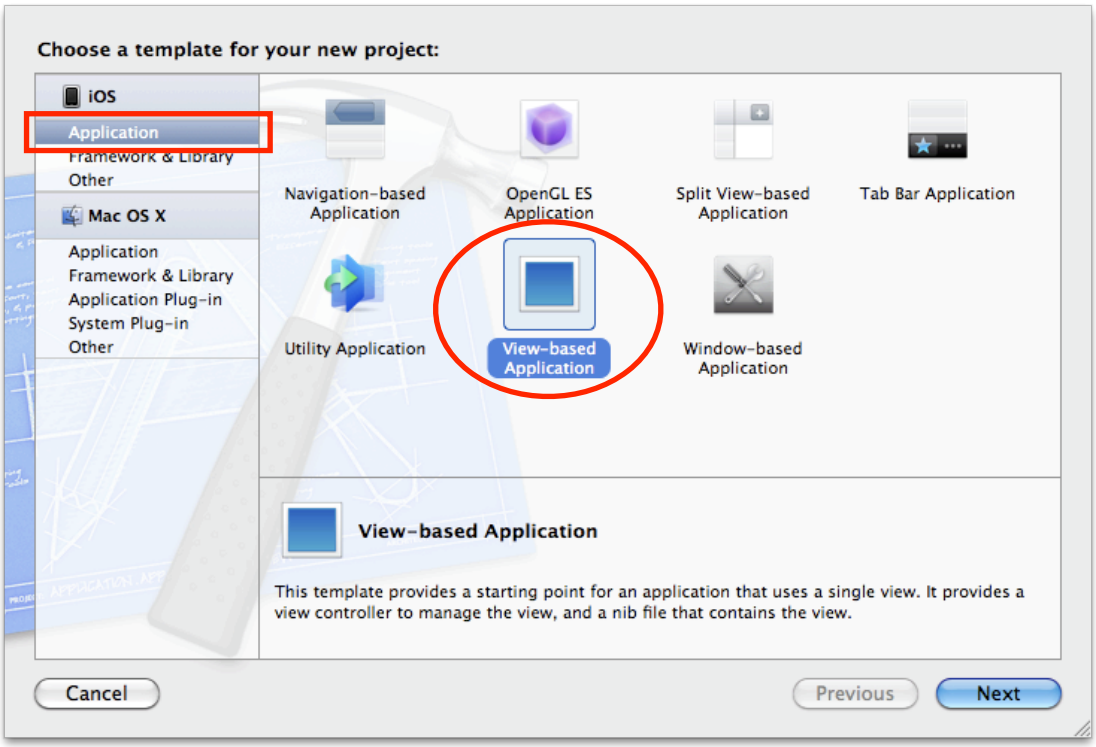
Lab MyEditor

在這個lab裡, 我們將學習到使用 text view 來實作一個簡單的text editor, 並且會學習到如何做 file I/O

Step 1. 在File>New>New Project開啓一個新的專案, 在iOS的Application目錄裡面選擇 view based application

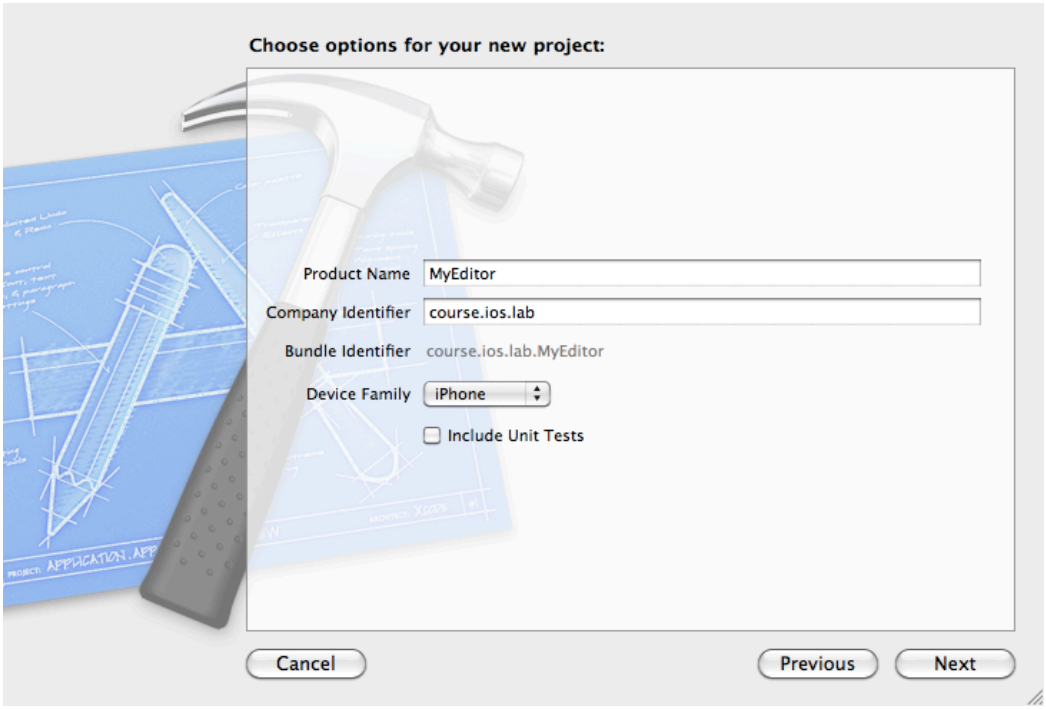


Step 2. 並將此專案命名為 **MyEditor**

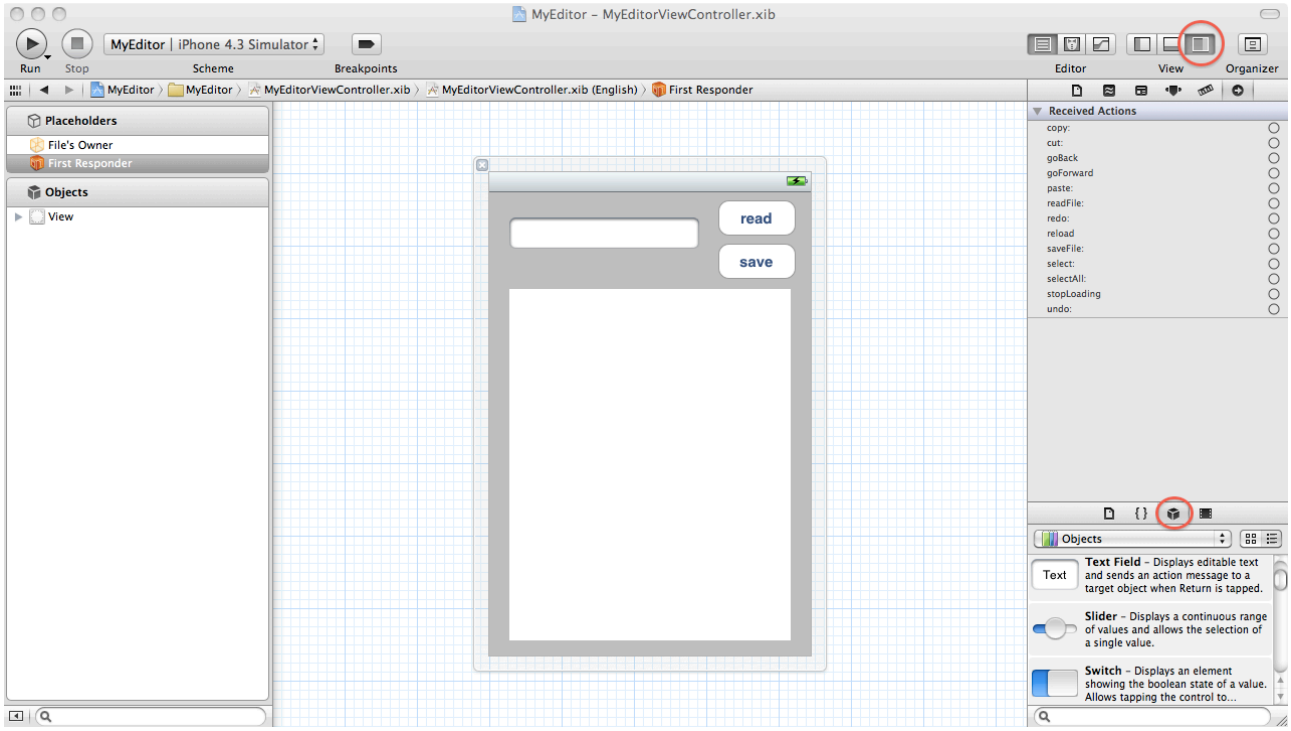
Company Identifier是填入Bundle的名稱,在此統一填入**course.ios.lab** (也可自行填入)

Device Family選擇**iPhone**

Include Unit Tests是做語意邏輯測試用,可勾選也可不勾選,在此我們統一**不勾選**, 存檔

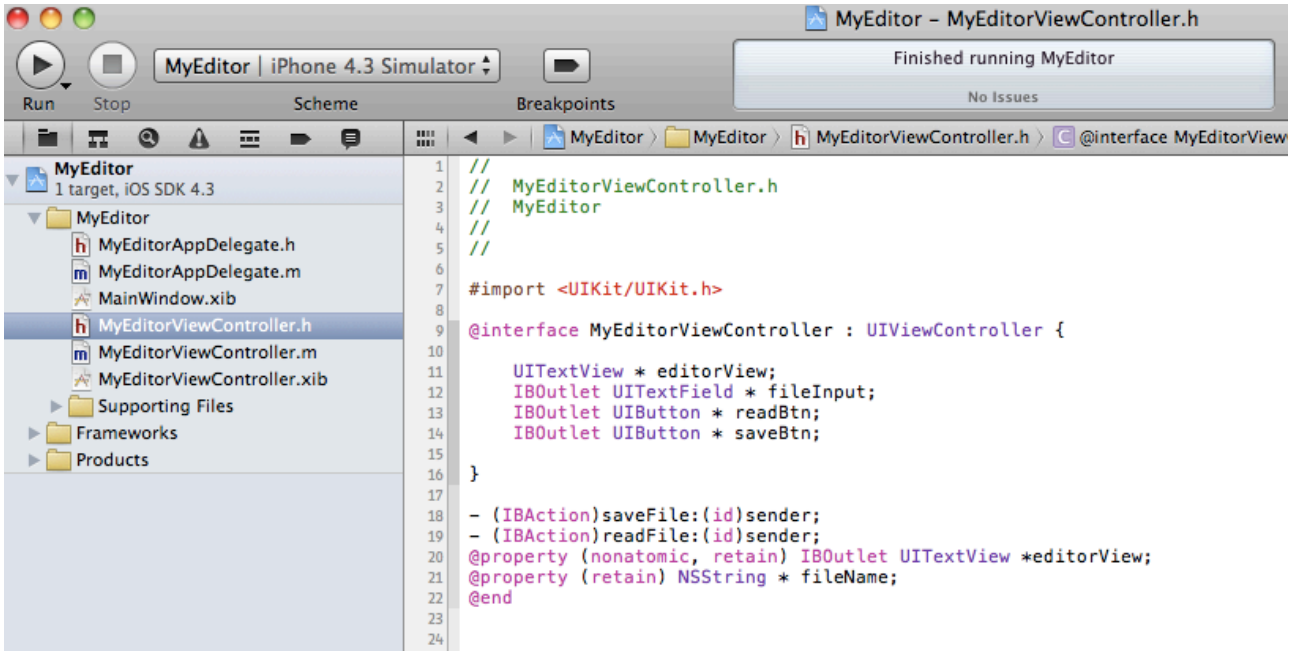


Step 3. Project建立完畢之後, 點選MyEditorViewController.xib , 並且在右上角喚出Show Utility. 接著如下圖 在View 上, 拉選左上角的一個 Text Field, 右上角的兩個 Button (read, save), 下方的 Text View.



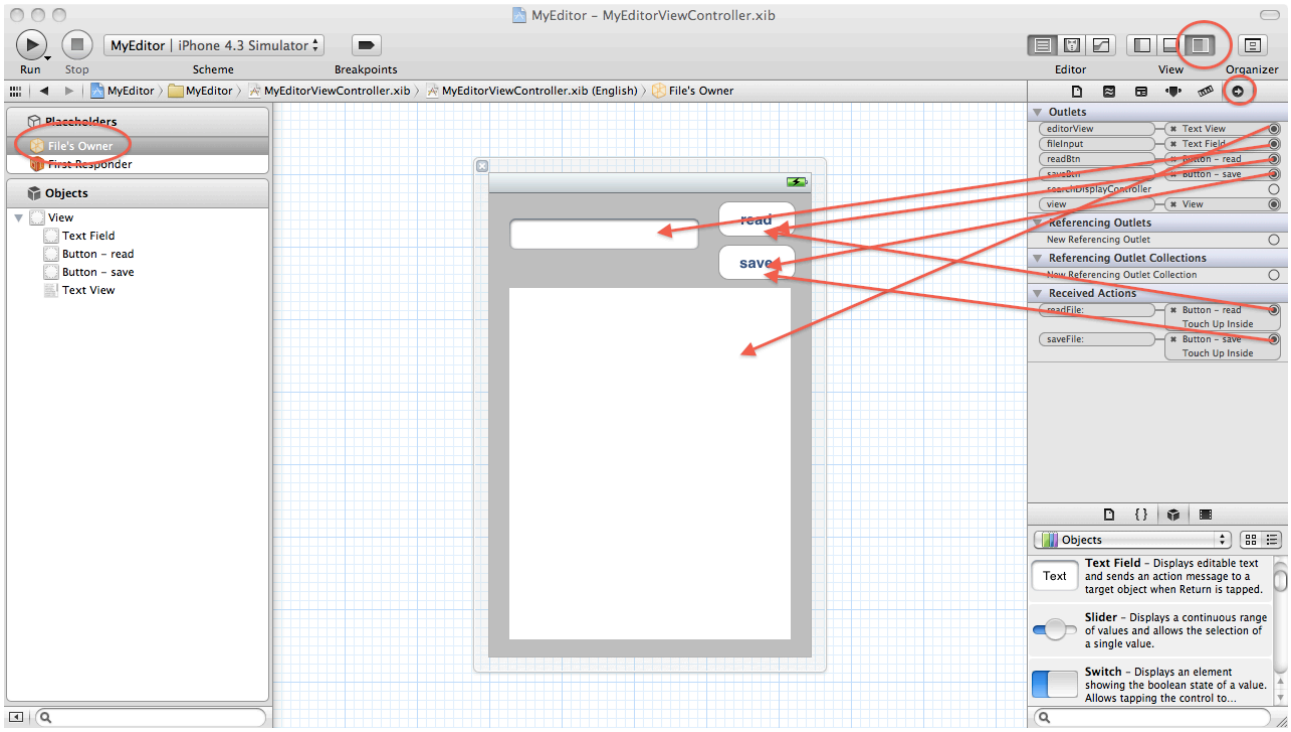
Step 4, 接著點選 MyEditorViewController.h, 定義 剛才在xib上拉選的物件: text field, text view, button. 兩個 button 動作: saveFile, readFile.

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

@interface MyEditorViewController : UIViewController {

    UITextView * editorView;
    IBOutlet UITextField * fileInput;
    IBOutlet UIButton * readBtn;
    IBOutlet UIButton * saveBtn;

}

- (IBAction)saveFile:(id)sender;
- (IBAction)readFile:(id)sender;
@property (nonatomic, retain) IBOutlet UITextView *editorView;
@property (retain) NSString * fileName;
@end
```



Step 5. 回到 MyEditorViewController.xib, 點選FileOwner 的 connections inspector.

將方才在MyEditorViewController.h中建立的物件與View上的物件做連結. 記得將 IBAction 的readFile, saveFile 連結到button的 Touch up inside 事件.



Step 6. 接著回到 MyEditorViewController.m, 實作我們要的動作.

首先先設定.

```
@synthesize editorView;
@synthesize fileName;
#define SCALEDOFFSET 200
```

接著在ViewDidLoad中設定鍵盤Notofication.

```
- (void)viewDidLoad
{
    [super viewDidLoad];

    [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(keyboardAppeared:)
name:UIKeyboardWillShowNotification object:nil];

    [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(keyboardHidden:) name:UIKeyboardWillHideNotification
object:nil];
}
```

然後在ViewDidUnload, 別忘了在View被 unload的時候, 養成好習慣, 要去釋放物件.

```
- (void)viewDidUnload
{
    [super viewDidUnload];

    [self setEditorView:nil];
    [fileInput release];
    fileInput = nil;
    [readBtn release];
    [saveBtn release];
}
```

接著實作鍵盤出現以及消失的時候的行為.

```
-(void) keyboardAppeared:(NSNotification *)noti{
    CGRect editorFrame = editorView.frame ;
    editorFrame.size = CGSizeMake(editorFrame.size.width,
editorFrame.size.height - SCALEDOFFSET);
    [UIView animateWithDuration:0.5 animations:^(void) {
        editorView.frame = editorFrame;
    }];
}
-(void) keyboardHidden:(NSNotification *)noti{
    CGRect editorFrame = editorView.frame ;
    editorFrame.size = CGSizeMake(editorFrame.size.width,
editorFrame.size.height +SCALEDOFFSET);
    [UIView animateWithDuration:0.5 animations:^(void) {
        editorView.frame = editorFrame;
    }];
}
```

接著實作 saveFile 這個IBAction, 首先先做個防呆機智, 判斷是否有輸入檔名, 若無則設定為預設值”lessonTest.txt”, 接著注意, 在 設定 filePath 時, 必須設定本機的絕對路徑. 一個簡單的方法就是, 打開Finder, 找到自己的使用者名稱(在此例子為 Berby, 那麼絕對位置就是 /Users/Berby/, 請依照實作時的名稱來寫)

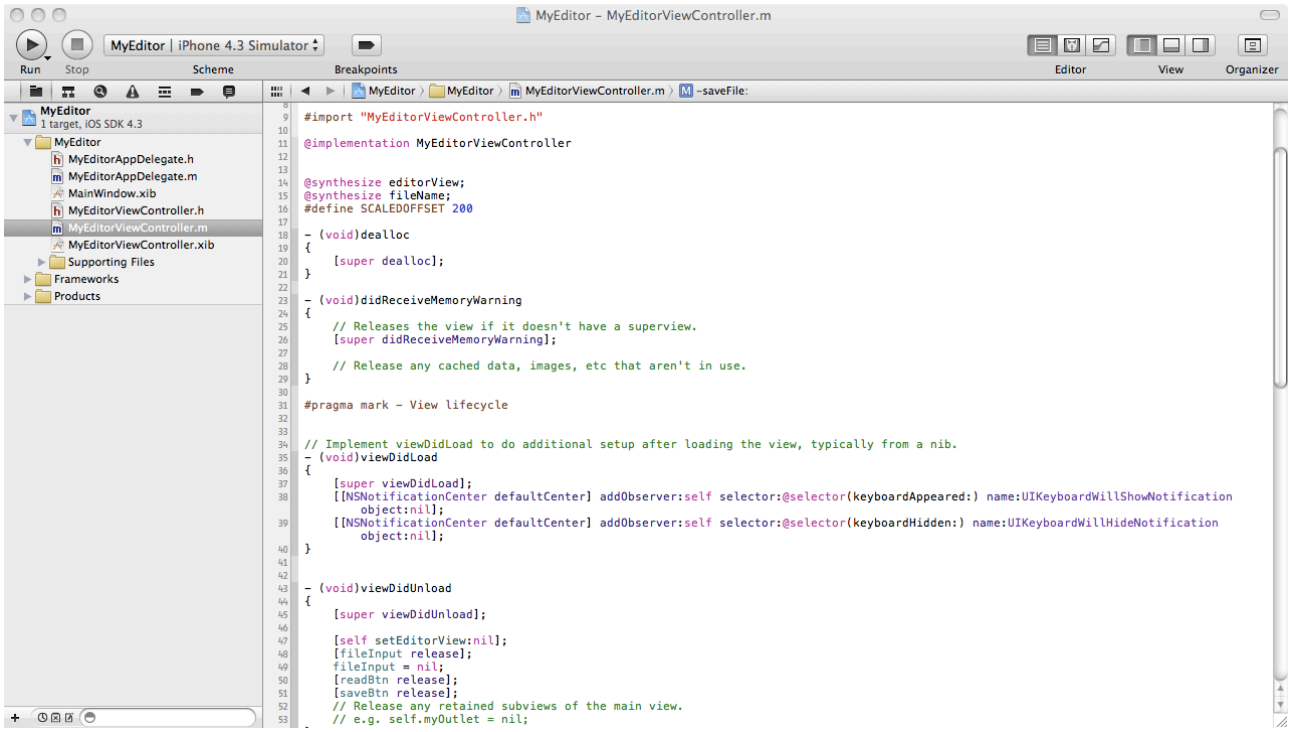
然後利用 NSFileManager 來將文字寫出.

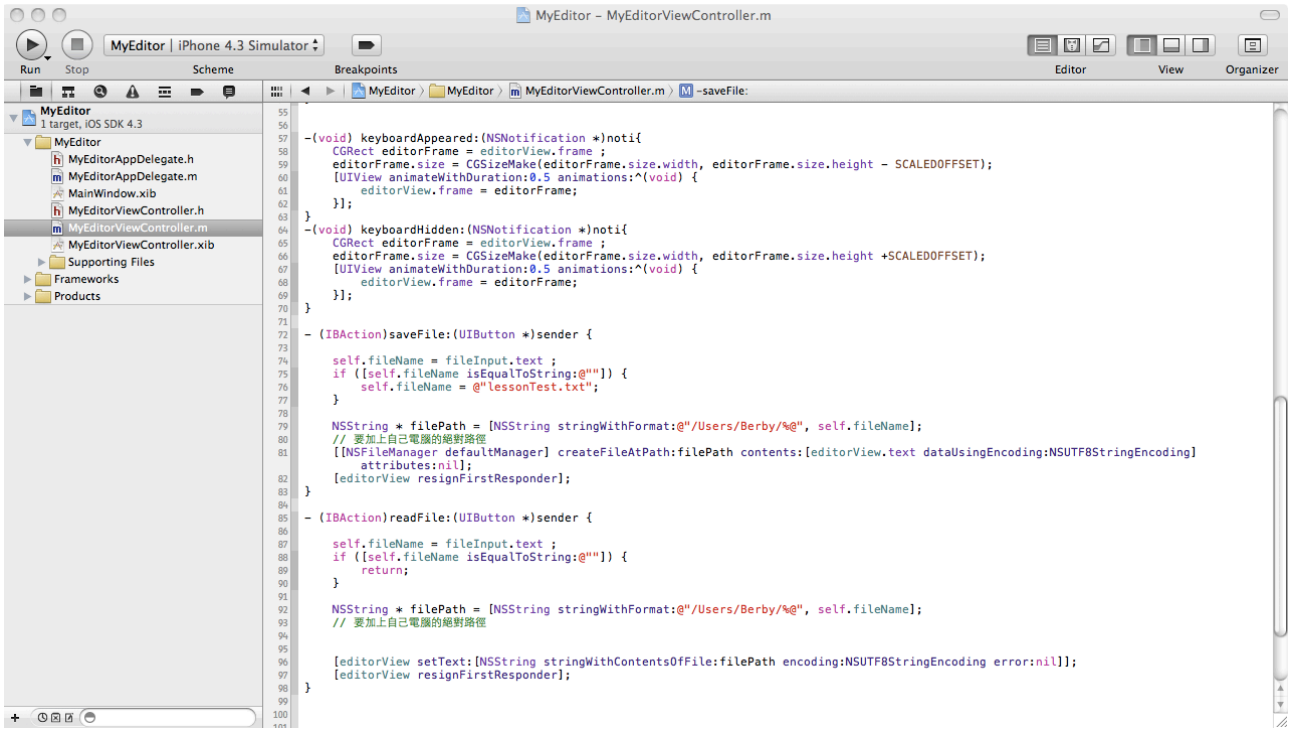
```
- (IBAction)saveFile:(UIButton *)sender {

    self.fileName = fileInput.text ;
    if ([self.fileName isEqualToString:@""]) {
        self.fileName = @"lessonTest.txt";
    }

    NSString * filePath = [NSString stringWithFormat:@"/Users/Berby/%@",
self.fileName];
    // 要加上自己電腦的絕對路徑
    [[NSFileManager defaultManager] createFileAtPath:filePath contents:
[editorView.text dataUsingEncoding:NSUTF8StringEncoding]
attributes:nil];
    [editorView resignFirstResponder];
}
```

接著實作 readFile, 同樣的, 路徑也必須設定為絕對名稱. 在這裡, 我們將讀取到的文字填回 editorView.

```
- (IBAction)readFile:(UIButton *)sender {

    self.fileName = fileInput.text ;
    if ([self.fileName isEqualToString:@""]) {
        return;
    }

    NSString * filePath = [NSString stringWithFormat:@"/Users/Berby/%@",
self.fileName];
    // 要加上自己電腦的絕對路徑


    [editorView setText:[NSString stringWithContentsOfFile:filePath
encoding:NSUTF8StringEncoding error:nil]];
    [editorView resignFirstResponder];
}
```

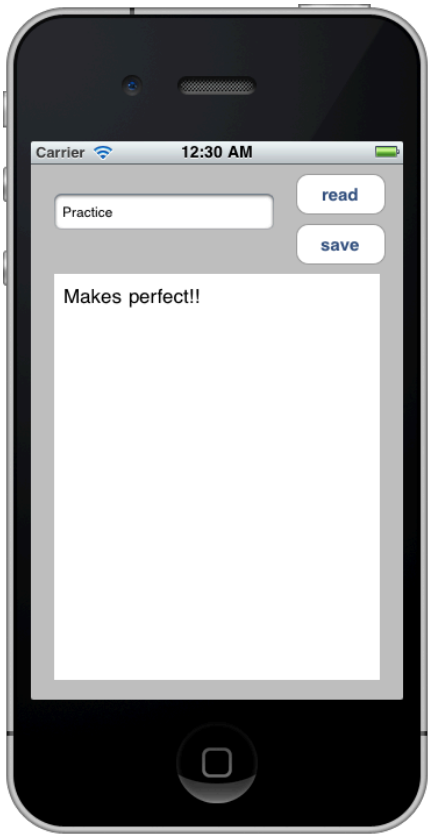
你可以如下圖去實作這些code.





Step 7. Run (⌘+R)

可以試著打入一些text, 然後save. 接著關掉程式重跑, 可以試著用指定的檔名將text再read回來.



輸出的檔案可以利用方才設定的絕對路徑在Finder中找到. 我們可以把他點開來看看.