

Creating iOS UI Programmatically

Using Objective-C

Default Initialization Method

- The default initialization method for views is the [initWithFrame:](#) method, which sets the initial size and position of the view relative to its (soon-to-be-established) parent view. For example, to create a new generic UIView object, you could use code similar to the following:
 - `CGRect viewRect = CGRectMake(0, 0, 100, 100);`
 - `UIView* myView = [[UIView alloc] initWithFrame:viewRect];`
- After you create a view, you must add it to a window (or to another view in a window) before it can become visible.

Adding and Removing Subviews

- To add a subview to a parent, call the [addSubview:](#) method of the parent view. This method adds the subview to the end of the parent's list of subviews.
- To insert a subview in the middle of the parent's list of subviews, call any of the `insertSubview:...` methods of the parent view. Inserting a subview in the middle of the list visually places that view behind any views that come later in the list.
- To reorder existing subviews inside their parent, call the [bringSubviewToFront:](#), [sendSubviewToBack:](#), or [exchangeSubviewAtIndex:withSubviewAtIndex:](#) methods of the parent view. Using these methods is faster than removing the subviews and reinserting them.
- To remove a subview from its parent, call the [removeFromSuperview](#) method of the subview (not the parent view).

ViewController – Declaring UI Objects

- First we need to create class properties for the text field and the label. Creating a property allows instances of an object to be accessed anywhere in the class, as opposed to just within the scope of one function.
- Private properties can be declared between the lines below in the MainViewController.m file.
- @interface MainViewController()
@end
- we can add in the lines:
- @interface MainViewController()
 - @property (nonatomic, strong) UITextField *textField;
 - @property (nonatomic, strong) UILabel *label;
- @end

ViewController – Creating and Adding UI Objects

- One place where you might add subviews to a view hierarchy is in the [loadView](#) or [viewDidLoad](#) methods of a view controller. If you are building your views programmatically, you put your view creation code in the `loadView` method of your view controller.

CGRect and CGRectMake

- The data structure CGRect represents the location and dimensions of a rectangle
- CGRectMake returns a rectangle with the specified coordinate and size values
- [CGRect](#) CGRectMake ([CGFloat](#) x, [CGFloat](#) y, [CGFloat](#) width, [CGFloat](#) height);
- A point's x- and y-coordinates or a size's width and height are unitless quantities—whether such measurements are treated as pixels, scale-factor-independent points, texture elements (texels), or some other unit depends on the API using the measurement, and on the context in which that API is used

ViewController – Adding a TextField

- - (void)viewDidLoad

```
{
```

```
[super viewDidLoad];
```

```
// Do any additional setup after loading the view.
```

```
/* this creates the text field and sets the frame “CGRectMake” creates the frame with (x,y,width,height) where x and y are pixel distances from the top left of the screen */
```

```
self.textField = [[UITextField alloc] initWithFrame: CGRectMake(10.0f, 30.0f, 300.0f, 30.0f)];
```

```
//changes the border style so the text field appears on screen
```

```
self.textField.borderStyle = UITextBorderStyleRoundedRect;
```

```
/* add the text field to the main view. Note: UITextField (along with a number of other UI classes are subclasses of UIView, which means they can be added onto the view hierarchy */
```

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

```
}
```

ViewController – Adding a Button

```
//initialize the button with the default, rounded rect type
UIButton *button =
    [UIButton buttonWithType:UIButtonTypeRoundedRect];
//set the button's frame
self.button.frame = CGRectMake(110.0f, 200.0f, 100.0f, 30.0f);
//set the target, action, and control event.
[self.button addTarget:self action:@selector(buttonPressed)
    forControlEvents:UIControlEventTouchUpInside];
//set the title
[self.button setTitle:@"Press Here!" forState:UIControlStateNormal];
//add the button to the main view
[self.view addSubview:self.button];
```


ViewController – Adding a Label

```
//initialize Label Size and position
    CGRect rect1 = CGRectMake(110.0f, 200.0f, 100.0f, 30.0f);
//Create the label object now
    self.label = [[UILabel alloc] initWithFrame: rect1]
//set the title
    self.label.text = @"Hello Sisoft!";
//add the Label to the main view
[self.view addSubview:label];
```