Collection Views

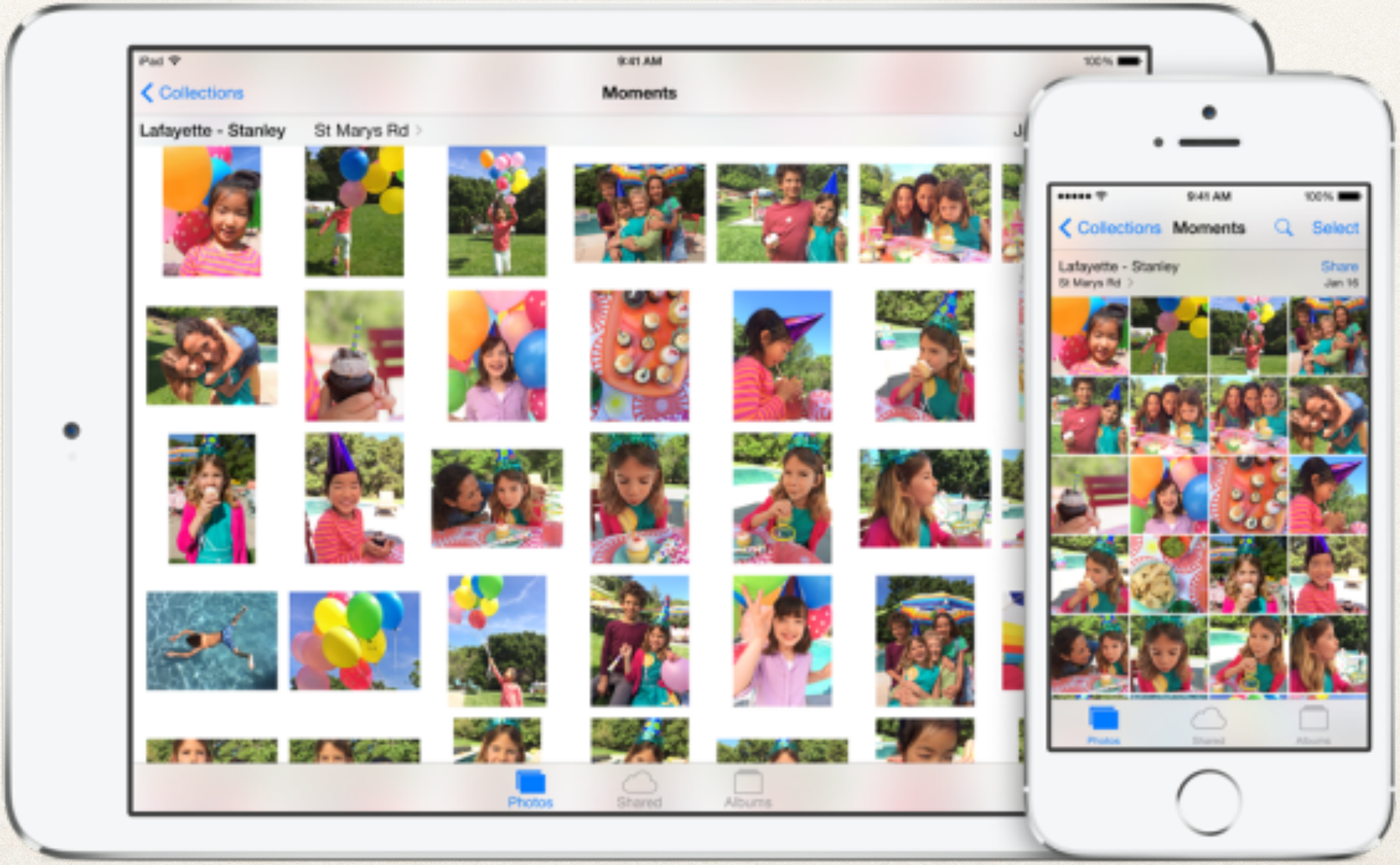
Dr. Sarah Abraham

University of Texas at Austin
CS329e
Spring 2020

Collection View

- Presents an ordered set of data items in a flexible layout
- Subclass of UIScrollView (like UITableView)
- Generalized notion of a table view
  - Uses view cells
  - Displays from a data source
  - Not limited to tabular rows

Collection View Example



Collection View Controller

- `UICollectionViewController` is an available class (same idea as `UITableViewController`)

- Also possible to hook `UICollectionView` into a standard `UIViewController` (more flexibility in presentation)

    - Create `IBOutlet` for collection view in view controller source code

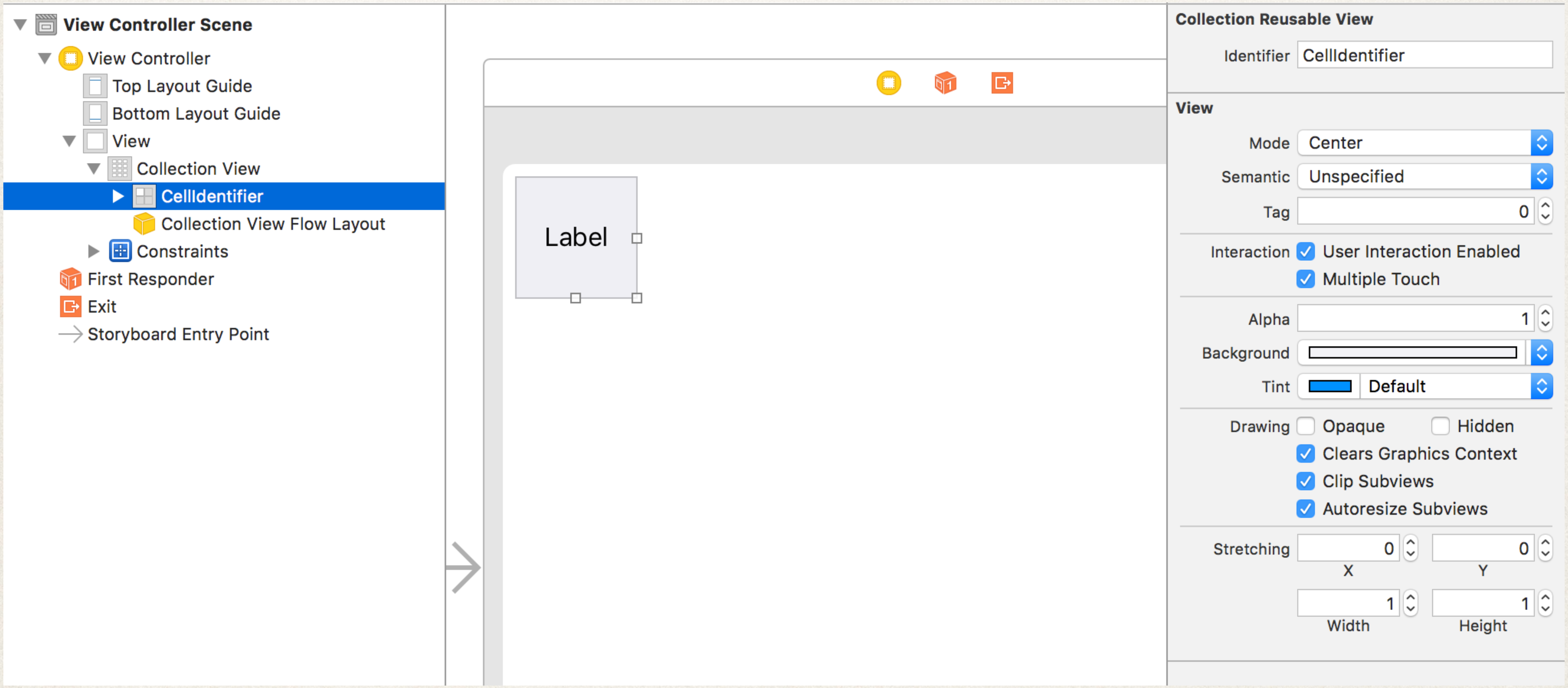
    - Connect outlet to view controller in Interface Builder

    - Connect this collection view's data source and delegate to parent view controller

Collection View in Xcode

```
class ViewController: UIViewController, UICollectionViewDataSource,
UICollectionViewDelegate, UICollectionViewDelegateFlowLayout {

    @IBOutlet weak var collectionView: UICollectionView!

    /* String identifier used to dynamically load cells into collection */
    let identifier = “CellIdentifier”

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from
a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```

Collection View in IB



Collection Reusable View

Identifier: CellIdentifier

View

Mode: Center

Semantic: Unspecified

Tag: 0

Interaction: ☑ User Interaction Enabled ☑ Multiple Touch

Alpha: 1

Background: □

Tint: 🟦 Default

Drawing: □ Opaque □ Hidden ☑ Clears Graphics Context ☑ Clip Subviews ☑ Autoresize Subviews

Stretching: X 0, Y 0, Width 1, Height 1

UICollectionViewDataSource

- Delegate to feed content into collection view
- Cell must have matching identifier in both Xcode and IB
- Returns information about items in the collection
- Required methods for implementation:
  - `collectionView(_: numberOfItemsInSection:)`
  - `collectionView(_: cellForItemAtIndexPath:)`

Data Source Protocol

- `func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int`
  - Data source provides collection with number of items in given `section`
- `func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell`
  - Data source provides collection with the cell at `indexPath`

Other Functions

- `numberOfSections(inSection:)`
  - Data source provides total number of sections in the collection view
- `collectionView(_:viewForSupplementaryElementOfKind:atIndexPath :)`
  - Data source provides a supplementary view to display in the collection view
  - Used for adding header and footer content
- `collectionView(_:moveItemAt:to:)`

UICollectionViewDelegate

- Handles events that interact with the cells
  - Cell selection
  - Cell highlighting
  - Addition and removal of views
  - Layout changes
  - Cell actions
  - Collection view focus

Collection View Layout

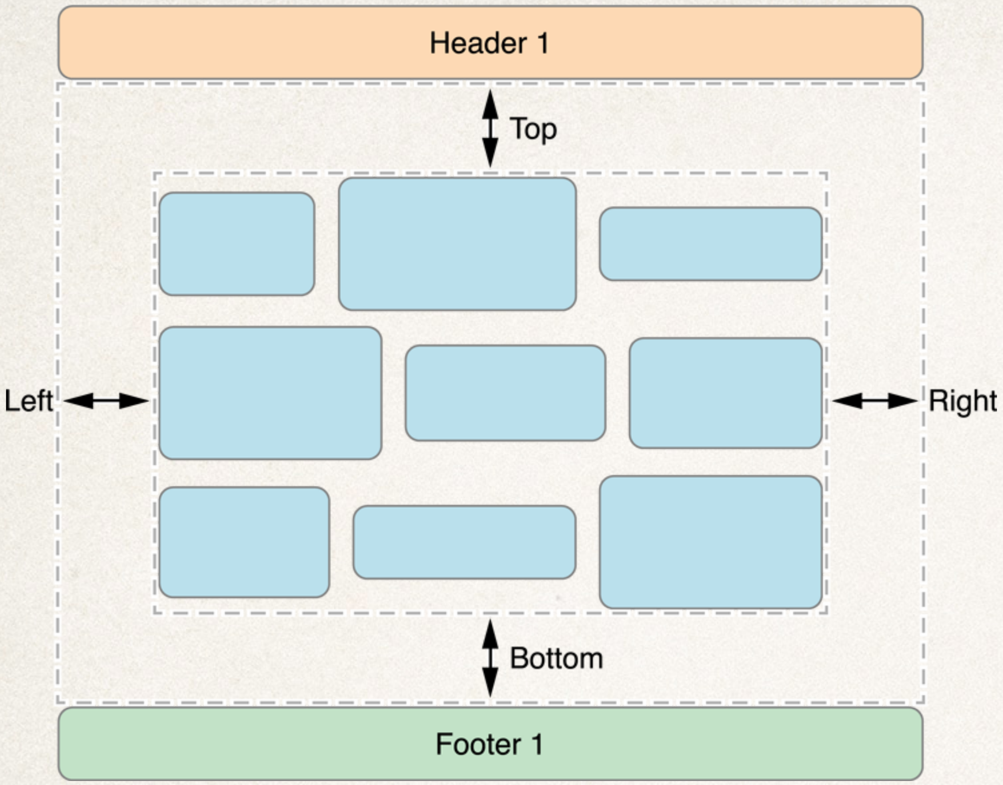
- `UICollectionViewFlowLayout`
- Collection views not restricted to tabular format
- Grid is default but can be modified for great deal of flexibility
- Contains delegate protocol: `UICollectionViewDelegateFlowLayout`
  - Defines layout appearance
  - Cell-spacing, scroll direction etc

Flow Layout Delegate Methods

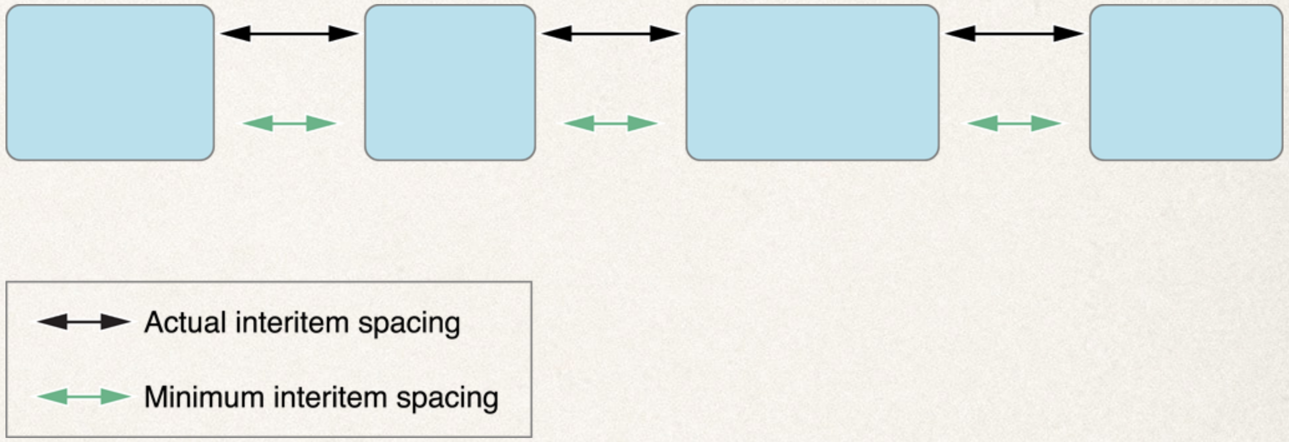
- All optional
- Set dimensions of specified cell
- Set margins around a section
- Set margins between items (rows and columns)
- Set size of header and footer around section

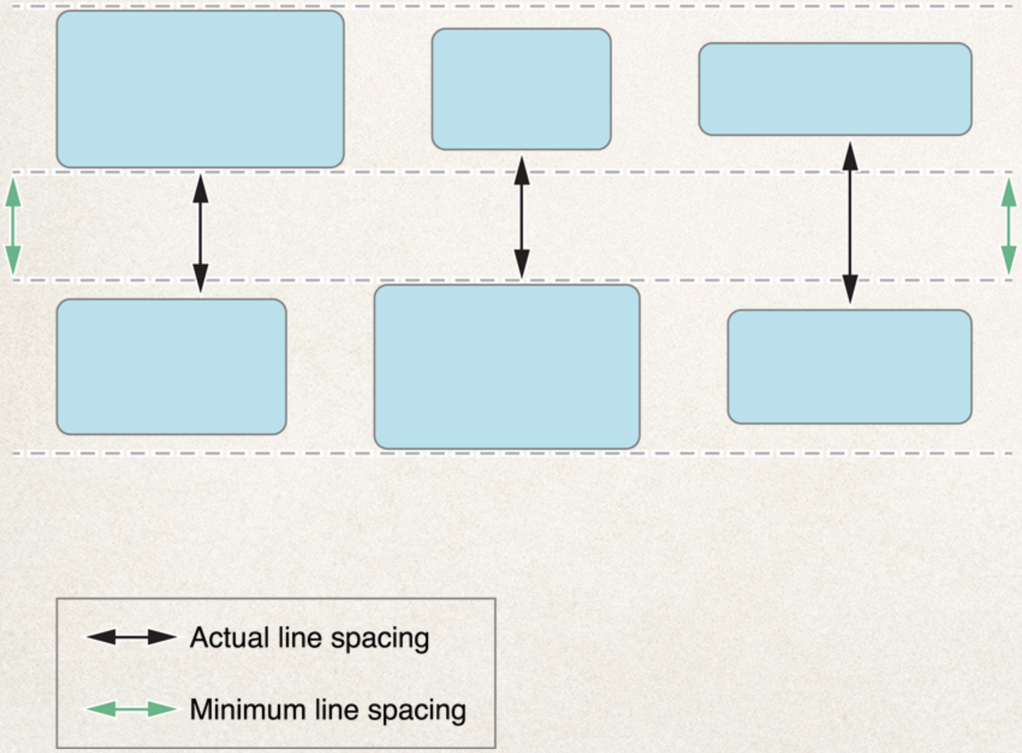
```
inset = UIEdgeInsetsMake(top, left, bottom, right)
```

```
func collectionView(UICollectionView,
    layout: UICollectionViewLayout,
     insetForSectionAtIndex: Int)
```



```
func collectionView(UICollectionView,
    layout: UICollectionViewLayout,
minimumInteritemSpacingForSectionAt:
             Int)
```





```
func collectionView(UICollectionView,
    layout: UICollectionViewLayout,
minimumLineSpacingForSectionAt: Int)
```

CGSize and UIEdgeInsets

- CGSize:
  - Part of CGGeometry, creates 2D geometric primitives
  - Structure that contains width and height
  - Width and height are unitless (not pixels)
- UIEdgeInsets:
  - Part of Views for setting areas and positions of interior views
  - Inset distance applied to rectangles to expand or shrink view
  - Can modify the top, bottom, left, and right separately

UICollectionViewFlowLayout

```
     //MARK: UICollectionViewFlowLayout

    func collectionView(_ collectionView: UICollectionView,
layout collectionViewLayout: UICollectionViewLayout,
sizeForItemAt indexPath: IndexPath) -> CGSize {
        return CGSize(width: 60, height: 60)
    }

    func collectionView(_ collectionView: UICollectionView,
layout collectionViewLayout: UICollectionViewLayout,
insetForSectionAt section: Int) -> UIEdgeInsets {
        return UIEdgeInsets.init(top: 0, left: 0, bottom:
0, right: 0)
    }
```

Collection View Cells

- UICollectionViewCell displays content
- Added as subviews to collection view
- Subclass to make cell appearance customizable
  - Create in Xcode
  - Customize in Interface Builder
- Must register cell with `reuseIdentifier`

```
    func collectionView(collectionView:
UICollectionView, cellForItemAtIndexPath indexPath:
NSIndexPath) -> UICollectionViewCell {

        let cell =
collectionView.dequeueReusableCellWithReuseIdentifier
(identifier, forIndexPath: indexPath) as!
UICollectionViewCell

        return cell

    }
```

Quiz Question: Collection Views

- Which class/delegate allows for UICollectionView's display flexibility?
    - UICollectionViewDataSource
    - UICollectionViewDelegate
    - UICollectionViewFlowLayout
    - UICollectionViewCell