

**CUSTOM  
COLLECTION VIEW  
LAYOUTS**

**WHO HAS EVER BUILT A  
CUSTOM LAYOUT?**

**THAT DOES NOT INHERIT  
FROM FLOW LAYOUT?**

**CUSTOM LAYOUT**  
**VS**  
**INHERITING FROM FLOW**  
**LAYOUT**

# **WHENEVER YOU CAN USE FLOW LAYOUT**

**Flow layout is a line breaking layout.**

**Any variation that lays items in a line  
and breaks when it gets to an end can be covered with flow layout.**

# WHENEVER YOU CAN USE FLOW LAYOUT

- ▶ Add new views
- ▶ Tweak layout attributes
- ▶ New layout attributes
- ▶ Gesture support
- ▶ Custom insert/delete animations

**FLOW LAYOUT IS HIGHLY OPTIMIZED AND  
VERSATILE TOOL**

**YOU SHOULD USE IT**



**WHY WOULD I WANT TO BUILD  
A CUSTOM LAYOUT?**

**WHAT ARE THE TYPES OF  
LAYOUTS?**

**LAYOUT DRIVEN**

**LAYOUT DRIVEN**  
**THINK FLOW LAYOUT**

**DATA DRIVEN**

**DATA DRIVEN  
THINK CALENDAR LAYOUT**

# UICollectionViewLayout Attributes

**DEFINES HOW ITEMS ARE LAID OUT BY YOUR  
COLLECTION VIEW.**



```
NS_CLASS_AVAILABLE_IOS(6_0) @interface UICollectionViewLayoutAttributes : NSObject

@property (nonatomic) CGRect frame;
@property (nonatomic) CGPoint center;
@property (nonatomic) CGSize size;
@property (nonatomic) CATransform3D transform3D;
@property (nonatomic) CGRect bounds NS_AVAILABLE_IOS(7_0);
@property (nonatomic) CGAffineTransform transform NS_AVAILABLE_IOS(7_0);
@property (nonatomic) CGFloat alpha;
@property (nonatomic) NSInteger zIndex; // default is 0
@property (nonatomic, getter=isVisible) BOOL hidden;
@property (nonatomic, retain) NSIndexPath *indexPath;

@property (nonatomic, readonly) UICollectionViewCellCategory representedElementCategory;
@property (nonatomic, readonly) NSString *representedElementKind;
```

**YOU CAN PROVIDE YOUR OWN SUBCLASS TO  
PASS ADDITIONAL INFORMATION TO CELLS,  
SUPPLEMENTARY AND DECORATION VIEWS.**

```
NS_CLASS_AVAILABLE_IOS(6_0) @interface UICollectionViewReusableView : UIView

(...)

- (void)applyLayoutAttributes:(UICollectionViewLayoutAttributes *)attributes;

(...)

@end
```

**YOU HAVE TO OVERRIDE**  
`isEqual` & `copy`

# **BUILDING CUSTOM COLLECTION VIEW LAYOUT**

# HOW DOES UICOLLECTIONVIEW PREPARE ITS LAYOUT?

- (void)prepareLayout;
- (CGSize)collectionViewContentSize;
- (NSArray \*)layoutAttributesForElementsInRect:(CGRect)rect;

# PREPARE LAYOUT

```
- (void)prepareLayout {  
    [super prepareLayout];  
  
    (...)  
}
```

# CONTENT SIZE

```
- (CGSize)collectionViewContentSize {  
    CGSize myCustomSize = (...)  
    return myCustomSize;  
}
```



# LAYOUT ATTRIBUTES FOR ELEMENTS IN RECT

```
- (NSArray *)layoutAttributesForElementsInRect:(CGRect)rect {  
    return [super layoutAttributesForElementsInRect:rect];  
}
```

**The `rect` parameter does not define whole bounds of collection view.**

**UICOLLECTIONVIEW CONSIDERS ITS LAYOUT  
UP AND READY AFTER THESE THREE  
METHODS ARE CALLED.**

# LAYOUT ATTRIBUTES FOR ELEMENT AT INDEX PATH

```
- (UICollectionViewLayoutAttributes *)layoutAttributesForItemAtIndexPath:(NSIndexPath *)path {  
    return [super layoutAttributesForItemAtIndexPath:indexPath];  
}
```

**INVALIDATING**

**LAYOUT**

**WHEN?**

# INVALIDATE WHEN A VALUE THAT DRIVES LAYOUT CHANGES

- ▶ Bounds
- ▶ Content offset
- ▶ Data (data driven layouts)
  - ▶ Layout-specific

# INVALIDATING LAYOUT

```
[layout invalidateLayout];
```

# INVALIDATING LAYOUT

```
- (BOOL)shouldInvalidateLayoutForBoundsChange:(CGRect)newBounds {  
    return [super shouldInvalidateLayoutForBoundsChange:newBounds];  
}
```



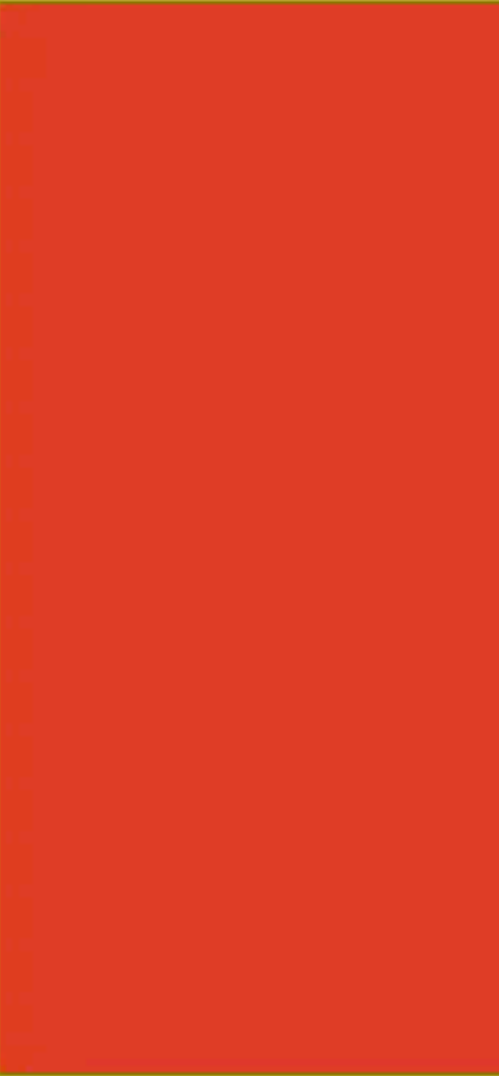
# ASSIGNMENT 1

## CAROUSEL LAYOUT

# GOALS

- ▶ Behaves like paging in scroll view with horizontal scrolling
  - ▶ Item is always centered vertically when scroll rests
- ▶ When iPad is rotated the same cell stays in the middle of the screen
  - ▶ No code related to setting up layout in controller
- ▶ Does not inherit from UICollectionViewFlowLayout (optional)

**EXAMPLE**



# TIPS

# USE TARGET CONTENT OFFSET WITH VELOCITY

- `(CGPoint)targetContentOffsetForProposedContentOffset:(CGPoint)proposedContentOffset  
withScrollingVelocity:(CGPoint)velocity;`

# AND TARGET CONTENT OFFSET

- `(CGPoint)targetContentOffsetForProposedContentOffset:(CGPoint)proposedContentOffset;`





**LET'S CODE!**

# ASSIGNMENT 2

CALENDAR LAYOUT

LET'S TRY SOMETHING HARDER

# GOALS

- ▶ **Displays items of different sizes that represent events**
  - ▶ **Displays a 'bead' view that shows current time**
    - ▶ **Does not inherit from flow layout**

**EXAMPLE**

Carrier 

2:48 PM



# GETTING DATA TO YOUR COLLECTION VIEW LAYOUT

# THE EXTENSIBLE DELEGATE PATTERN





**WHAT DATA WILL YOU NEED?**

**TIME TO POSITION BEAD VIEW**

**EVENTS FOR CALENDAR**

**START OF DISPLAYED DAY**

**END OF DISPLAYED DATE**

**BEAD VIEW**  
**CONSIDER IT AS**  
**A DECORATION VIEW**

# REGISTERING A DECORATION VIEW

```
[self registerClass:[BeadView class]  
forDecorationViewOfKind:DecorationKindBead];
```

**PROVIDING DECORATION  
LAYOUT ATTRIBUTES  
SAME AS WITH CELL LAYOUT ATTRIBUTES**



```
- (UICollectionViewLayoutAttributes *)layoutAttributesForDecorationViewOfKind:(NSString *)kind  
atIndexPath:(NSIndexPath *)indexPath
```

```
[UICollectionViewLayoutAttributes layoutAttributesForDecorationViewOfKind:DecorationKindBead  
withIndexPath:indexPath];
```

**HAVE TO ALSO BE PASSED IN  
LAYOUT ATTRIBUTES FOR  
RECT**

**1 MINUTE EQUALS 1 PIXEL**

```
[self.startOfDay mt_minutesUntilDate:self.endOfDay];
```

Carrier 

2:48 PM



**LET'S CODE!**

# ASSIGNMENT 3

CALENDAR LAYOUT


SEPARATORS



# GOALS

- ▶ **Display dark gray line at each hour**
- ▶ **Display light gray line at each half an hour**
- ▶ **Display hour next to each dark gray line**

**EXAMPLE**

Carrier 

3:14 PM



12:00 AM

1:00 AM

2:00 AM

3:00 AM

4:00 AM

5:00 AM

6:00 AM

7:00 AM

**TIPS**

**USE CUSTOM LAYOUT ATTRIBUTES TO GET  
COLOR/TEXT TO SEPARATOR VIEW**

# SEPARATORS

## DECORATION VIEWS

**LET'S CODE!**

# CONCLUSION

- ▶ **Flow layout can be easily modified to developer needs**
  - ▶ **Building custom layouts is not that hard**
- ▶ **Layout attributes can be a great way of providing additional data to collection view elements**



**THANK YOU FOR YOUR  
ATTENTION!**

**PAWEL DUDEK**

**@ELDUDI**

**[HTTPS://GITHUB.COM/PAWELDUDEK](https://github.com/paweldudek)**