
UICollectionView

CollectionView

- Cell
- Decoration view
- Supplementary view

UICollectionView

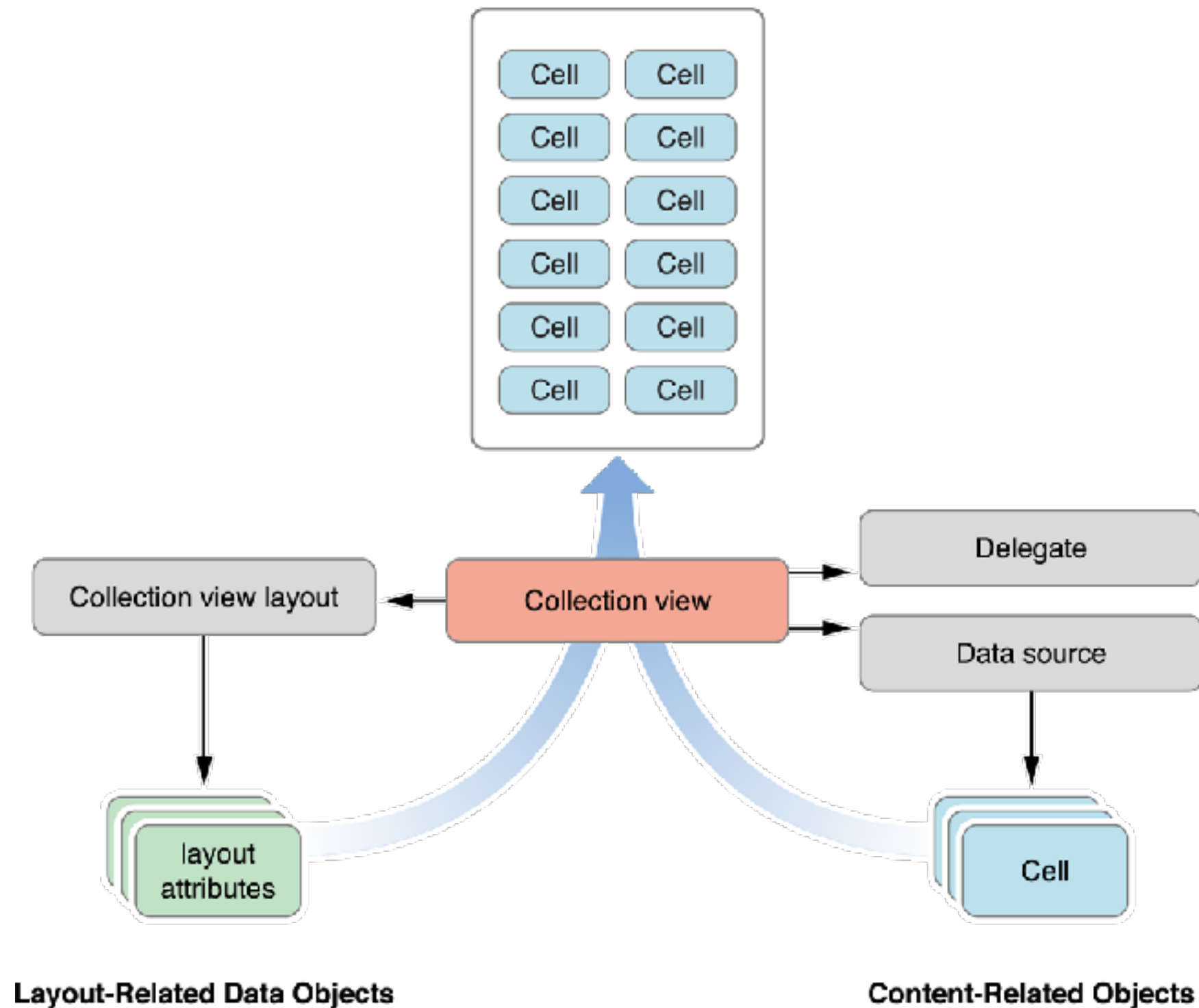
- 데이터 항목의 정렬 된 컬렉션을 관리하고 사용자 정의 레이아웃을 사용하여 데이터 항목을 제공하는 객체입니다



UICollectionView

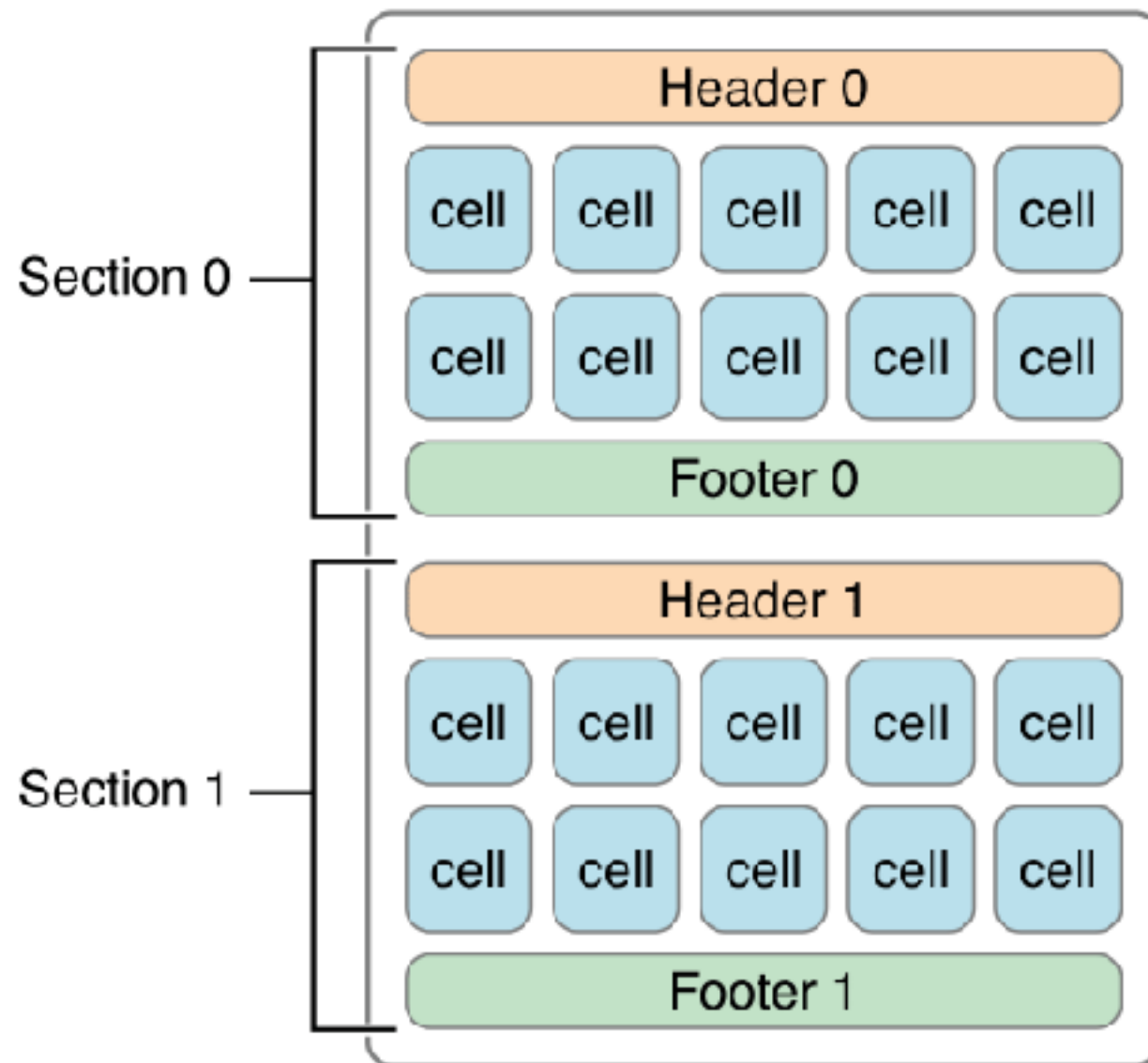
- The classes and protocols for implementing collection views
 - Manager
 - UICollectionView
 - UICollectionViewController
 - Protocol
 - UICollectionViewDataSource
 - UICollectionViewDelegate
 - Presentation
 - UICollectionViewReusableView
 - UICollectionViewCell
 - Layout
 - UICollectionViewLayout
 - UICollectionViewLayoutAttributes
 - UICollectionViewUpdateItem
 - Flow Layout
 - UICollectionViewFlowLayout
 - UICollectionViewDelegateFlowLayout

UICollectionView

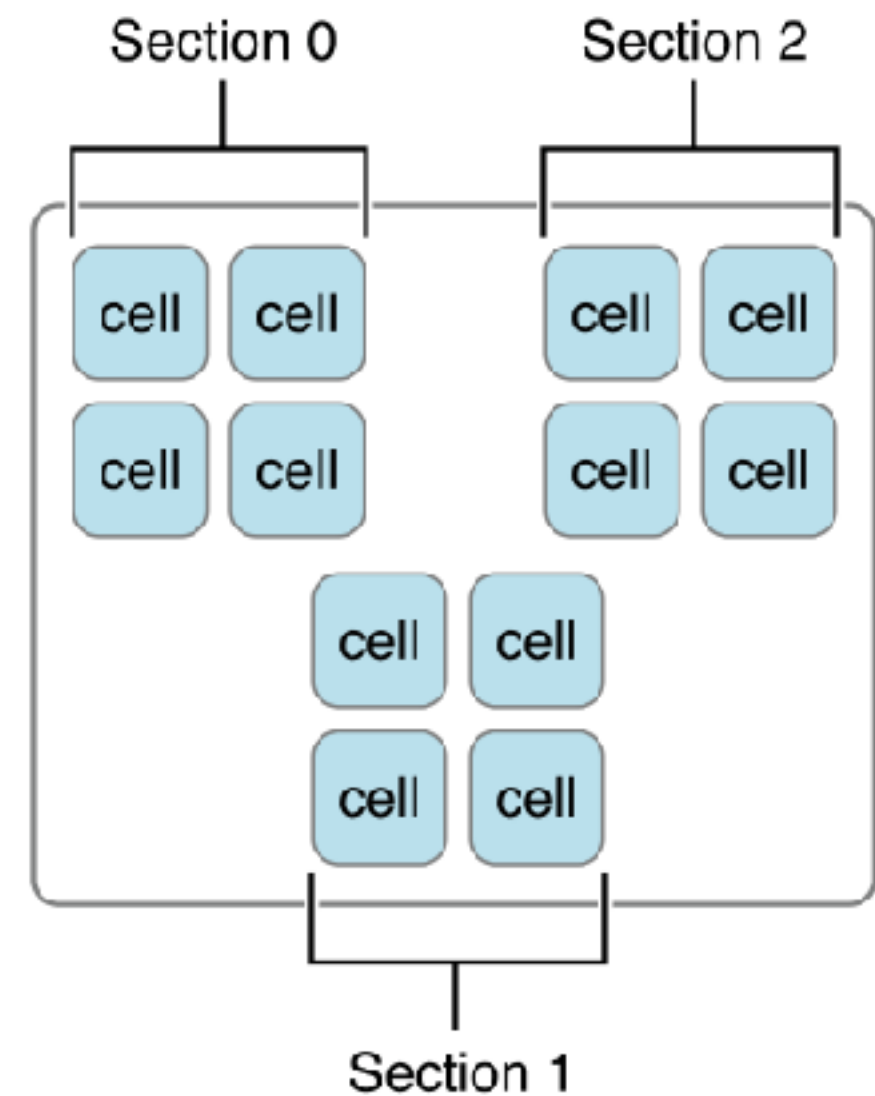


Layout

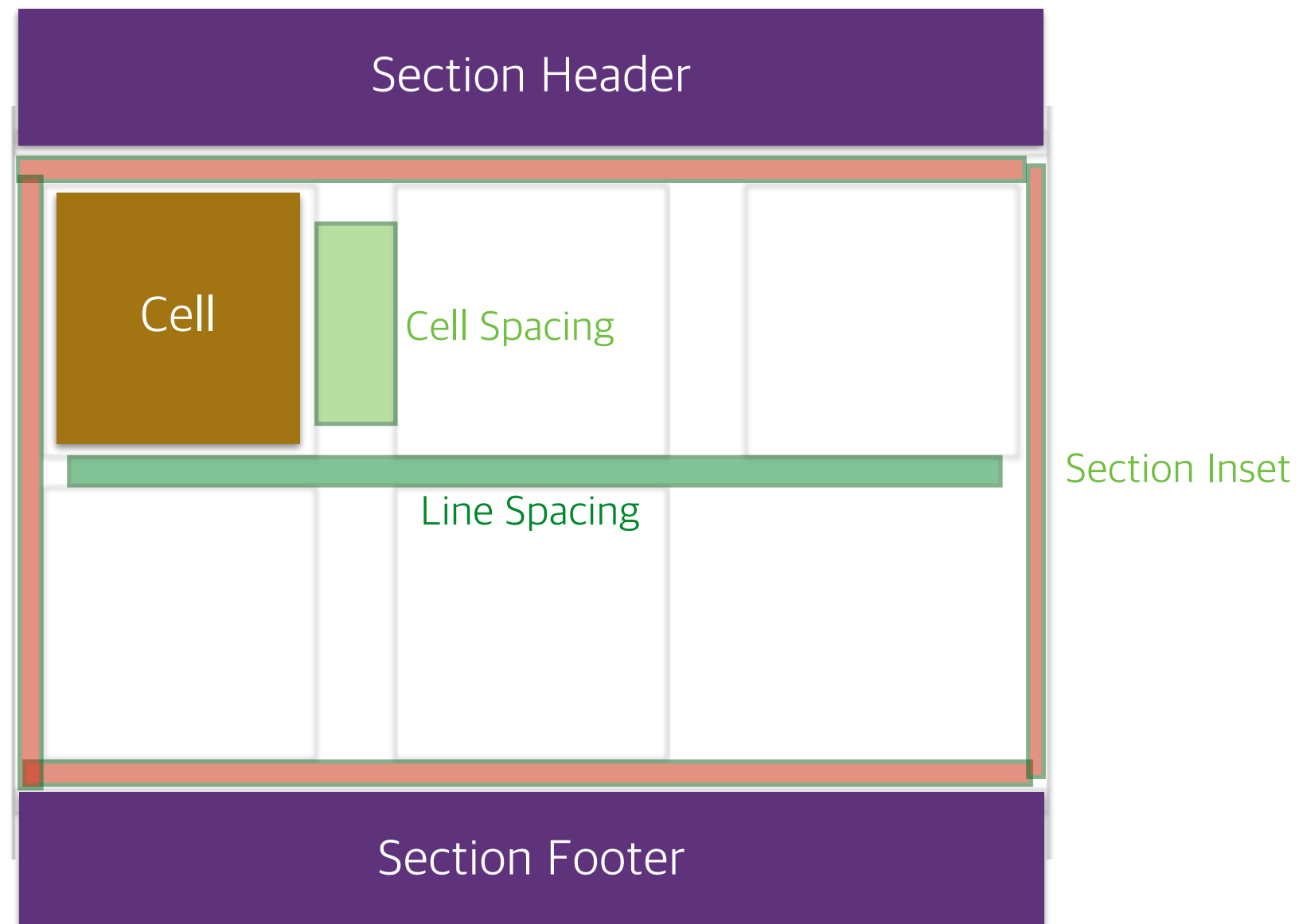
Flow Layout



Custom Layout

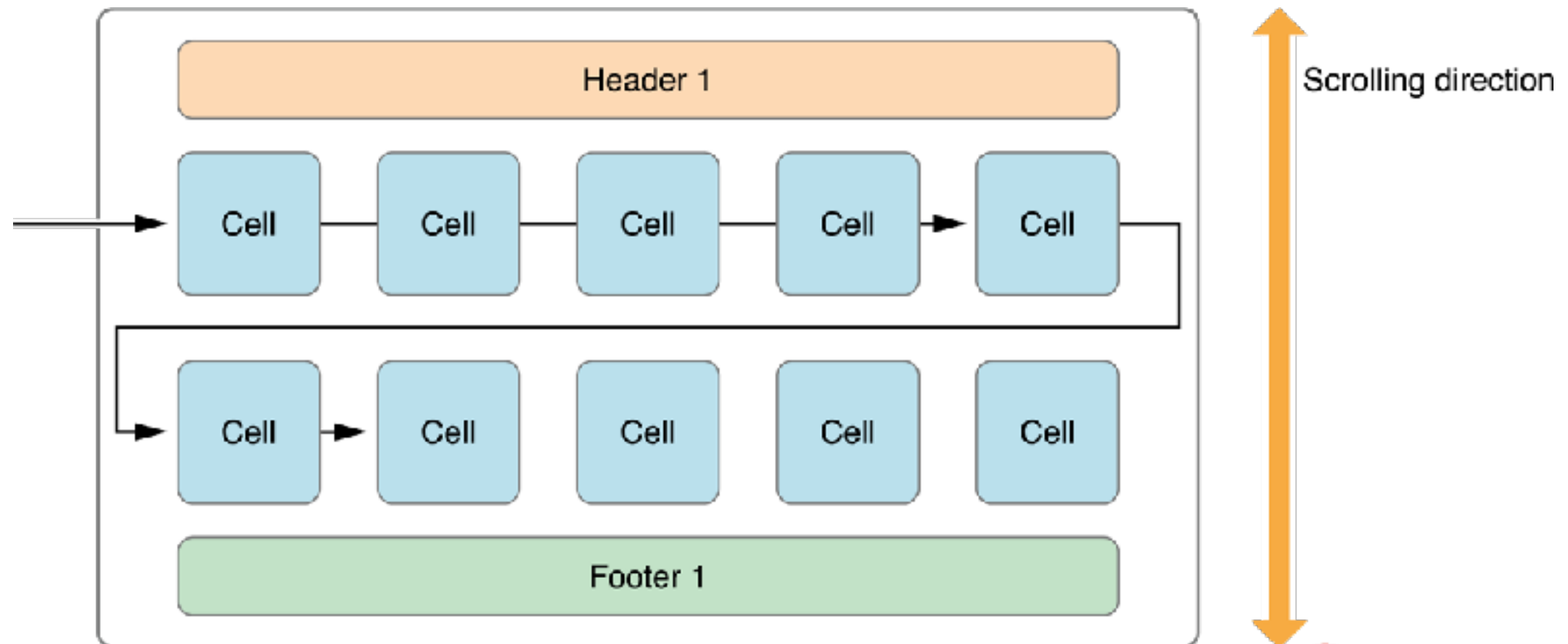


Section Layout



FlowLayout

- UICollectionViewFlowLayout Class에 정의
- Scroll Direction
 1. Vertical
 2. Horizontal



Flow Layout Attributes 정의

- UICollectionViewDelegateFlowLayout protocol 이용
- UICollectionViewDelegate가 정의된 인스턴스에서 작성.

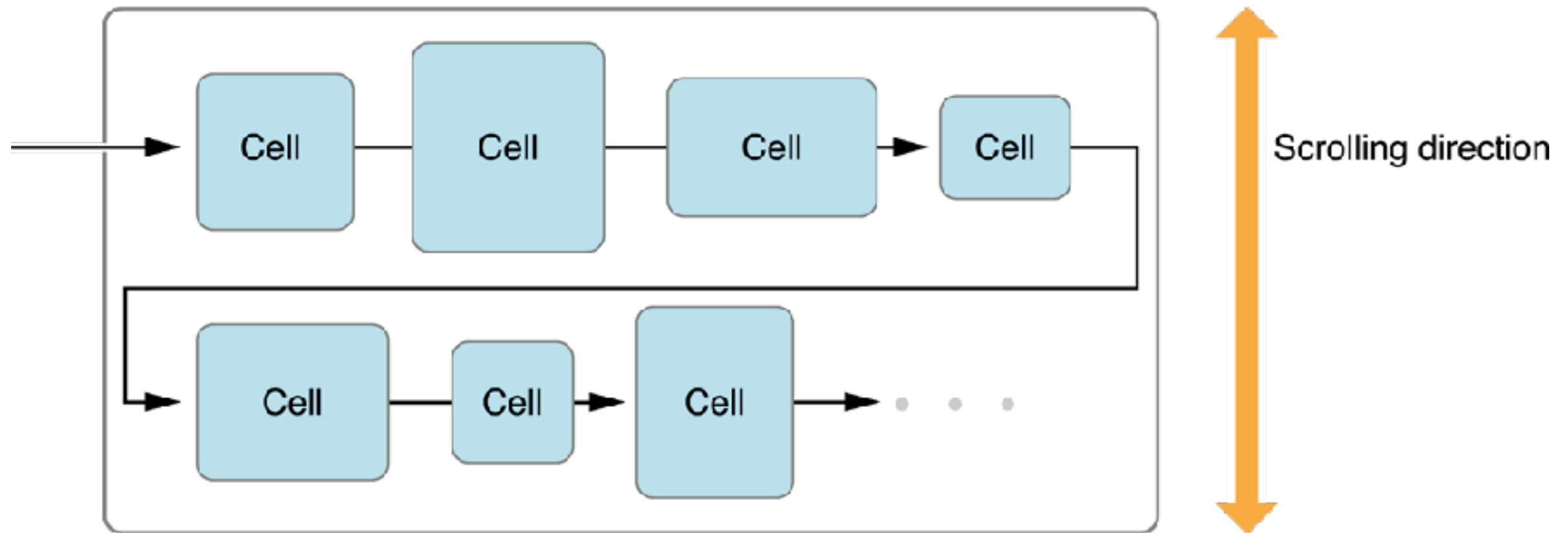
```
func collectionView(_ collectionView: UICollectionView, layout  
collectionViewLayout: UICollectionViewLayout, sizeForItemAt indexPath:  
IndexPath) -> CGSize
```

```
func collectionView(_ collectionView: UICollectionView, layout  
collectionViewLayout: UICollectionViewLayout,  
minimumInteritemSpacingForSectionAt section: Int) -> CGFloat
```

```
func collectionView(_ collectionView: UICollectionView, layout  
collectionViewLayout: UICollectionViewLayout,  
minimumLineSpacingForSectionAt section: Int) -> CGFloat
```

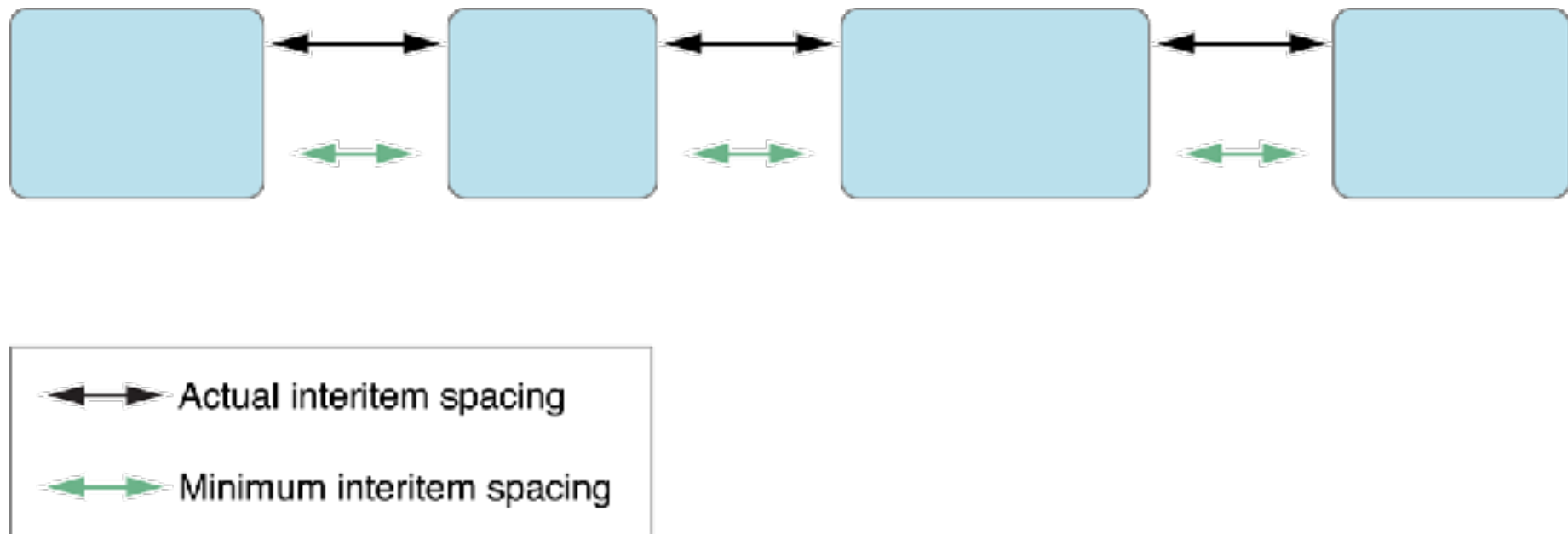
```
func collectionView(_ collectionView: UICollectionView, layout  
collectionViewLayout: UICollectionViewLayout, insetForSectionAt  
section: Int) -> UIEdgeInsets
```

Item Size



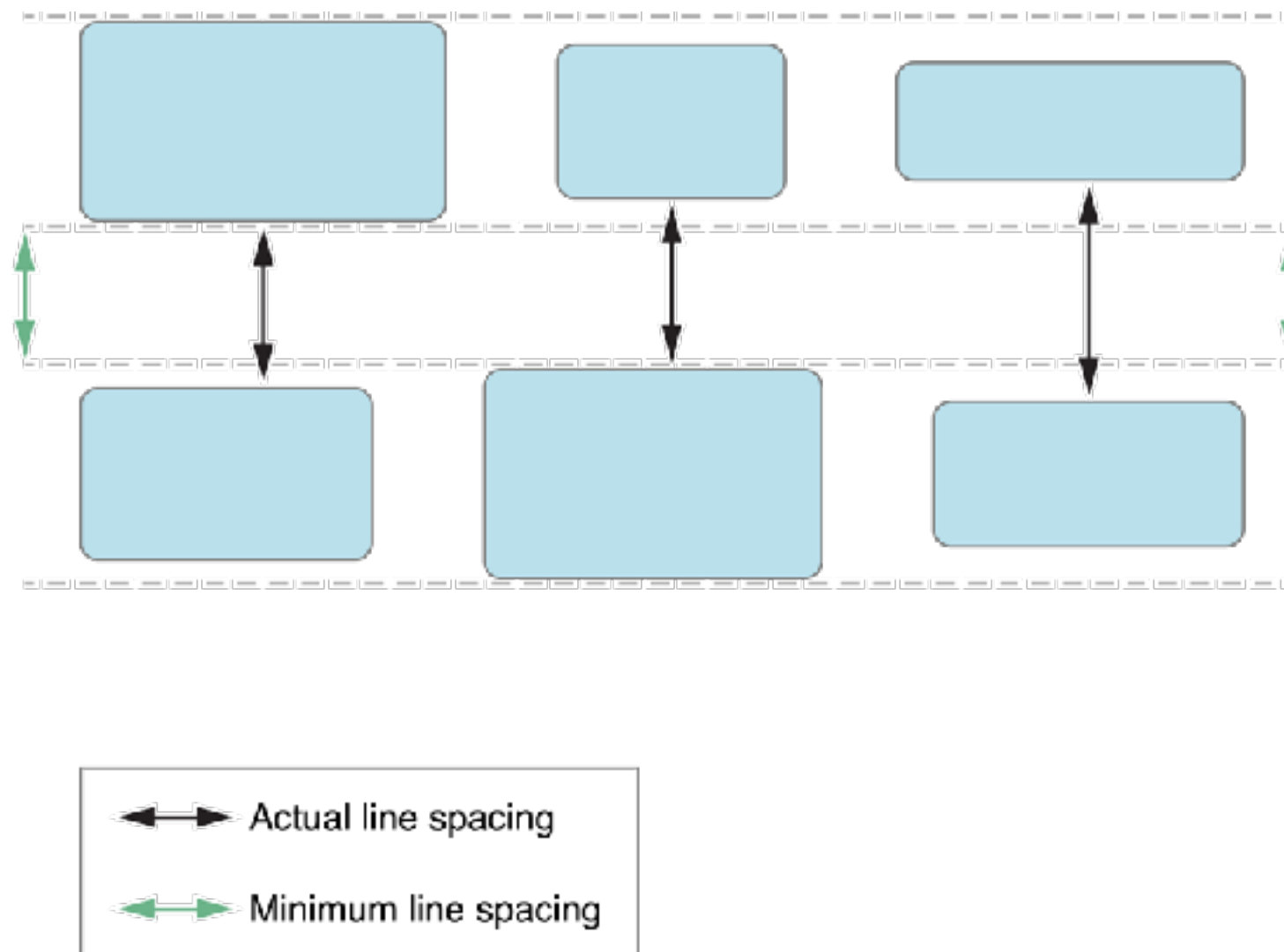
```
func collectionView(_ collectionView: UICollectionView,  
    layout collectionViewLayout: UICollectionViewLayout,  
    sizeForItemAt indexPath: IndexPath) -> CGSize
```

Item Spacing



```
func collectionView(_ collectionView: UICollectionView,  
    layout collectionViewLayout: UICollectionViewLayout,  
    minimumInteritemSpacingForSectionAt section: Int) -> CGFloat
```

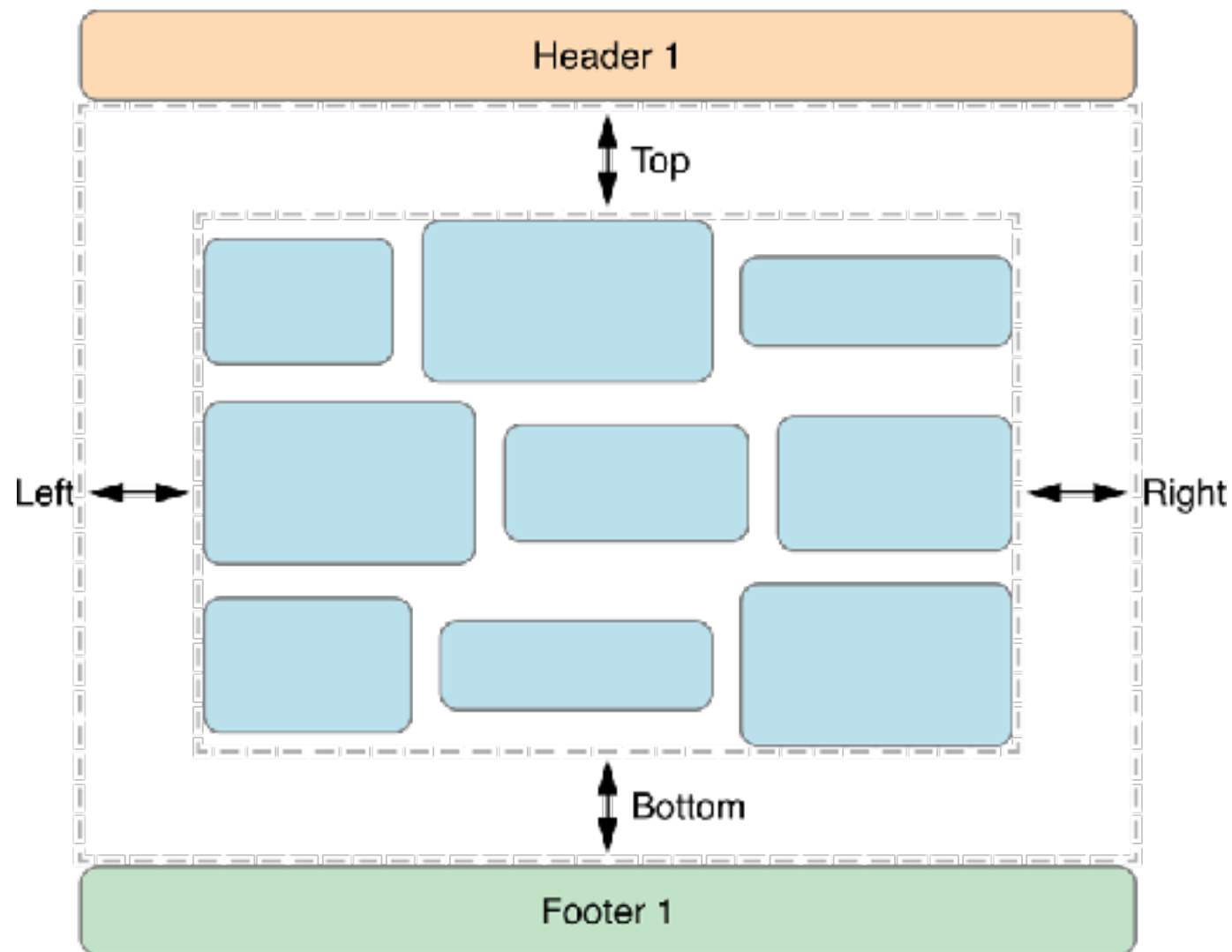
Line Spacing



```
func collectionView(_ collectionView: UICollectionView,  
    layout collectionViewLayout: UICollectionViewLayout,  
    minimumLineSpacingForSectionAt section: Int) -> CGFloat
```

Section Inset

```
inset = UIEdgeInsetsMake(top, left, bottom, right)
```



```
func collectionView(_ collectionView: UICollectionView,  
    layout collectionViewLayout: UICollectionViewLayout,  
    insetForSectionAt section: Int) -> UIEdgeInsets
```

Protocol

- UICollectionViewDataSource
- UICollectionViewDelegate
- UICollectionViewDataSourcePrefetching

UICollectionViewDataSource

```
public func collectionView(_ collectionView: UICollectionView,  
numberOfItemsInSection section: Int) -> Int
```

```
public func collectionView(_ collectionView: UICollectionView,  
cellForItemAt indexPath: IndexPath) -> UICollectionViewCell
```

```
optional public func numberOfSections(in collectionView: UICollectionView)  
-> Int
```

컬렉션 뷰 만들기

- 같이 해봐요

UICollectionViewDelegate

```
optional public func collectionView(_ collectionView: UICollectionView,  
shouldHighlightItemAt indexPath: IndexPath) -> Bool
```

```
optional public func collectionView(_ collectionView: UICollectionView,  
didHighlightItemAt indexPath: IndexPath)
```

```
optional public func collectionView(_ collectionView: UICollectionView,  
didUnhighlightItemAt indexPath: IndexPath)
```

```
optional public func collectionView(_ collectionView: UICollectionView,  
shouldSelectItemAt indexPath: IndexPath) -> Bool
```

```
optional public func collectionView(_ collectionView: UICollectionView,  
shouldDeselectItemAt indexPath: IndexPath) -> Bool // called when the user  
taps on an already-selected item in multi-select mode
```

```
optional public func collectionView(_ collectionView: UICollectionView,  
didSelectItemAt indexPath: IndexPath)
```

```
optional public func collectionView(_ collectionView: UICollectionView,  
didDeselectItemAt indexPath: IndexPath)
```

Multy-Selection 예제

```
@property (nonatomic) BOOL allowsSelection; // default is YES
@property (nonatomic) BOOL allowsMultipleSelection; // default is NO
```

- (BOOL)collectionView:(UICollectionView *)collectionView
shouldSelectItemAtIndexPath:(NSIndexPath *)indexPath;
- (BOOL)collectionView:(UICollectionView *)collectionView
shouldDeselectItemAtIndexPath:(NSIndexPath *)indexPath; // called when
the user taps on an already-selected item in multi-select mode

삽입, 삭제, 이동

```
func performBatchUpdates(_ updates: (() -> Swift.Void)?, completion: ((Bool) -> Swift.Void)? = nil)
```

- performBatchUpdates메소드의 update클로저를 사용해서 Section이나 Item의 업데이트 행동을 실행.
- 두가지 스텝을 필수로 진행해야한다.
 1. 데이터를 업데이트
 2. CollectionView의 해당 메소드를 호출하여 UI업데이트

Delete예제

```
func deleteSelectedItem()  
{  
    self.collectionView?.performBatchUpdates({  
        if let selectedItem = self.collectionView?.indexPathsForSelectedItems  
        {  
            //데이터 소스에서 항목 삭제  
  
            //뷰에서 제거  
            self.collectionView?.deleteItems(at: selectedItem!)  
        }  
    }, completion: { (isSuccess) in  
        //완료후 행동  
    })  
}
```

추가 같이 해보기

- 같이 해봐요