
UICollectionView-Layout

삽입, 삭제, 이동

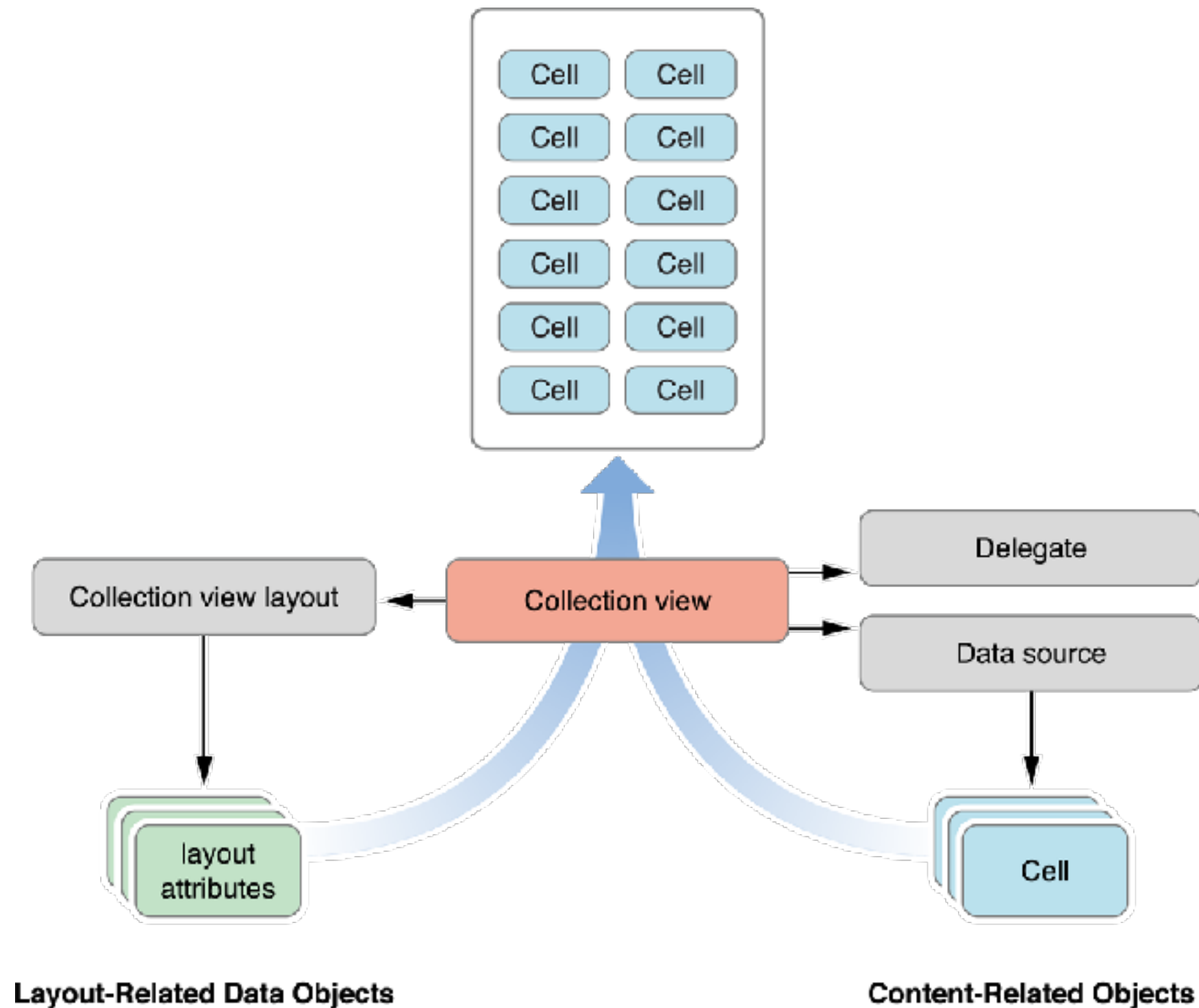
```
func performBatchUpdates(_ updates: (() -> Swift.Void)?, completion: ((Bool) -> Swift.Void)? = nil)
```

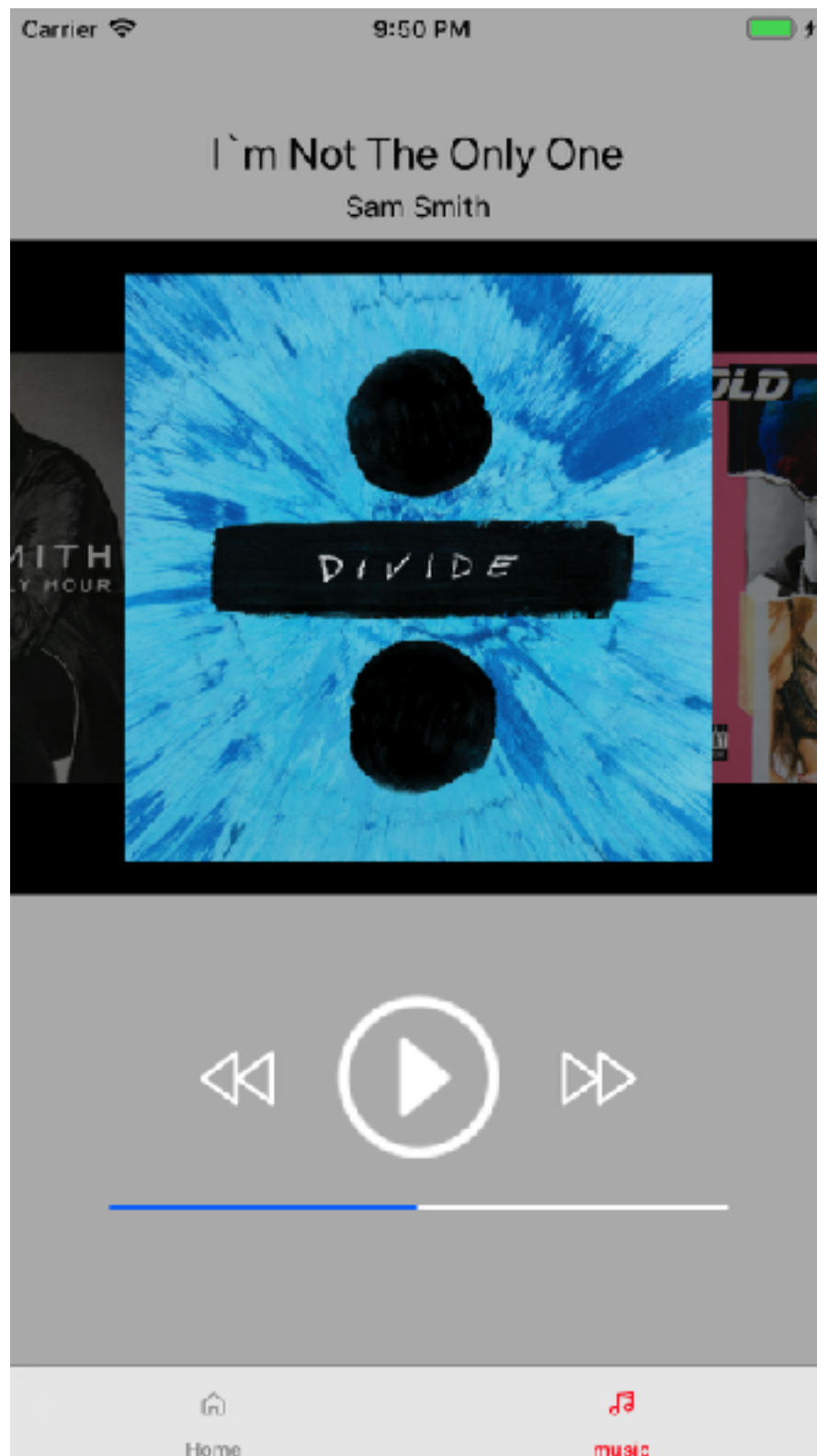
- performBatchUpdates메소드의 update클로저를 사용해서 Section이나 Item의 업데이트 행동을 실행.
- 두가지 스텝을 필수로 진행해야한다.
 1. 데이터를 업데이트
 2. CollectionView의 해당 메소드를 호출하여 UI업데이트

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  
    })  
}
```

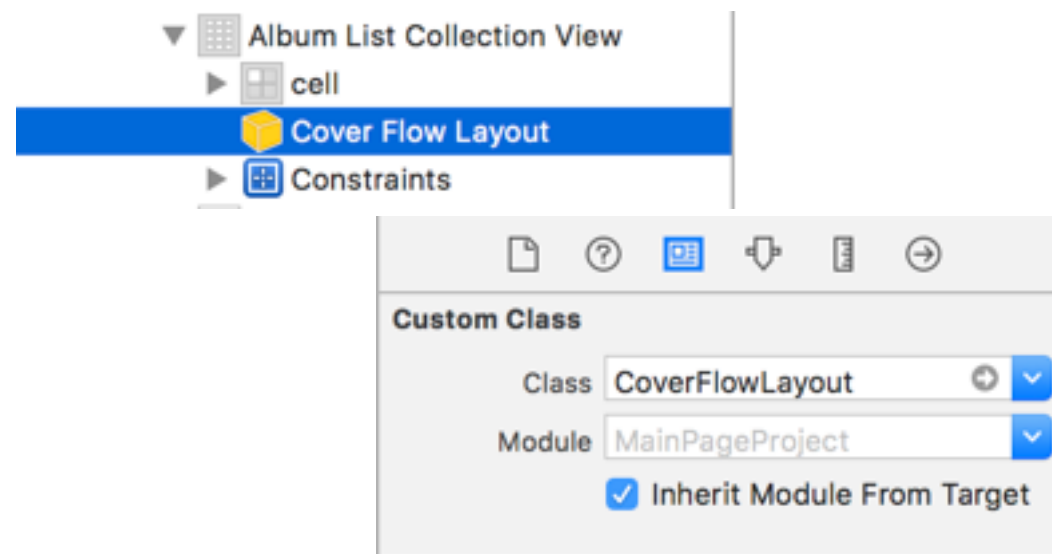
UICollectionView



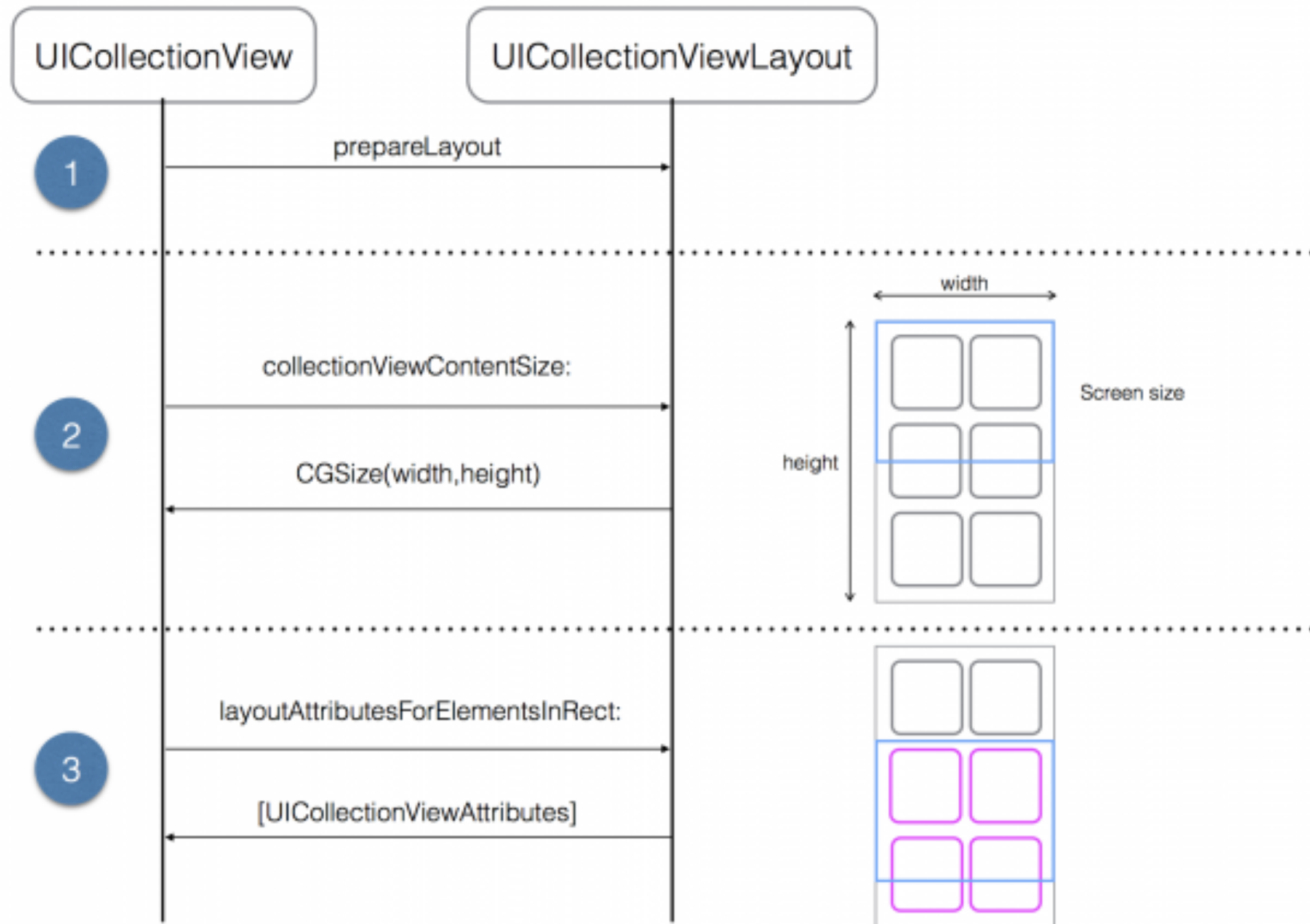


CoverFlow만들기

1. UICollectionViewFlowLayout 상속 받은 커스텀 layout 만들기
2. Layout CustomClass에 적용



Layout Process



CoverFlow

```
override fun layoutAttributesForElements(in rect: CGRect) ->
[UICollectionViewLayoutAttributes]? {

    guard let attributes = super.layoutAttributesForElements(in: rect) else
{return nil}

    var layoutAttribute:[UICollectionViewLayoutAttributes] = []
    for attribute in attributes
    {
        //change
        changeLayoutAttribute(attribute: attribute)
        //add
        layoutAttribute.append(attribute)
    }
    return layoutAttribute
}

//레이아웃 정보들 다시 불러오도록 허락함
override fun shouldInvalidateLayout(forBoundsChange newBounds: CGRect) ->
Bool {
    return true
}
```

CoverFlow

```
func distance(ofCenter centerx:CGFloat) -> CGFloat
{
    //컬렉션 뷰 센터 포지션
    let collectionViewCenter = (self.collectionView?.frame.size.width)! / 2.0
    //현재 아이템의 offsetX + 센터
    let offSet = (self.collectionView?.contentOffset.x)! + collectionViewCenter
    return fabs(offSet - centerx)
}

//실제 연산내용
func changeLayoutAttribute(attribute:UICollectionViewLayoutAttributes)
{
    //변경가능한 최대 거리(조정가능)
    let maxDistance = self.itemSize.width + self.minimumLineSpacing
    //실제 거리과 최대거리중 작은거리를 선택!
    let choiceDistance = min(distance(ofCenter: attribute.center.x), maxDistance)
    //비율
    let ratio = (maxDistance - choiceDistance) / maxDistance
    //비율에 따라 스케일과 투명도 변경
    let scale = ratio * (1 - self.itemScale) + 1.0
    let alpha = ratio * (1 - self.itemAlpha) + self.itemAlpha;
    attribute.alpha = alpha;
    attribute.transform3D = CATransform3DScale(CATransform3DIdentity, scale, scale, 1);
    //alpha값에 따른 z 좌표 변경
    attribute.zIndex = NSInteger(alpha * 10.0)
}
```


CoverFlow

```
//스크롤을 멈췄을때 실행 -> 컨텐츠 off
override func targetContentOffset(forProposedContentOffset proposedContentOffset:
CGPoint, withScrollingVelocity velocity: CGPoint) -> CGPoint {

    guard let collectionView = self.collectionView else {
        return proposedContentOffset
    }
    //현재 컬렉션 뷰의 [UICollectionViewLayoutAttributes] 가져오기
    guard let attributeList = self.layoutAttributesForElements(in:
collectionView.bounds) else {
        return proposedContentOffset
    }
    //거리가 가장 가까운 순으로 정렬
    let sortedAttributeList = attributeList.sorted {
        (attribute1, attribute2) -> Bool in
            distance(ofCenter: attribute1.center.x) < distance(ofCenter:
attribute2.center.x)
    }
    //가장 가까운 아이템의 센터 위치
    let xCenterOfMinimumAttributes = sortedAttributeList.first?.center.x
    //중앙으로 이동
    let screenXCenter = collectionView.frame.size.width / 2
    let targetContentOffset = CGPoint(x:xCenterOfMinimumAttributes! - screenXCenter,
                                     y: proposedContentOffset.y)

    return targetContentOffset
}
```

추가 학습

- 한번 해보세요

*핀터레스트 만들기

<https://www.raywenderlich.com/164608/uicollectionview-custom-layout-tutorial-pinterest-2>

*단축 주소

<https://goo.gl/NtLmRy>