UICollectionView Class Reference

|  |  |
| --- | --- |
| **Inherits from** | [UIScrollView](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIScrollView_Class/Reference/UIScrollView.html#//apple_ref/occ/cl/UIScrollView) : [UIView](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIView_Class/UIView/UIView.html#//apple_ref/occ/cl/UIView) : [UIResponder](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIResponder_Class/Reference/Reference.html#//apple_ref/occ/cl/UIResponder) : [NSObject](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSObject_Class/Reference/Reference.html#//apple_ref/occ/cl/NSObject) |
| **Conforms to** | [NSCoding (UIScrollView)](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Protocols/NSCoding_Protocol/Reference/Reference.html#//apple_ref/occ/intf/NSCoding) [NSCoding (UIView)](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Protocols/NSCoding_Protocol/Reference/Reference.html#//apple_ref/occ/intf/NSCoding) [UIAppearance (UIView)](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIAppearance_Protocol/Reference/Reference.html#//apple_ref/occ/intf/UIAppearance) [UIAppearanceContainer (UIView)](https://developer.apple.com/library/ios/recipes/UIAppearanceContainer_Protocol/Reference/Reference.html#//apple_ref/occ/intf/UIAppearanceContainer) [UIDynamicItem (UIView)](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIDynamicItem_Protocol/Reference/Reference.html#//apple_ref/occ/intf/UIDynamicItem) [NSObject (NSObject)](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Protocols/NSObject_Protocol/Reference/NSObject.html#//apple_ref/occ/intf/NSObject) |
| **Framework** | /System/Library/Frameworks/[UIKit.framework](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIKit_Framework/_index.html" \l "//apple_ref/doc/uid/TP40006955) |
| **Availability** | Available in iOS 6.0 and later. |
| **Companion guide** | [“Collection Views”](https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/UIKitUICatalog/UICollectionView.html#//apple_ref/doc/uid/TP40012857-UICollectionView) |
| **Declared in** | UICollectionView.h |
| **Related sample code** | [CollectionView-Simple](https://developer.apple.com/library/ios/samplecode/CollectionView-Simple/Introduction/Intro.html#//apple_ref/doc/uid/DTS40012860)  [iAdSuite](https://developer.apple.com/library/ios/samplecode/iAdSuite/Introduction/Intro.html#//apple_ref/doc/uid/DTS40010198)  [MyImagePicker](https://developer.apple.com/library/ios/samplecode/MyImagePicker/Introduction/Intro.html#//apple_ref/doc/uid/DTS40010135) |

## Overview

The UICollectionView class manages an ordered collection of data items and presents them using customizable layouts. Collection views provide the same general function as table views except that a collection view is able to support more than just single-column layouts. Collection views support customizable layouts that can be used to implement multi-column grids, tiled layouts, circular layouts, and many more. You can even change the layout of a collection view dynamically if you want.

UICollectionView类管理了数据项的有序集合，并使用自定义布局展示它们。集合视图提供与表视图一样的通用方法，此外集合视图还能够支持多列的布局。集合视图支持自定义布局，可以实现多列网格、平铺布局、圆形布局等等。如果你想，你甚至可以动态的改变集合视图的布局。

When adding a collection view to your user interface, your app’s main job is to manage the data associated with that collection view. The collection view gets its data from the data source object, which is an object that conforms to the [UICollectionViewDataSource](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewDataSource_protocol/Reference/Reference.html" \l "//apple_ref/occ/intf/UICollectionViewDataSource" \t "_self) protocol and is provided by your app. Data in the collection view is organized into individual items, which can then be grouped into sections for presentation. An item is the smallest unit of data you want to present. For example, in a photos app, an item might be a single image. The collection view presents items onscreen using a cell, which is an instance of the [UICollectionViewCell](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewCell_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewCell" \t "_self) class that your data source configures and provides.

在用户界面中添加集合视图，你的主要工作是管理与集合视图相关联的数据。集合视图从数据源对象获得数据，数据源对象是一个由你的应用提供的遵守[UICollectionViewDataSource](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewDataSource_protocol/Reference/Reference.html#//apple_ref/occ/intf/UICollectionViewDataSource)协议的对象。集合视图中的数据被组织成一个个独立的item，然后被分组成一个个section以展示。一个item是你展示的最小数据单元。例如在照片应用中，一个item可能是一张独立的图片。集合视图使用cell把item展示到屏幕上，cell是[UICollectionViewCell](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewCell_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewCell" \t "_self)类的实例，由你的数据源提供和配置。

In addition to its cells, a collection view can present data using other types of views too. These supplementary views can be things like section headers and footers that are separate from the individual cells but still convey some sort of information. Support for supplementary views is optional and defined by the collection view’s layout object, which is also responsible for defining the placement of those views.

除了cell之外，集合视图也能使用其他类型的视图用于展示数据。这些补充的视图可以是像段头段脚之类的东西，它们与独立cell不同，却仍然传递出一些信息。支持补充视图是可选的，由集合视图的布局对象决定，布局对象还负责决定这些视图位置。

Besides embedding it in your user interface, you use the methods of UICollectionView object to ensure that the visual presentation of items matches the order in your data source object. Thus, whenever you add, delete, or rearrange data in your collection, you use the methods of this class to insert, delete, and rearrange the corresponding cells. You also use the collection view object to manage the selected items, although for this behavior the collection view works with its associated [delegate](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/delegate) object.

在将其嵌入你的用户界面之后，你需要使用UICollectionView对象的方法保证各item的视觉展示与数据源对象相匹配。也就是说，无论何时你添加、删除或者重新排列了集合的数据，你必须使用这个类的方法插入、删除和重新排列相关联的cell。你也可以使用集合视图对象管理选中的item，但需要使用与集合视图相关联的[delegate](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/delegate)对象。

### Collection Views and Layout Objects

A very important object associated with a collection view is the layout object, which is a subclass of the [UICollectionViewLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewLayout" \t "_self) class. The layout object is responsible for defining the organization and location of all cells and supplementary views inside the collection view. Although it defines their locations, the layout object does not actually apply that information to the corresponding views. Because the creation of cells and supplementary views involves coordination between the collection view and your data source object, the collection view actually applies layout information to the views. Thus, in a sense, the layout object is like another data source, only providing visual information instead of item data.

与集合视图相关的一直非常重要的对象就是布局对象，[UICollectionViewLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewLayout" \t "_self)类的子类。布局对象负责决定所有在集合视图里的cell和补充视图的组织和定位。虽然布局视图决定了它们的位置，但是布局视图实际上并没有应用这些相关视图的信息。因为cell和补充视图的创建是包含集合视图与数据源对象的协作的，集合视图实际上应用了这些视图的布局信息。因此，在某种意义上说，布局对象就像另一个数据源，只不过提供的是视觉信息而不是item数据。

You normally specify a layout object when creating a collection view but you can also change the layout of a collection view dynamically. The layout object is stored in the[collectionViewLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/collectionViewLayout) property. Setting this property directly updates the layout immediately, without animating the changes. If you want to animate the changes, you must call the [setCollectionViewLayout:animated:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionView/setCollectionViewLayout:animated:completion:) method instead.

通常是在创建集合视图时就指定布局对象，但是你也可以动态的改变集合视图的布局。布局对象存在[collectionViewLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/collectionViewLayout)属性中。直接设置这个属性会立即更新布局，不含改变动画。如果你想要带动画的改变，你必须调用[setCollectionViewLayout:animated:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/setCollectionViewLayout:animated:completion:)方法。

If you want to create an interactive transition—one that is driven by a gesture recognizer or touch events—use the[startInteractiveTransitionToCollectionViewLayout:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/startInteractiveTransitionToCollectionViewLayout:completion:) method to change the layout object. That method installs an intermediate layout object whose purpose is to work with your gesture recognizer or event-handling code to track the transition progress. When your event-handling code determines that the transition is finished, it calls the [finishInteractiveTransition](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/finishInteractiveTransition) or [cancelInteractiveTransition](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/cancelInteractiveTransition) method to remove the intermediate layout object and install the intended target layout object.

如果你想要建立一个交互过渡——通过手势识别或者触碰事件来驱动——使用[startInteractiveTransitionToCollectionViewLayout:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionView/startInteractiveTransitionToCollectionViewLayout:completion:) 方法来改变布局对象。这个方法安装了一个中间的布局对象，它可以随着手势识别或者事件捕获代码跟踪过渡过程。当你的事件捕获代码决定过渡结束时，可以调用[finishInteractiveTransition](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/finishInteractiveTransition)方法或[cancelInteractiveTransition](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionView/cancelInteractiveTransition)方法移除中间布局对象，而安装预期的目标布局对象。

### Creating Cells and Supplementary Views

The collection view’s data source object provides both the content for items and the views used to present that content. When the collection view first loads its content, it asks its data source to provide a view for each visible item. To simplify the creation process for your code, the collection view requires that you always dequeue views, rather than create them explicitly in your code. There are two methods for dequeueing views. The one you use depends on which type of view has been requested:

集合视图的数据源对象提供了item和展示它们的视图的内容。当集合对象第一次加载内容时，它会请求其数据源提供每一个可见item的视图。为了简化编码创建过程，集合视图需要dequeue视图，而不是在代码中直接创建它们。有两种方法dequeue视图。使用哪一种取决于你需要哪一种视图：

* Use the [dequeueReusableCellWithReuseIdentifier:forIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/dequeueReusableCellWithReuseIdentifier:forIndexPath:) to get a cell for an item in the collection view.
* 使用[dequeueReusableCellWithReuseIdentifier:forIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionView/dequeueReusableCellWithReuseIdentifier:forIndexPath:) 方法从集合视图中给一个item获取一个cell。
* Use the [dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath:) method to get a supplementary view requested by the layout object.
* 使用[dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionView/dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath:)方法获取布局对象需要的补充视图。

Before you call either of these methods, you must tell the collection view how to create the corresponding view if one does not already exist. For this, you must register either a class or a nib file with the collection view. For example, when registering cells, you use the [registerClass:forCellWithReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/registerClass:forCellWithReuseIdentifier:) or[registerNib:forCellWithReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionView/registerNib:forCellWithReuseIdentifier:) method. As part of the registration process, you specify the reuse identifier that identifies the purpose of the view. This is the same string you use when dequeueing the view later.

在你使用任何一种方法之前，你必须告知集合视图如何创建相关联的视图，如果这个视图尚未存在。因此，你必须给集合视图注册一个类或一个nib文件。例如，当注册cell时，使用[registerClass:forCellWithReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionView/registerClass:forCellWithReuseIdentifier:)或[registerNib:forCellWithReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionView/registerNib:forCellWithReuseIdentifier:)方法。作为注册过程的一部分，你需要制定重用标识符以标识该视图的用途。这个字符串与稍后你在dequeue时使用的相同。

After dequeueing the appropriate view in your delegate method, configure its content and return it to the collection view for use. After getting the layout information from the layout object, the collection view applies it to the view and displays it.

在你的代理方法中dequeue合适的视图以后，配置其内容，并将其返回给集合视图使用。在从布局对象获取布局信息以后，集合视图把它应用到视图上并显示出来。

For more information about implementing the data source methods to create and configure views, see *[UICollectionViewDataSource Protocol Reference](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewDataSource_protocol/Reference/Reference.html" \l "//apple_ref/doc/uid/TP40012175" \t "_self)*.

For more information about appearance and behavior configuration, see [“Collection Views”](https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/UIKitUICatalog/UICollectionView.html#//apple_ref/doc/uid/TP40012857-UICollectionView).

## Tasks

### Initializing a Collection View

* [– initWithFrame:collectionViewLayout:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/initWithFrame:collectionViewLayout:)

### Configuring the Collection View

* [delegate](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/delegate)*property*
* [dataSource](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/dataSource)*property*
* [backgroundView](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/backgroundView)*property*

### Creating Collection View Cells

* [– registerClass:forCellWithReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/registerClass:forCellWithReuseIdentifier:)
* [– registerNib:forCellWithReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/registerNib:forCellWithReuseIdentifier:)
* [– registerClass:forSupplementaryViewOfKind:withReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/registerClass:forSupplementaryViewOfKind:withReuseIdentifier:)
* [– registerNib:forSupplementaryViewOfKind:withReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/registerNib:forSupplementaryViewOfKind:withReuseIdentifier:)
* [– dequeueReusableCellWithReuseIdentifier:forIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/dequeueReusableCellWithReuseIdentifier:forIndexPath:)
* [– dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath:)

### Changing the Layout

* [collectionViewLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/collectionViewLayout)*property*
* [– setCollectionViewLayout:animated:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/setCollectionViewLayout:animated:)
* [– setCollectionViewLayout:animated:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/setCollectionViewLayout:animated:completion:)
* [– startInteractiveTransitionToCollectionViewLayout:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/startInteractiveTransitionToCollectionViewLayout:completion:)
* [– finishInteractiveTransition](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/finishInteractiveTransition)
* [– cancelInteractiveTransition](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/cancelInteractiveTransition)

### Reloading Content

* [– reloadData](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/reloadData)
* [– reloadSections:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/reloadSections:)
* [– reloadItemsAtIndexPaths:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/reloadItemsAtIndexPaths:)

### Getting the State of the Collection View

* [– numberOfSections](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/numberOfSections)
* [– numberOfItemsInSection:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/numberOfItemsInSection:)
* [– visibleCells](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/visibleCells)

### Inserting, Moving, and Deleting Items

* [– insertItemsAtIndexPaths:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/insertItemsAtIndexPaths:)
* [– moveItemAtIndexPath:toIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/moveItemAtIndexPath:toIndexPath:)
* [– deleteItemsAtIndexPaths:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/deleteItemsAtIndexPaths:)

### Inserting, Moving, and Deleting Sections

* [– insertSections:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/insertSections:)
* [– moveSection:toSection:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/moveSection:toSection:)
* [– deleteSections:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/deleteSections:)

### Managing the Selection

* [allowsSelection](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/allowsSelection)*property*
* [allowsMultipleSelection](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/allowsMultipleSelection)*property*
* [– indexPathsForSelectedItems](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/indexPathsForSelectedItems)
* [– selectItemAtIndexPath:animated:scrollPosition:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/selectItemAtIndexPath:animated:scrollPosition:)
* [– deselectItemAtIndexPath:animated:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/deselectItemAtIndexPath:animated:)

### Locating Items in the Collection View

* [– indexPathForItemAtPoint:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/indexPathForItemAtPoint:)
* [– indexPathsForVisibleItems](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/indexPathsForVisibleItems)
* [– indexPathForCell:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/indexPathForCell:)
* [– cellForItemAtIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/cellForItemAtIndexPath:)

### Getting Layout Information

* [– layoutAttributesForItemAtIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/layoutAttributesForItemAtIndexPath:)
* [– layoutAttributesForSupplementaryElementOfKind:atIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/layoutAttributesForSupplementaryElementOfKind:atIndexPath:)

### Scrolling an Item Into View

* [– scrollToItemAtIndexPath:atScrollPosition:animated:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/scrollToItemAtIndexPath:atScrollPosition:animated:)

### Animating Multiple Changes to the Collection View

* [– performBatchUpdates:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/performBatchUpdates:completion:)

## Properties

### allowsMultipleSelection

A Boolean value that determines whether users can select more than one item in the collection view.

@property (nonatomic) BOOL allowsMultipleSelection

##### Discussion

This property controls whether multiple items can be selected simultaneously. The default value of this property is NO.

When the value of this property is YES, tapping a cell adds it to the current selection (assuming the delegate permits the cell to be selected). Tapping the cell again removes it from the selection.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [@property allowsSelection](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/allowsSelection)

##### Declared In

UICollectionView.h

### allowsSelection

A Boolean value that indicates whether users can select items in the collection view.

@property (nonatomic) BOOL allowsSelection

##### Discussion

If the value of this property is YES (the default), users can select items. If you want more fine-grained control over the selection of items, you must provide a delegate object and implement the appropriate methods of the [UICollectionViewDelegate](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewDelegate_protocol/Reference/Reference.html" \l "//apple_ref/occ/intf/UICollectionViewDelegate" \t "_self) protocol.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [@property allowsMultipleSelection](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/allowsMultipleSelection)

##### Declared In

UICollectionView.h

### backgroundView

The view that provides the background appearance.

@property (nonatomic, retain) [UIView](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIView_Class/UIView/UIView.html" \l "//apple_ref/doc/c_ref/UIView) \*backgroundView;

##### Discussion

The view (if any) in this property is positioned underneath all of the other content and sized automatically to fill the entire bounds of the collection view. The background view does not scroll with the collection view’s other content. The collection view maintains a strong reference to the background view object.

This property is nil by default, which displays the background color of the collection view.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### collectionViewLayout

The layout used to organize the collected view’s items.

@property (nonatomic, retain) [UICollectionViewLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html" \l "//apple_ref/doc/c_ref/UICollectionViewLayout) \*collectionViewLayout;

##### Discussion

Assigning a new layout object to this property causes the new layout to be applied (without animations) to the collection view’s items.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### dataSource

The object that provides the data for the collection view.

@property (nonatomic, assign) id <[UICollectionViewDataSource](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewDataSource_protocol/Reference/Reference.html" \l "//apple_ref/occ/intf/UICollectionViewDataSource)> dataSource;

##### Discussion

The data source must adopt the [UICollectionViewDataSource](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewDataSource_protocol/Reference/Reference.html" \l "//apple_ref/occ/intf/UICollectionViewDataSource" \t "_self) protocol. The collection view maintains a weak reference to the data source object.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### delegate

The object that acts as the delegate of the collection view.

@property (nonatomic, assign) id <[UICollectionViewDelegate](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewDelegate_protocol/Reference/Reference.html" \l "//apple_ref/occ/intf/UICollectionViewDelegate)> delegate;

##### Discussion

The delegate must adopt the [UICollectionViewDelegate](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewDelegate_protocol/Reference/Reference.html" \l "//apple_ref/occ/intf/UICollectionViewDelegate" \t "_self) protocol. The collection view maintains a weak reference to the delegate object.

The delegate object is responsible for managing selection behavior and interactions with individual items.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

## Instance Methods

### cancelInteractiveTransition

Tells the collection view to abort an interactive transition and return to its original layout object.

- (void)cancelInteractiveTransition

##### Discussion

Call this method after a call to the startInteractiveTransitionToCollectionViewLayout:completion: method and after you determine through a gesture recognizer or other event-handling code that the user wants to revert to the collection view’s original layout. This method removes the intermediate transition layout object from the collection view and reinstalls the original layout object. It then performs any final animations to get the collection view’s items from their current positions to the positions specified by the original layout object.

After calling this method, you can also remove the gesture recognizer or event-handling code you installed to manage the interactive portions of the transition.

##### Availability

* Available in iOS 7.0 and later.

##### See Also

* [– startInteractiveTransitionToCollectionViewLayout:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/startInteractiveTransitionToCollectionViewLayout:completion:)

##### Declared In

UICollectionView.h

### cellForItemAtIndexPath:

Returns the visible cell object at the specified index path.

- ([UICollectionViewCell](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewCell_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewCell) \*)cellForItemAtIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)indexPath

##### Parameters

indexPath

The index path that specifies the section and item number of the cell.

##### Return Value

The cell object at the corresponding index path or nil if the cell is not visible or indexPath is out of range.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### deleteItemsAtIndexPaths:

Deletes the items at the specified index paths.

- (void)deleteItemsAtIndexPaths:([NSArray](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/NSArray.html" \l "//apple_ref/doc/c_ref/NSArray) \*)indexPaths

##### Parameters

indexPaths

An array of [NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html" \l "//apple_ref/occ/cl/NSIndexPath" \t "_self) objects, each of which contains a section index and item index for the item you want to delete from the collection view. This parameter must not benil.

##### Discussion

Use this method to remove items from the collection view. You might do this when you remove the items from your data source object or in response to user interactions with the collection view. The collection view updates the layout of the remaining items to account for the deletions, animating the remaining items into position as needed.

You can also call this method from a block passed to the [performBatchUpdates:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/performBatchUpdates:completion:) method when you want to animate multiple separate changes into place at the same time. See the description of that method for more information.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### deleteSections:

Deletes the sections at the specified indexes.

- (void)deleteSections:([NSIndexSet](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexSet_Class/Reference/Reference.html" \l "//apple_ref/doc/c_ref/NSIndexSet) \*)sections

##### Parameters

sections

The indexes of the sections you want to delete. This parameter must not be nil.

##### Discussion

Use this method to remove the sections and their items from the collection view. You might do this when you remove the sections from your data source object or in response to user interactions with the collection view. The collection view updates the layout of the remaining sections and items to account for the deletions, animating the remaining items into position as needed.

You can also call this method from a block passed to the [performBatchUpdates:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/performBatchUpdates:completion:) method when you want to animate multiple separate changes into place at the same time. See the description of that method for more information.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### dequeueReusableCellWithReuseIdentifier:forIndexPath:

Returns a reusable cell object located by its identifier

- (id)dequeueReusableCellWithReuseIdentifier:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)identifier forIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath)\*)indexPath

##### Parameters

identifier

The reuse identifier for the specified cell. This parameter must not be nil.

indexPath

The index path specifying the location of the cell. The data source receives this information when it is asked for the cell and should just pass it along. This method uses the index path to perform additional configuration based on the cell’s position in the collection view.

##### Return Value

A valid [UICollectionReusableView](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionReusableView_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionReusableView" \t "_self) object.

##### Discussion

Call this method from your data source object when asked to provide a new cell for the collection view. This method dequeues an existing cell if one is available or creates a new one based on the class or nib file you previously registered.

**Important:** You must register a class or nib file using the registerClass:forCellWithReuseIdentifier: or registerNib:forCellWithReuseIdentifier: method before calling this method.

If you registered a class for the specified identifier and a new cell must be created, this method initializes the cell by calling its [initWithFrame:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIView_Class/UIView/UIView.html" \l "//apple_ref/occ/instm/UIView/initWithFrame:" \t "_self) method. For nib-based cells, this method loads the cell object from the provided nib file. If an existing cell was available for reuse, this method calls the cell’s [prepareForReuse](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionReusableView_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionReusableView/prepareForReuse" \t "_self) method instead.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– registerClass:forCellWithReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/registerClass:forCellWithReuseIdentifier:)
* [– registerNib:forCellWithReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/registerNib:forCellWithReuseIdentifier:)

##### Related Sample Code

* [CollectionView-Simple](https://developer.apple.com/library/ios/samplecode/CollectionView-Simple/Introduction/Intro.html#//apple_ref/doc/uid/DTS40012860)
* [iAdSuite](https://developer.apple.com/library/ios/samplecode/iAdSuite/Introduction/Intro.html#//apple_ref/doc/uid/DTS40010198)
* [MyImagePicker](https://developer.apple.com/library/ios/samplecode/MyImagePicker/Introduction/Intro.html#//apple_ref/doc/uid/DTS40010135)

##### Declared In

UICollectionView.h

### dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath:

Returns a reusable supplementary view located by its identifier and kind.

- (id)dequeueReusableSupplementaryViewOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString)\*)elementKind withReuseIdentifier:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)identifier forIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath)\*)indexPath

##### Parameters

elementKind

The kind of supplementary view to retrieve. This value is defined by the layout object. This parameter must not be nil.

identifier

The reuse identifier for the specified view. This parameter must not be nil.

indexPath

The index path specifying the location of the supplementary view in the collection view. The data source receives this information when it is asked for the view and should just pass it along. This method uses the information to perform additional configuration based on the view’s position in the collection view.

##### Return Value

A valid [UICollectionReusableView](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionReusableView_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionReusableView" \t "_self) object.

##### Discussion

Call this method from your data source object when asked to provide a new supplementary view for the collection view. This method dequeues an existing view if one is available or creates a new one based on the class or nib file you previously registered.

**Important:** You must register a class or nib file using the registerClass:forSupplementaryViewOfKind:withReuseIdentifier: orregisterNib:forSupplementaryViewOfKind:withReuseIdentifier: method before calling this method. You can also register a set of default supplementary views with the layout object using the [registerClass:forDecorationViewOfKind:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/registerClass:forDecorationViewOfKind:) or [registerNib:forDecorationViewOfKind:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionViewLayout/registerNib:forDecorationViewOfKind:) method.

If you registered a class for the specified identifier and a new cell must be created, this method initializes the cell by calling its [initWithFrame:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIView_Class/UIView/UIView.html" \l "//apple_ref/occ/instm/UIView/initWithFrame:" \t "_self) method. For nib-based cells, this method loads the cell object from the provided nib file. If an existing cell was available for reuse, this method calls the cell’s [prepareForReuse](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionReusableView_class/Reference/Reference.html" \l "//apple_ref/occ/instm/UICollectionReusableView/prepareForReuse" \t "_self) method instead.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### deselectItemAtIndexPath:animated:

Deselects the item at the specified index.

- (void)deselectItemAtIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)indexPath animated:(BOOL)animated

##### Parameters

indexPath

The index path of the item to select. Specifying nil for this parameter removes the current selection.

animated

Specify YES to animate the change in the selection or NO to make the change without animating it.

##### Discussion

If the [allowsSelection](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/allowsSelection) property is NO, calling this method has no effect.

This method does not cause any selection-related delegate methods to be called.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### finishInteractiveTransition

Tells the collection view to finish an interactive transition by installing the intended target layout.

- (void)finishInteractiveTransition

##### Discussion

Call this method after a call to the startInteractiveTransitionToCollectionViewLayout:completion: method and after you determine through a gesture recognizer or other event-handling code that the user wants to transition to the new layout. This method removes the intermediate transition layout object from the collection view and installs the intended target layout object. It then performs any final animations to get the collection view’s items from their current positions to the positions specified by the newly installed layout object.

After calling this method, you can also remove the gesture recognizer or event-handling code you installed to manage the interactive portions of the transition.

##### Availability

* Available in iOS 7.0 and later.

##### See Also

* [– startInteractiveTransitionToCollectionViewLayout:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/startInteractiveTransitionToCollectionViewLayout:completion:)

##### Declared In

UICollectionView.h

### indexPathForCell:

Returns the index path of the specified cell.

- ([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html" \l "//apple_ref/doc/c_ref/NSIndexPath) \*)indexPathForCell:([UICollectionViewCell](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewCell_class/Reference/Reference.html" \l "//apple_ref/doc/c_ref/UICollectionViewCell) \*)cell

##### Parameters

cell

The cell object whose index path you want.

##### Return Value

The index path of the cell or nil if the specified cell is not in the collection view.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### indexPathForItemAtPoint:

Returns the index path of the item at the specified point in the collection view.

- ([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html" \l "//apple_ref/doc/c_ref/NSIndexPath) \*)indexPathForItemAtPoint:([CGPoint](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CGGeometry/Reference/reference.html" \l "//apple_ref/doc/c_ref/CGPoint))point

##### Parameters

point

A point in the collection view’s coordinate system.

##### Return Value

The index path of the item at the specified point or nil if no item was found at the specified point.

##### Discussion

This method relies on the layout information provided by the associated layout object to determine which item contains the point.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### indexPathsForSelectedItems

Returns the index paths for the selected items.

- ([NSArray](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/NSArray.html" \l "//apple_ref/doc/c_ref/NSArray) \*)indexPathsForSelectedItems

##### Return Value

An array of [NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html" \l "//apple_ref/occ/cl/NSIndexPath" \t "_self) objects, each of which corresponds to a single selected item. If there are no selected items, this method returns an empty array.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### indexPathsForVisibleItems

Returns an array of the visible items in the collection view.

- ([NSArray](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/NSArray.html" \l "//apple_ref/doc/c_ref/NSArray) \*)indexPathsForVisibleItems

##### Return Value

An array of [NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html" \l "//apple_ref/occ/cl/NSIndexPath" \t "_self) objects, each of which corresponds to a visible cell in the collection view. This array does not include any supplementary views that are currently visible. If there are no visible items, this method returns an empty array.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– visibleCells](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/visibleCells)

##### Declared In

UICollectionView.h

### initWithFrame:collectionViewLayout:

Initializes and returns a newly allocated collection view object with the specified frame and layout.

- (id)initWithFrame:([CGRect](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CGGeometry/Reference/reference.html#//apple_ref/doc/c_ref/CGRect))frame collectionViewLayout:([UICollectionViewLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayout) \*)layout

##### Parameters

frame

The frame rectangle for the collection view, measured in points. The origin of the frame is relative to the superview in which you plan to add it. This frame is passed to the superclass during initialization.

layout

The layout object to use for organizing items. The collection view stores a strong reference to the specified object. You may specify nil for this parameter.

##### Return Value

An initialized collection view object or nil if the object could not be created.

##### Discussion

Use this method when initializing a collection view object programmatically. If you specify nil for the layout parameter, you must assign a layout object to the[collectionViewLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/collectionViewLayout) property before displaying the collection view onscreen. If you do not, the collection view will be unable to present any items onscreen.

This method is the designated initializer.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### insertItemsAtIndexPaths:

Inserts new items at the specified index paths.

- (void)insertItemsAtIndexPaths:([NSArray](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/NSArray.html" \l "//apple_ref/doc/c_ref/NSArray) \*)indexPaths

##### Parameters

indexPaths

An array of [NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html" \l "//apple_ref/occ/cl/NSIndexPath" \t "_self) objects, each of which contains a section index and item index at which to insert a new cell. This parameter must not be nil.

##### Discussion

Call this method to insert one or more new items into the collection view. You might do this when your data source object receives data for new items or in response to user interactions with the collection view. The collection view gets the layout information for the new cells as part of calling this method. And if the layout information indicates that the cells should appear onscreen, the collection view asks your data source to provide the appropriate views, animating them into position as needed.

You can also call this method from a block passed to the [performBatchUpdates:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/performBatchUpdates:completion:) method when you want to animate multiple separate changes into place at the same time. See the description of that method for more information.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### insertSections:

Inserts new sections at the specified indexes.

- (void)insertSections:([NSIndexSet](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexSet_Class/Reference/Reference.html" \l "//apple_ref/doc/c_ref/NSIndexSet) \*)sections

##### Parameters

sections

An array of [NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html" \l "//apple_ref/occ/cl/NSIndexPath" \t "_self) objects, each of which contains the index of a section you want to insert. This parameter must not be nil.

##### Discussion

Use this method to insert one or more sections into the collection view. This method adds the sections, and it is up to your data source to report the number of items in each section when asked for the information. The collection view then uses that information to get updated layout attributes for the newly inserted sections and items. If the insertions cause a change in the collection view’s visible content, those changes are animated into place.

You can also call this method from a block passed to the [performBatchUpdates:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/performBatchUpdates:completion:) method when you want to animate multiple separate changes into place at the same time. See the description of that method for more information.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### layoutAttributesForItemAtIndexPath:

Returns the layout information for the item at the specified index path.

- ([UICollectionViewLayoutAttributes](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutAttributes_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutAttributes) \*)layoutAttributesForItemAtIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)indexPath

##### Parameters

indexPath

The index path of the item.

##### Return Value

The layout attributes for the item or nil if no item exists at the specified path.

##### Discussion

Use this method to retrieve the layout information for a particular item. You should always use this method instead of querying the layout object directly.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### layoutAttributesForSupplementaryElementOfKind:atIndexPath:

Returns the layout information for the specified supplementary view.

- ([UICollectionViewLayoutAttributes](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayoutAttributes_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutAttributes) \*)layoutAttributesForSupplementaryElementOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)kind atIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)indexPath

##### Parameters

kind

A string specifying the kind of supplementary view whose layout attributes you want. Layout classes are responsible for defining the kinds of supplementary views they support.

indexPath

The index path of the supplementary view. The interpretation of this value depends on how the layout implements the view. For example, a view associated with a section might contain just a section value.

##### Return Value

The layout attributes of the supplementary view or nil if the specified supplementary view does not exist.

##### Discussion

Use this method to retrieve the layout information for a particular supplementary view. You should always use this method instead of querying the layout object directly.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### moveItemAtIndexPath:toIndexPath:

Moves an item from one location to another in the collection view.

- (void)moveItemAtIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)indexPath toIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)newIndexPath

##### Parameters

indexPath

The index path of the item you want to move. This parameter must not be nil.

newIndexPath

The index path of the item’s new location. This parameter must not be nil.

##### Discussion

Use this method to reorganize existing data items. You might do this when you rearrange the items within your data source object or in response to user interactions with the collection view. You can move items between sections or within the same section. The collection view updates the layout as needed to account for the move, animating cells into position as needed.

You can also call this method from a block passed to the [performBatchUpdates:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/performBatchUpdates:completion:) method when you want to animate multiple separate changes into place at the same time. See the description of that method for more information.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### moveSection:toSection:

Moves a section from one location to another in the collection view.

- (void)moveSection:([NSInteger](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Miscellaneous/Foundation_DataTypes/Reference/reference.html#//apple_ref/doc/c_ref/NSInteger))section toSection:([NSInteger](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Miscellaneous/Foundation_DataTypes/Reference/reference.html#//apple_ref/doc/c_ref/NSInteger))newSection

##### Parameters

section

The index path of the section you want to move. This parameter must not be nil.

newSection

The index path of the section’s new location. This parameter must not be nil.

##### Discussion

Use this method to reorganize existing sections and their contained items. You might do this when you rearrange sections within your data source object or in response to user interactions with the collection view. The collection view updates the layout as needed to account for the move, animating new views into position as needed.

You can also call this method from a block passed to the [performBatchUpdates:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/performBatchUpdates:completion:) method when you want to animate multiple separate changes into place at the same time. See the description of that method for more information.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### numberOfItemsInSection:

Returns the number of items in the specified section.

- ([NSInteger](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Miscellaneous/Foundation_DataTypes/Reference/reference.html" \l "//apple_ref/doc/c_ref/NSInteger))numberOfItemsInSection:([NSInteger](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Miscellaneous/Foundation_DataTypes/Reference/reference.html" \l "//apple_ref/doc/c_ref/NSInteger))section

##### Parameters

section

The index of the section for which you want a count of the items.

##### Return Value

The number of items in the specified section.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### numberOfSections

Returns the number of sections displayed by the collection view.

- ([NSInteger](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Miscellaneous/Foundation_DataTypes/Reference/reference.html" \l "//apple_ref/doc/c_ref/NSInteger))numberOfSections

##### Return Value

The number of sections in the collection view.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### performBatchUpdates:completion:

Animates multiple insert, delete, reload, and move operations as a group.

- (void)performBatchUpdates:(void (^)(void))updates completion:(void (^)(BOOL finished))completion

##### Parameters

updates

The block that performs the relevant insert, delete, reload, or move operations.

completion

A completion handler block to execute when all of the operations are finished. This block takes a single Boolean parameter that contains the value YES if all of the related animations completed successfully or NO if they were interrupted. This parameter may be nil.

##### Discussion

You can use this method in cases where you want to make multiple changes to the collection view in one single animated operation, as opposed to in several separate animations. You might use this method to insert, delete, reload or move cells or use it to change the layout parameters associated with one or more cells. Use the blocked passed in the updatesparameter to specify all of the operations you want to perform.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– insertItemsAtIndexPaths:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/insertItemsAtIndexPaths:)
* [– moveItemAtIndexPath:toIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/moveItemAtIndexPath:toIndexPath:)
* [– deleteItemsAtIndexPaths:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/deleteItemsAtIndexPaths:)
* [– insertSections:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/insertSections:)
* [– moveSection:toSection:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/moveSection:toSection:)
* [– deleteSections:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/deleteSections:)

##### Declared In

UICollectionView.h

### registerClass:forCellWithReuseIdentifier:

Register a class for use in creating new collection view cells.

- (void)registerClass:(Class)cellClass forCellWithReuseIdentifier:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)identifier

##### Parameters

cellClass

The class of a cell that you want to use in the collection view.

identifier

The reuse identifier to associate with the specified class. This parameter must not be nil and must not be an empty string.

##### Discussion

Prior to calling the dequeueReusableCellWithReuseIdentifier:forIndexPath: method of the collection view, you must use this method or theregisterNib:forCellWithReuseIdentifier: method to tell the collection view how to create a new cell of the given type. If a cell of the specified type is not currently in a reuse queue, the collection view uses the provided information to create a new cell object automatically.

If you previously registered a class or nib file with the same reuse identifier, the class you specify in the cellClass parameter replaces the old entry. You may specify nil forcellClass if you want to unregister the class from the specified reuse identifier.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– registerNib:forCellWithReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/registerNib:forCellWithReuseIdentifier:)
* [– dequeueReusableCellWithReuseIdentifier:forIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/dequeueReusableCellWithReuseIdentifier:forIndexPath:)

##### Declared In

UICollectionView.h

### registerClass:forSupplementaryViewOfKind:withReuseIdentifier:

Registers a class for use in creating supplementary views for the collection view.

- (void)registerClass:(Class)viewClass forSupplementaryViewOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)elementKind withReuseIdentifier:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)identifier

##### Parameters

viewClass

The class to use for the supplementary view.

elementKind

The kind of supplementary view to create. This value is defined by the layout object. This parameter must not be nil.

identifier

The reuse identifier to associate with the specified class. This parameter must not be nil and must not be an empty string.

##### Discussion

Prior to calling the dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath: method of the collection view, you must use this method or theregisterNib:forSupplementaryViewOfKind:withReuseIdentifier: method to tell the collection view how to create a supplementary view of the given type. If a view of the specified type is not currently in a reuse queue, the collection view uses the provided information to create a view object automatically.

If you previously registered a class or nib file with the same element kind and reuse identifier, the class you specify in the viewClass parameter replaces the old entry. You may specify nil for viewClass if you want to unregister the class from the specified element kind and reuse identifier.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– registerNib:forSupplementaryViewOfKind:withReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/registerNib:forSupplementaryViewOfKind:withReuseIdentifier:)
* [– dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath:)

##### Declared In

UICollectionView.h

### registerNib:forCellWithReuseIdentifier:

Register a nib file for use in creating new collection view cells.

- (void)registerNib:([UINib](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UINib_Ref/Reference/Reference.html#//apple_ref/doc/c_ref/UINib) \*)nib forCellWithReuseIdentifier:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)identifier

##### Parameters

nib

The nib object containing the cell object. The nib file must contain only one top-level object and that object must be of the type [UICollectionViewCell](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewCell_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewCell" \t "_self).

identifier

The reuse identifier to associate with the specified nib file. This parameter must not be nil and must not be an empty string.

##### Discussion

Prior to calling the dequeueReusableCellWithReuseIdentifier:forIndexPath: method of the collection view, you must use this method or theregisterClass:forCellWithReuseIdentifier: method to tell the collection view how to create a new cell of the given type. If a cell of the specified type is not currently in a reuse queue, the collection view uses the provided information to create a new cell object automatically.

If you previously registered a class or nib file with the same reuse identifier, the object you specify in the nib parameter replaces the old entry. You may specify nil for nib if you want to unregister the nib file from the specified reuse identifier.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– registerClass:forCellWithReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/registerClass:forCellWithReuseIdentifier:)
* [– dequeueReusableCellWithReuseIdentifier:forIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/dequeueReusableCellWithReuseIdentifier:forIndexPath:)

##### Declared In

UICollectionView.h

### registerNib:forSupplementaryViewOfKind:withReuseIdentifier:

Registers a nib file for use in creating supplementary views for the collection view.

- (void)registerNib:([UINib](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UINib_Ref/Reference/Reference.html#//apple_ref/doc/c_ref/UINib) \*)nib forSupplementaryViewOfKind:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)kind withReuseIdentifier:([NSString](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/Reference/NSString.html#//apple_ref/doc/c_ref/NSString) \*)identifier

##### Parameters

nib

The nib object containing the view object. The nib file must contain only one top-level object and that object must be of the type [UICollectionReusableView](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionReusableView_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionReusableView" \t "_self).

kind

The kind of supplementary view to create. The layout defines the types of supplementary views it supports. The value of this string may correspond to one of the predefined kind strings or to a custom string that the layout added to support a new type of supplementary view. This parameter must not be nil.

identifier

The reuse identifier to associate with the specified nib file. This parameter must not be nil and must not be an empty string.

##### Discussion

Prior to calling the dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath: method of the collection view, you must use this method or theregisterClass:forSupplementaryViewOfKind:withReuseIdentifier: method to tell the collection view how to create a supplementary view of the given type. If a view of the specified type is not currently in a reuse queue, the collection view uses the provided information to create a view object automatically.

If you previously registered a class or nib file with the same element kind and reuse identifier, the class you specify in the viewClass parameter replaces the old entry. You may specify nil for nib if you want to unregister the class from the specified element kind and reuse identifier.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– registerClass:forSupplementaryViewOfKind:withReuseIdentifier:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/registerClass:forSupplementaryViewOfKind:withReuseIdentifier:)
* [– dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/dequeueReusableSupplementaryViewOfKind:withReuseIdentifier:forIndexPath:)

##### Declared In

UICollectionView.h

### reloadData

Reloads all of the data for the collection view.

- (void)reloadData

##### Discussion

Call this method to reload all of the items in the collection view. This causes the collection view to discard any currently visible items and redisplay them. For efficiency, the collection view only displays those cells and supplementary views that are visible. If the collection data shrinks as a result of the reload, the collection view adjusts its scrolling offsets accordingly.

You should not call this method in the middle of animation blocks where items are being inserted or deleted. Insertions and deletions automatically cause the table’s data to be updated appropriately.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### reloadItemsAtIndexPaths:

Reloads just the items at the specified index paths.

- (void)reloadItemsAtIndexPaths:([NSArray](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/NSArray.html" \l "//apple_ref/doc/c_ref/NSArray) \*)indexPaths

##### Parameters

indexPaths

An array of [NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html" \l "//apple_ref/occ/cl/NSIndexPath" \t "_self) objects identifying the items you want to update.

##### Discussion

Call this method to selectively reload only the specified items. This causes the collection view to discard any cells associated with those items and redisplay them.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### reloadSections:

Reloads the data in the specified sections of the collection view.

- (void)reloadSections:([NSIndexSet](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexSet_Class/Reference/Reference.html" \l "//apple_ref/doc/c_ref/NSIndexSet) \*)sections

##### Parameters

sections

The indexes of the sections to reload.

##### Discussion

Call this method to selectively reload only the items in the specified sections. This causes the collection view to discard any cells associated with those items and redisplay them.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### scrollToItemAtIndexPath:atScrollPosition:animated:

Scrolls the collection view contents until the specified item is visible.

- (void)scrollToItemAtIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)indexPath atScrollPosition:([UICollectionViewScrollPosition](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewScrollPosition))scrollPosition animated:(BOOL)animated

##### Parameters

indexPath

The index path of the item to scroll into view.

scrollPosition

An option that specifies where the item should be positioned when scrolling finishes. For a list of possible values, see [“UICollectionViewScrollPosition”](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/c/tdef/UICollectionViewScrollPosition).

animated

Specify YES to animate the scrolling behavior or NO to adjust the scroll view’s visible content immediately.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### selectItemAtIndexPath:animated:scrollPosition:

Selects the item at the specified index path and optionally scrolls it into view.

- (void)selectItemAtIndexPath:([NSIndexPath](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSIndexPath_Class/Reference/Reference.html#//apple_ref/doc/c_ref/NSIndexPath) \*)indexPath animated:(BOOL)animated scrollPosition:([UICollectionViewScrollPosition](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewScrollPosition))scrollPosition

##### Parameters

indexPath

The index path of the item to select. Specifying nil for this parameter clears the current selection.

animated

Specify YES to animate the change in the selection or NO to make the change without animating it.

scrollPosition

An option that specifies where the item should be positioned when scrolling finishes. For a list of possible values, see [“UICollectionViewScrollPosition”](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/c/tdef/UICollectionViewScrollPosition).

##### Discussion

If the [allowsSelection](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/allowsSelection) property is NO, calling this method has no effect. If there is an existing selection with a different index path and the [allowsMultipleSelection](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instp/UICollectionView/allowsMultipleSelection) property is NO, calling this method replaces the previous selection.

This method does not cause any selection-related delegate methods to be called.

##### Availability

* Available in iOS 6.0 and later.

##### Declared In

UICollectionView.h

### setCollectionViewLayout:animated:

Changes the collection view’s layout and optionally animates the change.

- (void)setCollectionViewLayout:([UICollectionViewLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayout) \*)layout animated:(BOOL)animated

##### Parameters

layout

The new layout object for the collection view.

animated

Specify YES if you want to animate changes from the current layout to the new layout specified by the layout parameter. Specify NO to make the change without animations.

##### Discussion

This method makes the layout change without further interaction from the user. If you choose to animate the layout change, the animation timing and parameters are controlled by the collection view.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– startInteractiveTransitionToCollectionViewLayout:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/startInteractiveTransitionToCollectionViewLayout:completion:)

##### Declared In

UICollectionView.h

### setCollectionViewLayout:animated:completion:

Changes the collection view’s layout and notifies you when the animations complete.

- (void)setCollectionViewLayout:([UICollectionViewLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayout) \*)layout animated:(BOOL)animated completion:(void (^)(BOOL finished))completion

##### Parameters

layout

The new layout object for the collection view.

animated

Specify YES if you want to animate changes from the current layout to the new layout specified by the layout parameter. Specify NO to make the change without animations.

completion

The block that is executed when the layout transition finishes or is aborted by the user. This block takes the following parameter:

finished

A Boolean indicating whether the transition completed successfully. This parameter is YES if the transition finished and the new layout is installed. It is NO if the user aborted the transition and returned to the old layout.

##### Discussion

This method initiates a layout change programmatically, notifying you when the transition is complete. If you choose to animate the layout change, the animation timing and parameters are controlled by the collection view.

##### Availability

* Available in iOS 7.0 and later.

##### See Also

* [– startInteractiveTransitionToCollectionViewLayout:completion:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/startInteractiveTransitionToCollectionViewLayout:completion:)

##### Declared In

UICollectionView.h

### startInteractiveTransitionToCollectionViewLayout:completion:

Changes the collection view’s current layout using an interactive transition effect.

- ([UICollectionViewTransitionLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewTransitionLayout_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewTransitionLayout) \*)startInteractiveTransitionToCollectionViewLayout:([UICollectionViewLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewLayout_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayout) \*)layout completion:([UICollectionViewLayoutInteractiveTransitionCompletion](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/doc/c_ref/UICollectionViewLayoutInteractiveTransitionCompletion)) completion

##### Parameters

layout

The new layout object for the collected views. This is the layout that you want the collection view to use after the interactive transition is done.

completion

A completion handler to execute after the transition finishes.

##### Return Value

The intermediate transition layout object responsible for managing the interactive transition behavior.

##### Discussion

Call this method when you want to change the layout of your collection view using an intermediate transition. When you call this method, the collection view quietly makes the returned transition layout object its current layout object. It is your responsibility to set up a gesture recognizer or other touch-event handling code to track the transition progress. As progress changes, update the [transitionProgress](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewTransitionLayout_class/Reference/Reference.html" \l "//apple_ref/occ/instp/UICollectionViewTransitionLayout/transitionProgress" \t "_self) property of the transition layout object and invalidate the layout. Invalidating its layout causes the transition layout object to update the position of items based on the new progress value.

When your event-handling code determines that the user has finished the transition to the new layout, call the finishInteractiveTransition method. If your code determines that the user has canceled the transition, call the cancelInteractiveTransition method to revert the changes instead. Calling either of these methods removes the transition layout object from the collection view and installs the appropriate target layout object.

This method returns an instance of the [UICollectionViewTransitionLayout](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewTransitionLayout_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewTransitionLayout" \t "_self) class by default. If you want it to return a custom transition object instead, implement the[collectionView:transitionLayoutForOldLayout:newLayout:](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewDelegate_protocol/Reference/Reference.html#//apple_ref/occ/intfm/UICollectionViewDelegate/collectionView:transitionLayoutForOldLayout:newLayout:) method of your collection view delegate and use that method to return your custom object.

##### Availability

* Available in iOS 7.0 and later.

##### See Also

* [– finishInteractiveTransition](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/finishInteractiveTransition)
* [– cancelInteractiveTransition](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/cancelInteractiveTransition)

##### Declared In

UICollectionView.h

### visibleCells

Returns an array of visible cells currently displayed by the collection view.

- ([NSArray](https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/NSArray.html" \l "//apple_ref/doc/c_ref/NSArray) \*)visibleCells

##### Return Value

An array of [UICollectionViewCell](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionViewCell_class/Reference/Reference.html" \l "//apple_ref/occ/cl/UICollectionViewCell" \t "_self) objects. If no cells are visible, this method returns an empty array.

##### Discussion

This method returns the complete list of visible cells displayed by the collection view.

##### Availability

* Available in iOS 6.0 and later.

##### See Also

* [– indexPathsForVisibleItems](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UICollectionView_class/Reference/Reference.html#//apple_ref/occ/instm/UICollectionView/indexPathsForVisibleItems)

##### Declared In

UICollectionView.h

## Constants

### UICollectionViewScrollPosition

Constants that indicate how to scroll an item into the visible portion of the collection view.

enum {

UICollectionViewScrollPositionNone = 0,

UICollectionViewScrollPositionTop = 1 << 0,

UICollectionViewScrollPositionCenteredVertically = 1 << 1,

UICollectionViewScrollPositionBottom = 1 << 2,

UICollectionViewScrollPositionLeft = 1 << 3,

UICollectionViewScrollPositionCenteredHorizontally = 1 << 4,

UICollectionViewScrollPositionRight = 1 << 5

};

typedef NSUInteger UICollectionViewScrollPosition;

##### Constants

UICollectionViewScrollPositionNone

Do not scroll the item into view.

Available in iOS 6.0 and later.

Declared in UICollectionView.h.

UICollectionViewScrollPositionTop

Scroll so that the item is positioned at the top of the collection view’s bounds. This option is mutually exclusive with theUICollectionViewScrollPositionCenteredVertically and UICollectionViewScrollPositionBottom options.

Available in iOS 6.0 and later.

Declared in UICollectionView.h.

UICollectionViewScrollPositionCenteredVertically

Scroll so that the item is centered vertically in the collection view. This option is mutually exclusive with the UICollectionViewScrollPositionTop andUICollectionViewScrollPositionBottom options.

Available in iOS 6.0 and later.

Declared in UICollectionView.h.

UICollectionViewScrollPositionBottom

Scroll so that the item is positioned at the bottom of the collection view’s bounds. This option is mutually exclusive with the UICollectionViewScrollPositionTop andUICollectionViewScrollPositionCenteredVertically options.

Available in iOS 6.0 and later.

Declared in UICollectionView.h.

UICollectionViewScrollPositionLeft

Scroll so that the item is positioned at the left edge of the collection view’s bounds. This option is mutually exclusive with theUICollectionViewScrollPositionCenteredHorizontally and UICollectionViewScrollPositionRight options.

Available in iOS 6.0 and later.

Declared in UICollectionView.h.

UICollectionViewScrollPositionCenteredHorizontally

Scroll so that the item is centered horizontally in the collection view. This option is mutually exclusive with the UICollectionViewScrollPositionLeft andUICollectionViewScrollPositionRight options.

Available in iOS 6.0 and later.

Declared in UICollectionView.h.

UICollectionViewScrollPositionRight

Scroll so that the item is positioned at the right edge of the collection view’s bounds. This option is mutually exclusive with the UICollectionViewScrollPositionLeft andUICollectionViewScrollPositionCenteredHorizontally options.

Available in iOS 6.0 and later.

Declared in UICollectionView.h.

### UICollectionViewLayoutInteractiveTransitionCompletion

The completion block called at the end of an interactive transition for a collection view.

typedef void(^UICollectionViewLayoutInteractiveTransitionCompletion)(BOOL completed, BOOL finish);

##### Discussion

This completion block takes the following parameters:

completed

A Boolean indicating whether the animations ran to completion.

finish

A Boolean indicating whether the transition finished or was canceled. This parameter is YES if the transition ran to completion and the new layout is installed. It is NO if the user canceled the transition and the old layout is installed.

##### Availability

* Available in iOS 7.0 and later.

##### Declared In

UICollectionView.h