

iPhone道場

体験コース

「ストップウォッチを作ろう」

今回のテーマは...

- 前回のラベルとボタンにタイマー機能を加えてストップウォッチを作ってみよう。
- まずはシンプルに...



OOViewController.h

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

@interface OOViewController : UIViewController {
    IBOutlet UILabel *label1;
    NSTimer *timer;
    NSDate *stdate;
    BOOL timeflg;
}

- (IBAction) start:(id) sender;
- (IBAction) stop:(id) sender;
- (IBAction) clear:(id) sender;

@end
```

OOViewController.m

```
#import "OOViewController.h"

@implementation OOViewController

- (void)onTimer:(NSTimer*)timer {
    if(timeflg) {
        NSDate *now = [NSDate date];
        label1.text = [NSString stringWithFormat:@"%%.2f",
            [now timeIntervalSinceDate:stdat]];
    }
}
```

○○ViewController.m ...続き

```
- (IBAction) start:(id) sender {
    timeflg = TRUE;
    stdate = [NSDate date];
    [stdate retain];
}

- (IBAction) stop:(id) sender {
    timeflg = FALSE;
}

- (IBAction) clear:(id) sender {
    timeflg = FALSE;
    label1.text = @"0.00";
}
```

○○ViewController.m ...さらに続き

```
/*  
// Implement viewDidLoad to do additional setup after loading the view, typically from  
a nib.  
- (void)viewDidLoad {  
    [super viewDidLoad];  
}  
*/
```

↓ コメントアウトを外して編集 ↓

```
// Implement viewDidLoad to do additional setup after loading the  
view, typically from a nib.  
- (void)viewDidLoad {  
    [super viewDidLoad];  
    timer = [NSTimer scheduledTimerWithTimeInterval:(0.01)  
target:self  
selector:@selector(onTimer:)  
userInfo:nil  
repeats:YES];  
}
```

実行してみる

先ほどのソースでは、
現在の時間から
STOPを押されるまでの
時間を表示しているため、
STOPした時間からの
続きが計れない。
(毎回リセットされる)



アップグレードしよう

〇〇ViewController.h に追記

```
@interface 〇〇ViewController : UIViewController {
    IBOutlet UILabel *label1;
    NSTimer *timer;
    NSDate *stdate;
    BOOL timeflg;
    double stoptime;
    double oldtime;
}
```


アップグレードしよう

○○ViewController.m を編集

```
- (void)onTimer:(NSTimer*)timer {
    if(timeflg) {
        NSDate *now = [NSDate date];
        stoptime = [now timeIntervalSinceDate:stdatetime];
        label1.text = [NSString stringWithFormat:@"%%.2f",
            stoptime+oldtime];
    }
}

-(IBAction) start:(id) sender {
    //変更なし
}
```

アップグレードしよう

○○ViewController.m を編集 (続き)

```
-(IBAction) stop:(id) sender {
    timeflg = FALSE;
    oldtime = [[label1 text] doubleValue];
}

-(IBAction) clear:(id) sender {
    timeflg = FALSE;
    stoptime = 0.0;
    oldtime = 0.0;
    label1.text = [NSString stringWithFormat:@"%f", stoptime];
}
```

アイコンとアプリ名の設定

The screenshot shows the Xcode interface with the 'doujou2-Info.plist' file open. The 'Icon file' is set to 'tokei.png' and the 'Bundle display name' is set to '道場2'. The 'Executable file' is set to '\$(EXECUTABLE_NAME)'. The 'Bundle identifier' is 'com.yourcompany.\${PRODUCT_NAME:rfc1034identifier}'. The 'InfoDictionary version' is '6.0'. The 'Bundle name' is '\$(PRODUCT_NAME)'. The 'Bundle OS Type code' is 'APPL'. The 'Bundle creator OS Type code' is '????'. The 'Bundle version' is '1.0'. The 'Application requires iPhone environment' checkbox is checked. The 'Main nib file base name' is 'MainWindow'.

Key	Value
Information Property List	(12 items)
Localization native development region	English
Bundle display name	道場2
Executable file	\$(EXECUTABLE_NAME)
Icon file	tokei.png
Bundle identifier	com.yourcompany.\${PRODUCT_NAME:rfc1034identifier}
InfoDictionary version	6.0
Bundle name	\$(PRODUCT_NAME)
Bundle OS Type code	APPL
Bundle creator OS Type code	????
Bundle version	1.0
Application requires iPhone environment	<input checked="" type="checkbox"/>
Main nib file base name	MainWindow

Resourcesフォルダ

- 画像ファイルや音楽ファイルなどの外部ファイルを収納するフォルダ。
- Resourcesフォルダに画像ファイルをドロップするとアイコンやボタンの背景などに使用することができる。

画像を貼り付けてみよう

