Class

UICollectionViewFlowLayout

A concrete layout object that organizes items into a grid with optional header and footer views for each section.

Declaration

```
class UICollectionViewFlowLayout : UICollectionViewLayout
```

Overview

The items in the collection view flow from one row or column (depending on the scrolling direction) to the next, with each row comprising as many cells as will fit. Cells can be the same sizes or different sizes.

A flow layout works with the collection view's delegate object to determine the size of items, headers, and footers in each section and grid. That delegate object must conform to the `UICollectionViewDelegateFlowLayout` protocol. Use of the delegate allows you to adjust layout information dynamically. For example, you would need to use a delegate object to specify different sizes for items in the grid. If you do not provide a delegate, the flow layout uses the default values you set using the properties of this class.

Flow layouts lay out their content using a fixed distance in one direction and a scrollable distance in the other. For example, in a vertically scrolling grid, the width of the grid content is constrained to the width of the corresponding collection view while the height of the content adjusts dynamically to match the number of sections and items in the grid. The layout is configured to scroll vertically by default but you can configure the scrolling direction using the `scrollDirection` property.

Each section in a flow layout can have its own custom header and footer. To configure the header or footer for a view, you must configure the size of the header or footer to be non zero. You can do this by implementing the appropriate delegate methods or by assigning appropriate values to the `headerReferenceSize` and `footerReferenceSize` properties. If the header or footer size is 0, the corresponding view is not added to the collection view.

Topics

Configuring the Flow Layout

`protocol UICollectionViewDelegateFlowLayout`

The UICollectionViewDelegateFlowLayout protocol defines methods that let you coordinate with a UICollectionViewFlowLayout object to implement a grid-based layout. The methods of this protocol define the size of items and the spacing between items in the grid.

Configuring the Scroll Direction

```
var scrollDirection: UICollectionView.ScrollDirection
```
The scroll direction of the grid.

Configuring the Item Spacing

```
var minimumLineSpacing: CGFloat
```
The minimum spacing to use between lines of items in the grid.

```
var minimumInteritemSpacing: CGFloat
```
The minimum spacing to use between items in the same row.

```
var itemSize: CGSize
```
The default size to use for cells.

```
var estimatedItemSize: CGSize
```
The estimated size of cells in the collection view.

```
var sectionInset: UIEdgeInsets
```
The margins used to lay out content in a section

```
var sectionInsetReference: UICollectionViewFlowLayout.SectionInset
Reference
```

```
enum UICollectionViewFlowLayout.SectionInsetReference
```

Configuring the Supplementary Views

```
var headerReferenceSize: CGSize
```
The default sizes to use for section headers.

```
var footerReferenceSize: CGSize
```
The default sizes to use for section footers.

Pinning Headers and Footers

```
var sectionHeadersPinToVisibleBounds: Bool
```
A Boolean value indicating whether headers pin to the top of the collection view bounds during scrolling.

```
var sectionFootersPinToVisibleBounds: Bool
```
A Boolean value indicating whether footers pin to the bottom of the collection view bounds during scrolling.

Constants

enum UICollectionView.ScrollDirection

Constants indicating the direction of scrolling for the layout.

▤ Flow Layout Supplementary Views

Constants that specify the types of supplementary views that can be presented using a flow layout.

class let automaticSize: CGSize

Relationships

Inherits From

UICollectionViewLayout

Conforms To

CVarArg, Equatable, Hashable

See Also

Layouts

class UICollectionViewLayout

An abstract base class for generating layout information for a collection view.

class UICollectionViewTransitionLayout

A special type of layout object that lets you implement behaviors when changing from one layout to another in your collection view.

class UICollectionViewLayoutAttributes

A layout object that manages the layout-related attributes for a given item in a collection view.

{} Customizing Collection View Layouts

Customize a view layout by changing the size of cells in the flow or implementing a mosaic style.