:

1) **Drag an instance of your gesture recognizer from the library and place it on top of your object receiving the gesture.**
2) **Make sure that your object is both user interaction and multi touch enabled.**
3) **Add the following code in .h file and implement it in the .m file.**
   - (IBAction)handlePan:(UIPanGestureRecognizer *)recognizer;
4) **Control drag from your recognizer to the MainViewController in the left side panel, and choose the appropriate method (usually called selector).**

The following are the different methods to handle all recognizers types. (declare in .h & implement in .m)

```
- (IBAction)handlePan:(UIPanGestureRecognizer *)recognizer {

    CGPoint translation = [recognizer translationInView:self.view];
    recognizer.view.center = CGPointMake(recognizer.view.center.x + 
translation.x,
                                         recognizer.view.center.y + 
translation.y);
    [recognizer setTranslation:CGPointMake(0, 0) inView:self.view];

      }
```

TOP STOP SMOOTHLY WITH DECREASING VELOCITY ADD THIS:

```
if (recognizer.state == UIGestureRecognizerStateEnded) {

    CGPoint velocity = [recognizer velocityInView:self.view];
    CGFloat magnitude = sqrtf((velocity.x * velocity.x) + (velocity.y * 
velocity.y));
    CGFloat slideMult = magnitude / 200;
    NSLog(@"magnitude: %f, slideMult: %f", magnitude, slideMult);
```

```
 float slideFactor = 0.1 * slideMult; // Increase for more of a slide
    CGPoint finalPoint = CGPointMake(recognizer.view.center.x +
(velocity.x * slideFactor),
                                     recognizer.view.center.y +
(velocity.y * slideFactor));
    finalPoint.x = MIN(MAX(finalPoint.x, 0), self.view.bounds.size.width);
    finalPoint.y = MIN(MAX(finalPoint.y, 0),
self.view.bounds.size.height);

    [UIView animateWithDuration:slideFactor*2 delay:0
options:UIViewAnimationOptionCurveEaseOut animations:^{
        recognizer.view.center = finalPoint;
    } completion:nil];

      }
```

PINCH:

```
- (IBAction)handlePinch:(UIPinchGestureRecognizer *)recognizer {
    recognizer.view.transform =
CGAffineTransformScale(recognizer.view.transform, recognizer.scale,
recognizer.scale);
    recognizer.scale = 1;
}
```

Rotate:

```
- (IBAction)handleRotate:(UIRotationGestureRecognizer *)recognizer {
    recognizer.view.transform =
CGAffineTransformRotate(recognizer.view.transform, recognizer.rotation);
    recognizer.rotation = 0;
      }
```

Simultaneous Gesture Recognizers

Open up ViewController.h and mark the class as implementing UIGestureRecognizerDelegate as shown below:

```
@interface ViewController : UIViewController <UIGestureRecognizerDelegate>
```

Then switch to ViewController.m and implement one of the optional methods **you can override:**

```
- (BOOL)gestureRecognizer:(UIGestureRecognizer *)gestureRecognizer
 shouldRecognizeSimultaneouslyWithGestureRecognizer:(UIGestureRecognizer *)
otherGestureRecognizer {
    return YES;
}
```

Next, open MainStoryboard.storyboard, and for each gesture recognizer connect its delegate outlet to the view controller.

XVI. NSTimer:

```
- (IBAction)startTimer:(id)sender {
    NSTimer  *timer;
    timer= [NSTimer scheduledTimerWithTimeInterval:0.1 target:self
selector:@selector(moreprogress) userInfo:nil repeats:YES ];
}

-(void) moreprogress{
    // Set whatever you want to executed each interval of time
}
```

XVIII. UIAlertView:

(The following examples apply to a UIAlertView called alert)

- **Creating and displaying an alert:**
  Include this code inside a specific method to trigger this alert on the screen .

```
UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@”Hi”
                                          message:@”This is a msg”
                                          delegate:self
                                          cancelButtonTitle:@"OK"
                                          otherButtonTitles: nil];

  [alert show];
```

Dismissing alert with a forced selected index (0 for cancel)

```
[myAlert dismissWithClickedButtonIndex:0 animated:NO];
```

Executing some code depending on the user choice:

```
 - (void)alertView:(UIAlertView *)alertView
clickedButtonAtIndex:(NSInteger)buttonIndex { NSLog([NSString
stringWithFormat:@“%d”, buttonIndex]);
}
```

XIX. UIActionSheet:

add this to the .h :

```
@interface MainViewController : UIViewController <UIActionSheetDelegate> )

- (IBAction)hehe:(id)sender {
    UIActionSheet *actionSheet = [[UIActionSheet alloc]
initWithTitle:@"This is an action sheet"
delegate:self
cancelButtonTitle:@"Cancel"
destructiveButtonTitle:@"Predicted answer"
otherButtonTitles:@"Second choice",@"Third choice",nil];

[actionSheet showInView:self.view];
```

// What to do after the user selects an option

```
- (void) actionSheet: (UIActionSheet *)actionSheet
didDismissWithButtonIndex:(NSInteger)buttonIndex{
    switch (buttonIndex) {
        case 0:
            NSLog(@"Predicted answer pressed");
            break;
        case 1:
            NSLog(@"Second choice pressed");
            break;
        case 2:
            NSLog(@"Third choice pressed");
            break;
        case 3:
            NSLog(@"cancel button pressed");
            break;

        default:
            break;
    }
}
```

XX. Audio:

To play an audio file:

Insert audio framework first.

Add this to your .h:

```
#import <AVFoundation/AVFoundation.h>
@property (strong) AVAudioPlayer * chompPlayer;
```

Add this to your .m:

```
- (AVAudioPlayer *)loadWav:(NSString *)filename {
    NSURL * url = [[NSBundle mainBundle] URLForResource:filename
withExtension:@"wav"];
    NSError * error;
    AVAudioPlayer * player = [[AVAudioPlayer alloc]
initWithContentsOfURL:url error:&error];
    if (!player) {
        NSLog(@"Error loading %@: %@", url,
error.localizedDescription);
    } else {
        [player prepareToPlay];
    }
    return player;
}



- (void)viewDidLoad
{
    [super viewDidLoad];

    self.chompPlayer = [self loadWav:@"mySong"];
    [self.chompPlayer play];

      }
```

XXI. Recording:

Add this to your .h:

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

@interface recordViewController : UIViewController
<AVAudioRecorderDelegate, AVAudioPlayerDelegate>
{
    AVAudioRecorder *audioRecorder;
    AVAudioPlayer *audioPlayer;
    UIButton *playButton;
    UIButton *recordButton;
    UIButton *stopButton;
}
@property (nonatomic, retain) IBOutlet UIButton *playButton;
@property (nonatomic, retain) IBOutlet UIButton *recordButton;
@property (nonatomic, retain) IBOutlet UIButton *stopButton;
-(IBAction) recordAudio;
-(IBAction) playAudio;
-(IBAction) stop;
      @end
```

Add this to your .m:

```
-(void) recordAudio
{
    if (!audioRecorder.recording)
    {
        playButton.enabled = NO;
        stopButton.enabled = YES;
        [audioRecorder record];
    }
}
-(void)stop
{
    stopButton.enabled = NO;
    playButton.enabled = YES;
    recordButton.enabled = YES;

    if (audioRecorder.recording)
    {
        [audioRecorder stop];
    } else if (audioPlayer.playing) {
        [audioPlayer stop];
    }
}
-(void) playAudio
{
    if (!audioRecorder.recording)
    {
        stopButton.enabled = YES;
        recordButton.enabled = NO;

        if (audioPlayer)
            [audioPlayer release];
        NSError *error;

        audioPlayer = [[AVAudioPlayer alloc]
                       initWithContentsOfURL:audioRecorder.url
                       error:&error];

        audioPlayer.delegate = self;

        if (error)
            NSLog(@"Error: %@",
                  [error localizedDescription]);
        else
            [audioPlayer play];
    }
}

-(void)audioPlayerDidFinishPlaying:
(AVAudioPlayer *)player successfully:(BOOL)flag
{
    recordButton.enabled = YES;
    stopButton.enabled = NO;
}
-(void)audioPlayerDecodeErrorDidOccur:
(AVAudioPlayer *)player
                                error:(NSError *)error
{
    NSLog(@"Decode Error occurred");
}
-(void)audioRecorderDidFinishRecording:
(AVAudioRecorder *)recorder
                          successfully:(BOOL)flag
{
}
-(void)audioRecorderEncodeErrorDidOccur:
(AVAudioRecorder *)recorder
                                  error:(NSError *)error
{
    NSLog(@"Encode Error occurred");
}
// Implement viewDidLoad to do additional setup after loading the view,
typically from a nib.
- (void)viewDidLoad
```

```
{
    [super viewDidLoad];
    playButton.enabled = NO;
    stopButton.enabled = NO;

    NSArray *dirPaths;
    NSString *docsDir;

    dirPaths = NSSearchPathForDirectoriesInDomains(
                                                   NSDocumentDirectory,
NSUserDomainMask, YES);
    docsDir = [dirPaths objectAtIndex:0];
    NSString *soundFilePath = [docsDir

stringByAppendingPathComponent:@"sound.caf"];

    NSURL *soundFileURL = [NSURL fileURLWithPath:soundFilePath];

    NSDictionary *recordSettings = [NSDictionary
                                    dictionaryWithObjectsAndKeys:
                                    [NSNumber
numberWithInt:AVAudioQualityMin],
                                    AVEncoderAudioQualityKey,
                                    [NSNumber numberWithInt:16],
                                    AVEncoderBitRateKey,
                                    [NSNumber numberWithInt: 2],
                                    AVNumberOfChannelsKey,
                                    [NSNumber numberWithFloat:44100.0],
                                    AVSampleRateKey,
                                    nil];

    NSError *error = nil;

    audioRecorder = [[AVAudioRecorder alloc]
                     initWithURL:soundFileURL
                     settings:recordSettings
                     error:&error];

    if (error)
    {
        NSLog(@"error: %@", [error localizedDescription]);

    } else {
        [audioRecorder prepareToRecord];
    }
}
```

XXII. MAIL COMPOSER:

MailComposerViewController.h:

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

@interface MailComposerViewController : UIViewController
<MFMailComposeViewControllerDelegate>

-(IBAction)openMail:(id)sender;

@end
```

MailComposerViewController.m:

```
- (IBAction)openMail:(id)sender {
    MFMailComposeViewController *picker = [[MFMailComposeViewController
alloc] init];
     picker.mailComposeDelegate = self;
    [self presentModalViewController:picker animated:YES];
}

// Used to dismiss the mailComposer when cancel is pressed
- (void)mailComposeController:(MFMailComposeViewController*)controller
didFinishWithResult:(MFMailComposeResult)result error:(NSError*)error
{
    [self dismissModalViewControllerAnimated:YES];
}
```

XXIII. Auto-Method Call:

In seconds:

```
[self performSelector:@selector(myMethodName)
withObject:nil afterDelay:2.0f];
```

XXIV. Animation:

```
  [UIView beginAnimations:nil context:nil];
  [UIView setAnimationDuration:0.2];
// here you can implement whatever you want to be animated:
  [UIView commitAnimations];
```

XXV. Data-Persistence:

1) Key/Value way:

a) **SAVING**:

```
-(void)saveString:(NSString*)myString
{
[[NSUserDefaults standardUserDefaults] setObject:myString
forKey:@"String"];

      }
```

Calling the function:

```
[self saveString:@”hello this is a string”];
```

b) **RETRIEVING**:

```
-(NSString*)retrieveString
{
NSString* recoveredString = [[NSUserDefaults standardUserDefaults]
objectForKey:@"String"];
return recoveredString;
      }
```

function call;

```
NSString *myNewString = [self retrieveString];
```

2) DEALING WITH FILES:

```
      // will be available in the next update ☺
```

3) DATABSE:

Database is a big subject to talk about in just few pages, so I’ll be leading you through creating a database.
After that u’ll know how to add items and search for them.

Add sqlite3.h from your framework menu

Add this to your .h:

```
#import <UIKit/UIKit.h>
#import "sqlite3.h"

@interface ViewController : UIViewController {
    NSString *databasePath;
    sqlite3 *contactDB;
}
@property (retain, nonatomic) IBOutlet UITextField *name;
@property (retain, nonatomic) IBOutlet UITextField *address;
@property (retain, nonatomic) IBOutlet UITextField *phone;
@property (retain, nonatomic) IBOutlet UILabel *status;

      @end
```

Add this to your .m:

```
- (void)viewDidLoad
{
    NSString *docsDir;
    NSArray *dirPaths;

    // Get the documents directory
 dirPaths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
NSUserDomainMask, YES);

    docsDir = [dirPaths objectAtIndex:0];

    // Build the path to the database file
databasePath = [[NSString alloc] initWithString: [docsDir
stringByAppendingPathComponent: @"contacts.db"]];

 NSFileManager *filemgr = [NSFileManager defaultManager];

if ([filemgr fileExistsAtPath: databasePath ] == NO)
    { const char *dbpath = [databasePath UTF8String];
 if (sqlite3_open(dbpath, &contactDB) == SQLITE_OK)
        {
           char *errMsg;
          const char *sql_stmt = "CREATE TABLE IF NOT EXISTS CONTACTS (ID
INTEGER PRIMARY KEY AUTOINCREMENT, NAME TEXT, ADDRESS TEXT, PHONE TEXT)";

            if (sqlite3_exec(contactDB, sql_stmt, NULL, NULL, &errMsg) !=
SQLITE_OK)
            {
                status.text = @"Failed to create table";
            }

            sqlite3_close(contactDB);

        } else {
            status.text = @"Failed to open/create database";
        }
    }
    [filemgr release];
    [super viewDidLoad];
     // Do any additional setup after loading the view, typically from a
nib.
}
- (IBAction)saveData:(id)sender {
    [phone resignFirstResponder];
    [address resignFirstResponder];
    [name resignFirstResponder];

    sqlite3_stmt    *statement;

    const char *dbpath = [databasePath UTF8String];

    if (sqlite3_open(dbpath, &contactDB) == SQLITE_OK)
    {
        NSString *insertSQL = [NSString stringWithFormat: @"INSERT INTO
CONTACTS (name, address, phone) VALUES (\"%@\", \"%@\", \"%@\")",
name.text, address.text, phone.text];

        const char *insert_stmt = [insertSQL UTF8String];
      sqlite3_prepare_v2(contactDB, insert_stmt, -1, &statement, NULL);
        if (sqlite3_step(statement) == SQLITE_DONE)
        {
            status.text = @"Contact added";
            name.text = @"";
            address.text = @"";
            phone.text = @"";
        } else {
            status.text = @"Failed to add contact";
        }
        sqlite3_finalize(statement);
        sqlite3_close(contactDB);
    }

      }

- (IBAction)findItem:(id)sender {
    const char *dbpath = [databasePath UTF8String];
    sqlite3_stmt    *statement;

    if (sqlite3_open(dbpath, &contactDB) == SQLITE_OK)
    {
        NSString *querySQL = [NSString stringWithFormat: @"SELECT address,
phone FROM contacts WHERE name=\"%@\"", name.text];

        const char *query_stmt = [querySQL UTF8String];

if (sqlite3_prepare_v2(contactDB, query_stmt, -1, &statement, NULL) ==
SQLITE_OK)
        {
            if (sqlite3_step(statement) == SQLITE_ROW)
            {
NSString *addressField = [[NSString alloc] initWithUTF8String:(const char
*) sqlite3_column_text(statement, 0)];
                address.text = addressField;

NSString *phoneField = [[NSString alloc] initWithUTF8String:(const char *)
sqlite3_column_text(statement, 1)];
                phone.text = phoneField;

                status.text = @"Match found";

                [addressField release];
                [phoneField release];
            } else {
                status.text = @"Match not found";
                address.text = @"";
                phone.text = @"";
            }
            sqlite3_finalize(statement);
        }
        sqlite3_close(contactDB);
    }

      }
```

SUBSTRING:

```
    NSString *string = @"hello my name is midresho";
    NSString *substring=@"my name";
    NSRange textRange = [string rangeOfString:substring];

    if(textRange.location != NSNotFound)
    {
        //Does contain the substring
        NSLog(@"Exists");
    }
```

StoryBoards: modals:

1. link between a view & another using MODAL way.
2. Go to the right side: identifier: hello
3. In your .m add the following line to the method that will be responsible for navigating to the second view
   `[self performSegueWithIdentifier:@"hello" sender:self];`