

## View

### UIKit Framework

- cocoatouch framework에 추가된 UI관련 기능의 클래스가 모여있는 프레임워크
- #import <UIKit/UIKit.h>로 추가한다. (무조건 import해야한다.)

### UIView

- 가장 기본이 되는 View
- UIComponent들의 조합으로 화면이 구성되며 UIView를 상속 받았다. 즉 IOS 화면구성은 UIView의 집합으로 되어있다.

```
//UI를 이용해 뷰만들어보기
//view
UIView *newView = [[UIView alloc] initWithFrame:CGRectMake(15, 15, self.view.frame.size.width -30,
                                                         self.view.frame.size.height /2 + 100)];

newView.backgroundColor = [UIColor whiteColor];
newView.alpha = 1;
[self.view addSubview:newView];
```

### UILabel

- 텍스트 정보를 표시하는 UI
- 주요항목
  - NSString \*text; 텍스트
  - UIFont \*font; 폰트설정, 크기
  - UIColor \*textColor; 텍스트 색상
  - NSTextAlignment textAlignment; 텍스트 정렬
  - NSLineBreakMode linebreakmode; 줄나누기, 0이 무제한

```
//라벨
UILabel *label1 = [[UILabel alloc] initWithFrame:CGRectMake(0,box1.frame.size.height - 30,
                                                           box1.frame.size.width, 30)];

label1.text = @"1000클릭";
label1.textColor = [UIColor redColor];
label1.textAlignment = NSTextAlignmentCenter;
[box1 addSubview:label1];
```

### UIImageView

- 이미지 정보를 표시하는 UI
- 주요항목
  - UIImage \*image; 이미지
  - UIViewContentMode contentMode; 콘텐츠모드에 따라서 보여진다.

```
//이미지뷰
UIImageView *image1 = [[UIImageView alloc] initWithFrame:CGRectMake(0, 0, box1.frame.size.width,
                                                                      box1.frame.size.height - 30)];

image1.image = [UIImage imageNamed:@"람보르기니.jpg"];
[image1 setContentMode:UIViewContentModeScaleToFill];
[box1 addSubview:image1];
```

### UIControl

- UICoponent에 사용자 인터렉션에 의한 응답에 대해 특별한 액션을 줄수 있게 설정하는 클래스
- BOOL enabled; 사용할것인가?
- BOOL selected; 눌렀을경우
- BOOL highlighted; 누르고 있을경우

### UIButton

- 사용자의 이벤트를 받아 처리해주는 UI
- 버튼구조 : title, image, backgroundImage

참조 1

```
//UIButton
UIButton *btn = [UIButton buttonWithType:UIButtonTypeCustom];
[btn addTarget:self action:@selector(onTouchUpInsideCoin:) forControlEvents:UIControlEventTouchUpInside];
btn.autoresizingMask = UIViewAutoresizingFlexibleLeftMargin| UIViewAutoresizingFlexibleHeight;
[btn setTitle:[self.costData objectAtIndex:i] forState:UIControlStateNormal];
[btn setTitleColor:[UIColor blueColor] forState:UIControlStateNormal];
[btn setTitleColor:[UIColor grayColor] forState:UIControlStateHighlighted];
btn.tag = i;
[inputView addSubview:btn];
[self.inputBtns addObject:btn];
```

참조 2

```
//회원가입 버튼
UIButton *pwbt = [UIButton buttonWithType:UIButtonTypeCustom];
pwbt.frame = CGRectMake(self.mainScroll.frame.size.width - 150, (self.mainScroll.frame.size.height /2) + 100, 100, 30);
[pwbt setTitle:@"회원가입" forState:UIControlStateNormal];
[pwbt setTitle:@"회원가입" forState:UIControlStateSelected];
[pwbt setTitleColor:[UIColor whiteColor] forState:UIControlStateNormal];
[pwbt setTitleColor:[UIColor darkGrayColor] forState:UIControlStateSelected];
pwbt.backgroundColor = [UIColor orangeColor];
pwbt.titleLabel.textAlignment = NSTextAlignmentCenter;
[pwbt addTarget:self action:@selector(touchin:) forControlEvents:UIControlEventTouchUpInside];
[self.mainScroll addSubview:pwbt];
```

참조 3

```
//아이디, 회원가입버튼 눌렀을 경우, 버튼액션
- (void)touchin:(UIButton *)button
{
    if (button.selected) {
        button.selected = NO;
    }else {
        button.selected = YES;
    }
}
```

### Delegate

- protocol
  - 복수의 컴퓨터 사이나 중앙 컴퓨터와 단말기 사이에서 데이터 통신을 원활하게 하기 위해 필요한 통신 규약, 신호 송신의 순서, 데이터의 표현법, 오류검출법 등을 정함

- 객체지향에선 두 클래스 간의 약속
  - 이런 이름의 함수를 만들어라!!

- 텍스트필드 delegate 프로토콜 연결(다른것과 연결시 , 후 연달아 적으면됨)

```
#import "ViewController.h"
```

```
@interface ViewController () <UITextFieldDelegate>
```

### UITextField

- 사용자 텍스트 입력을 위한 UI Component

- 주요항목
    - NSString \*text; 텍스트 작성
    - UIFont \*font; 폰트지정, 크기
    - UIColor \*textColor; 텍스트 색상
    - NSTextAlignment textAlignment; 텍스트 정렬
    - UITextBorderStyle borderstyle; 테두리스타일
    - NSString \*placeholder; 테두리안의 글자
    - UIImage \*background; 테두리배경
- ```
//password 텍스트필드
self.pwText = [[UITextField alloc] initWithFrame:CGRectMake(50, (self.mainScroll.frame.size.height / 2) + 50,
                                                            self.mainScroll.frame.size.width - 100, 40)];

self.pwText.font = [UIFont boldSystemFontOfSize:20];
self.pwText.textColor = [UIColor blackColor];
self.pwText.layer.borderWidth = 2.0f;
self.pwText.layer.borderColor = [[UIColor orangeColor]CGColor];
self.pwText.textAlignment = NSTextAlignmentCenter;
self.pwText.borderStyle = UITextBorderStyleLine;
self.pwText.secureTextEntry = YES;
self.pwText.placeholder = @"pw를 입력해주세요.";
self.pwText.delegate = self;
[self.mainScroll addSubview:self.pwText];
```

### UIScrollView

- ViewSize보다 확장된 뷰를 보기위한 View
- UIScrollView에 추가된 View는 ContentView위에 추가된다.

```
//스크롤뷰 생성
UIScrollView *scroll = [[UIScrollView alloc] initWithFrame:CGRectMake(0, 0, 320, 460)];
[scroll setContentSize:CGSizeMake(scroll.frame.size.width * 3, 460)]; //컨텐츠뷰 설정
scroll.delegate = self; //델리게이트 연결
scroll.pagingEnabled = YES; //스크롤 사용
[self.view addSubview:scroll];
//페이지 1
UIView *newView1 = [[UIView alloc] initWithFrame:CGRectMake(0, 0, 320, 460)];
newView1.backgroundColor = [UIColor purpleColor];
[scroll addSubview:newView1];
//페이지 2
UIView *newView2 = [[UIView alloc] initWithFrame:CGRectMake(320, 0, 320, 460)];
newView2.backgroundColor = [UIColor redColor];
[scroll addSubview:newView2];
//페이지 3
UIView *newView3 = [[UIView alloc] initWithFrame:CGRectMake(640, 0, 320, 460)];
newView3.backgroundColor = [UIColor greenColor];
[scroll addSubview:newView3];
```

### UITableView

- style
  - plain
    - 기본적인 테이블 뷰
    - 여러개의 세션을 가질수 있다.
    - 한 세션에는 여러개의 로우(열)을 포함하고있다.
    - 각각의 세션에는 세션을 표시하는 헤더, 풋터 title을 사용할수있다.
    - 세션을 빠르게 검색할수 있는 네비게이터바를 index list라고 부른다.
  - grouped
    - 각 세션을 그룹의 형태로 나타내는 테이블뷰
    - 데이터의 디테일한 정보를 표현할때 많이 사용된다.
- protocol
  - DataSource : userinterface를 다루는데 사용
    - 프로토콜을 사용하여 테이블뷰에서 보여줄 데이터를 관리할 대리인의 역할을 정의해둔것
    - 역할
      - @requires : 테이블뷰의 각 세션별 열의 개수를 설정, row(열)별 cell객체
      - @optional : 테이블뷰 세션의 개수를 설정
  - Delegate : data를 불러오는데 사용
    - 프로토콜을 사용하여 테이블뷰의 대리자로서 수행할 수 있는 역할들을 정의해 둔것
    - 역할
      - 헤더 또는 풋터의 높이를 설정 : Variable height support
      - 헤더 또는 풋터 뷰를 제공 : Section informations
      - 셀을 선택하였을때 수행할 동작 관리 : Selection
      - 셀의 삭제 될때의 동작 등의 관리 : Editing
- NSIndexPath Class
  - 셀의 위치를 나타낼 data type으로 seccion정보와 해당 섹션에서의 열 정보를 가지고 있다.
- UITableViewCell
  - contentView
    - .textLabel, detailTextLabel, imageView
  - accessoryView
- 재사용(reuse)
  - 테이블 뷰는 한줄 한줄이 굉장히 비슷한 모습을 가지고 있다.
  - 수백개의 테이블 리스트가 있다고 가정할때....
  - 공통적인 레이아웃을 사용하여 여러번 화면을 보여줄 필요가 있는 경우 뷰를 재사용
  - 재사용하기위해서는 셀들의 형식이 비슷해야함