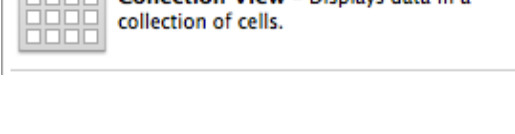


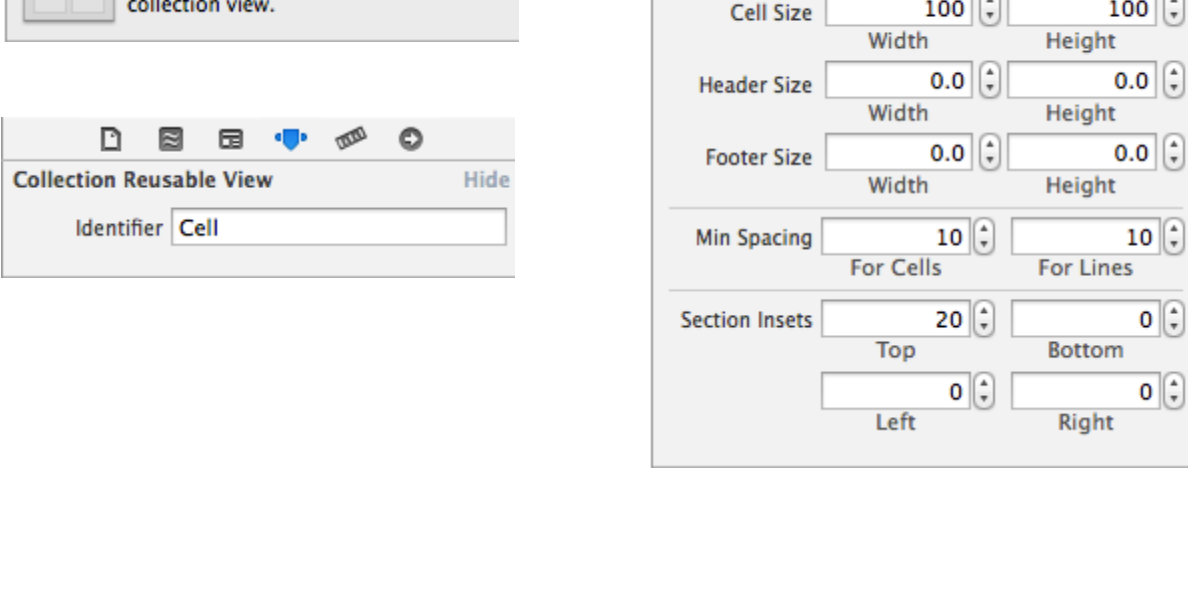
# UI Collection View

## Storyboarding

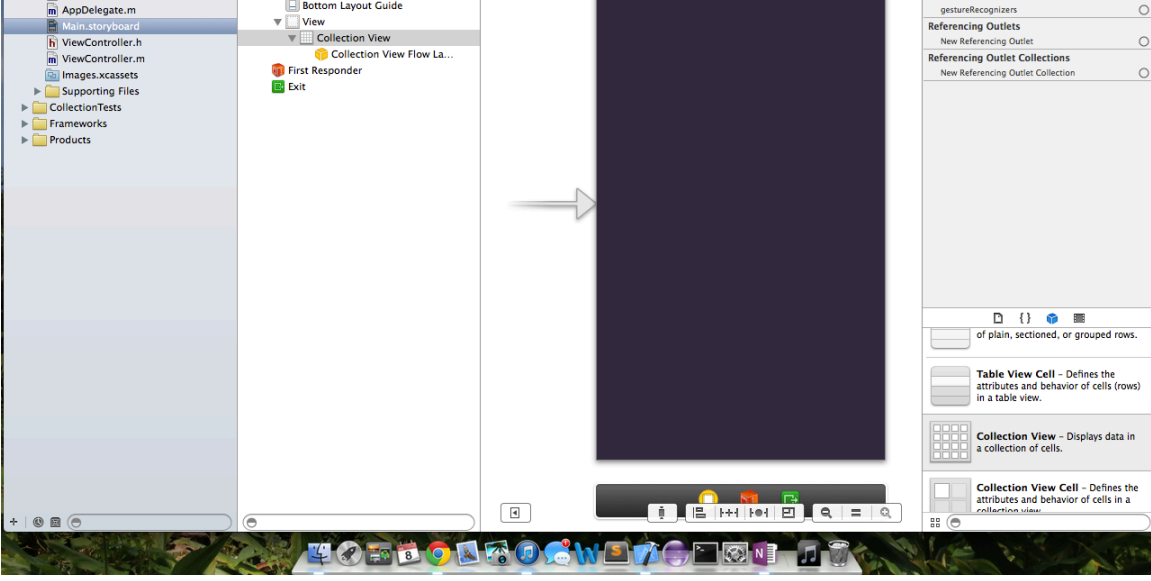
In the right column where the objects are listed, select Collection View and drag it onto the view.



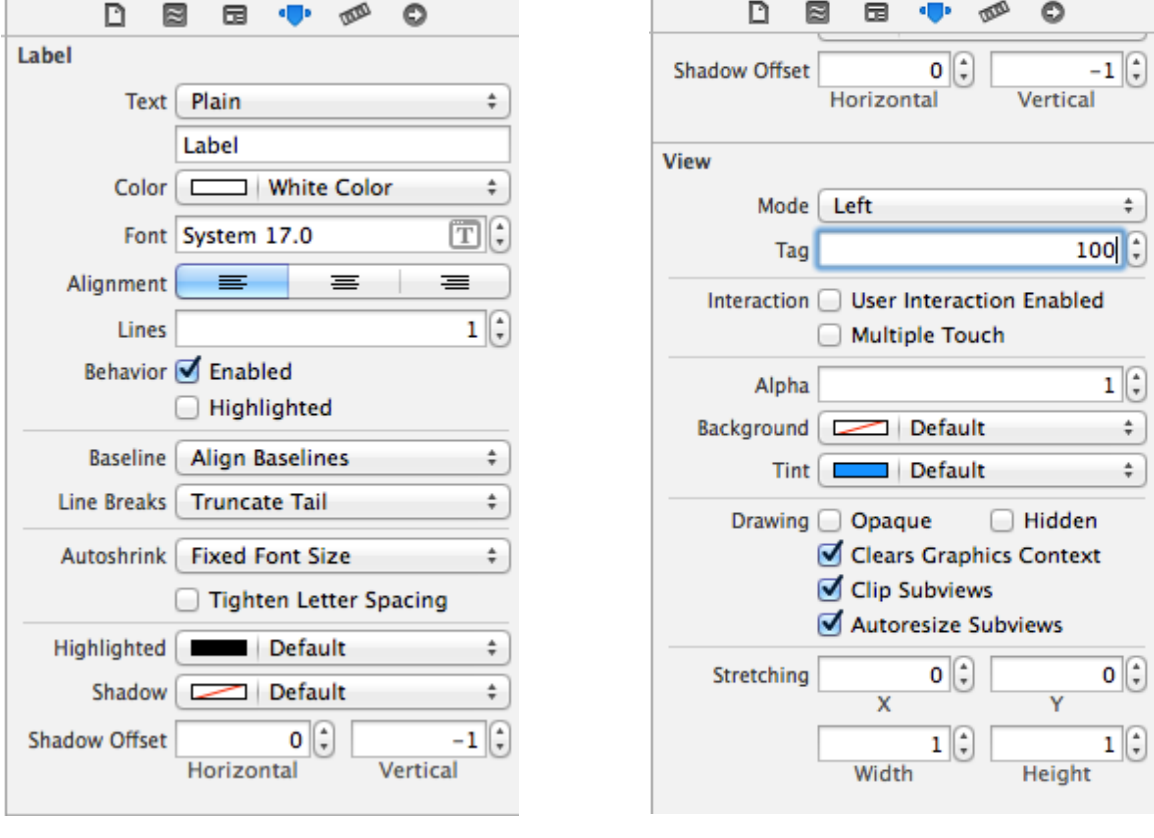
Insert a Collection View Cell into the Collection View; resize and arrange under the Size Inspector and add an Identifier named Cell under the Attributes Inspector



Be sure to connect the dataSource and delegate from the Collection View to the View Controller as shown in the screenshot below



Next, Add a label to the center of the cell. Change the color of the text to white and the Tag value to 100.



## Programming

In your ViewController.h file add <UICollectionViewDataSource, UICollectionViewDelegate> to the end of @interface ViewController :

UICollectionView so the program can implement the methods called with UICollectionView. In the ViewController.m file, add the code below:

```
@implementation ViewController {
    NSMutableArray * array;
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    array = [[NSMutableArray alloc] init];

    [array addObject:@"hello"];
    [array addObject:@"this"];
    [array addObject:@"is"];
    [array addObject:@"a"];
    [array addObject:@"test"];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

//number of sections you have in your collection
- (NSInteger) numberOfSectionsInCollectionView:(UICollectionView *)collectionView{
    return 1;
}

//returns what content is in those sections
- (NSInteger) collectionView:(UICollectionView *)collectionView numberOfItemsInSection:(NSInteger)section{
    return [array count];
}

//customized the cells to be circles with white borders and a black background
- (UICollectionViewCell *) collectionView:(UICollectionView *)collectionView cellForItemAtIndexPath:(NSIndexPath *)indexPath{

    UICollectionViewCell *cell = [collectionView dequeueReusableCellWithReuseIdentifier:@"Cell" forIndexPath:indexPath];
    UILabel *label = (UILabel *)[cell viewWithTag:100];
    label.text = [array objectAtIndex:indexPath.row];

    [cell.layer setBorderWidth:2.0f];
    [cell.layer setBorderColor:[UIColor whiteColor].CGColor];
    [cell.layer setCornerRadius:50.0f];

    return cell;
}
```

## Programmatically implement UICollectionView:

### ViewController.h file

```
@interface ViewController : UIViewController
<UICollectionViewDataSource, UICollectionViewDelegate>

    UICollectionView *_collectionView;
}
```

### ViewController.m file

```
- (void)viewDidLoad
{
    self.view = [[UIView alloc] initWithFrame:[UIScreen mainScreen] bounds];

    //sets up collection view layout and calls the dataSource and delegate
    UICollectionViewFlowLayout *layout=[[UICollectionViewFlowLayout alloc] init];
    _collectionView=[[UICollectionView alloc] initWithFrame:self.view.frame collectionViewLayout:layout];
    [_collectionView setDataSource:self];
    [_collectionView setDelegate:self];

    [_collectionView registerClass:[UICollectionViewCell class] forCellWithReuseIdentifier:@"CellIdentifier"];
    [_collectionView setBackgroundColor:[UIColor whiteColor]];

    [self.view addSubview:_collectionView];

    [super viewDidLoad];
}

//number of cells in the collection view
- (NSInteger)collectionView:(UICollectionView *)collectionView numberOfItemsInSection:(NSInteger)section
{
    return 12;
}

//this method helps with the creation and arrangement of the cells in your collection; can use to change color, add pictures or add text
(with a label already in place)
- (UICollectionViewCell *)collectionView:(UICollectionView *)collectionView cellForItemAtIndexPath:(NSIndexPath *)indexPath
{
    UICollectionViewCell *cell=[collectionView dequeueReusableCellWithReuseIdentifier:@"CellIdentifier" forIndexPath:indexPath];

    cell.backgroundColor=[UIColor blueColor];
    return cell;
}

// this method determines the size/layout of your cells
- (CGSize)collectionView:(UICollectionView *)collectionView layout:(UICollectionViewLayout*)collectionViewLayout
sizeForItemAtIndexPath:(NSIndexPath *)indexPath
{
    return CGSizeMake(100, 100);
}
```