

# **3.UIKit / android.widget / Layout**

# UIKit

- iOS화면의 기본이 되는 화면요소를 정의하는 Framework
- 해당 Kit를 이해하는 것이 Native프로그래밍의 99%임. 언어는 중요하지 않음.
- 접두어에서 컨트롤에 UI가 붙음

# UIKit

## - Objective-C

```
UIViewController* vc = [[UIViewController alloc] init];  
UILabel* label = [[UILabel alloc] init];  
UITextField* textField = [[UITextField alloc] init];  
[vc addSubview:label];  
[vc addSubview:textField];  
[self pushViewController:vc];
```

## -Swift

```
let vc = UIViewController()  
var label = UILabel()  
var textField = UITextField()  
vc.addSubview(label)  
vc.addSubview(textField)  
self.pushViewController(vc)
```

# android.widget

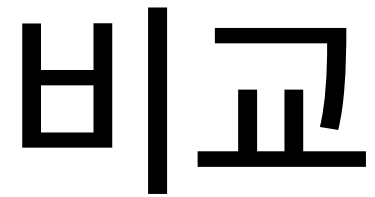
- android.widget.\* / android.view.\* 안드로이드 화면요소를 정의한 패키지
- 안드로이드 홈화면에서 사용하는 위젯과는 다름(패키지 이름을 위젯이 나오기 전에 지었기 때문에)
- compat이 들어가는 파생 패키지도 있으나 compat은 과거 호환을 위한 것으로 기본과 거의 차이가 없음.

# android.widget

```
Intent intent = new Intent(this, TestActivity.class);  
this.startActivity(intent);
```

TestActivity.java

```
onCreate(...) {  
    super.onCreate(...)  
    ViewGroup viewGroup = getContentView();  
    TextView label = new TextView(this);  
    EditText textField = new EditText(this);  
    viewGroup.addView(label)  
    viewGroup.addView(textField)  
}
```



UIViewController

Activity  
Fragment

UIView

android.view.ViewGroup  
android.view.View  
android.widget.FrameLayout  
android.widget.RelativeLayout  
android.widget.LinearLayout

UILabel

android.widget.TextView

UITextField  
UITextView

android.widget.EditText

UIButton

android.widget.Button

UIScrollView

android.widget.HorizontalScrollView  
android.widget.VerticalScrollView

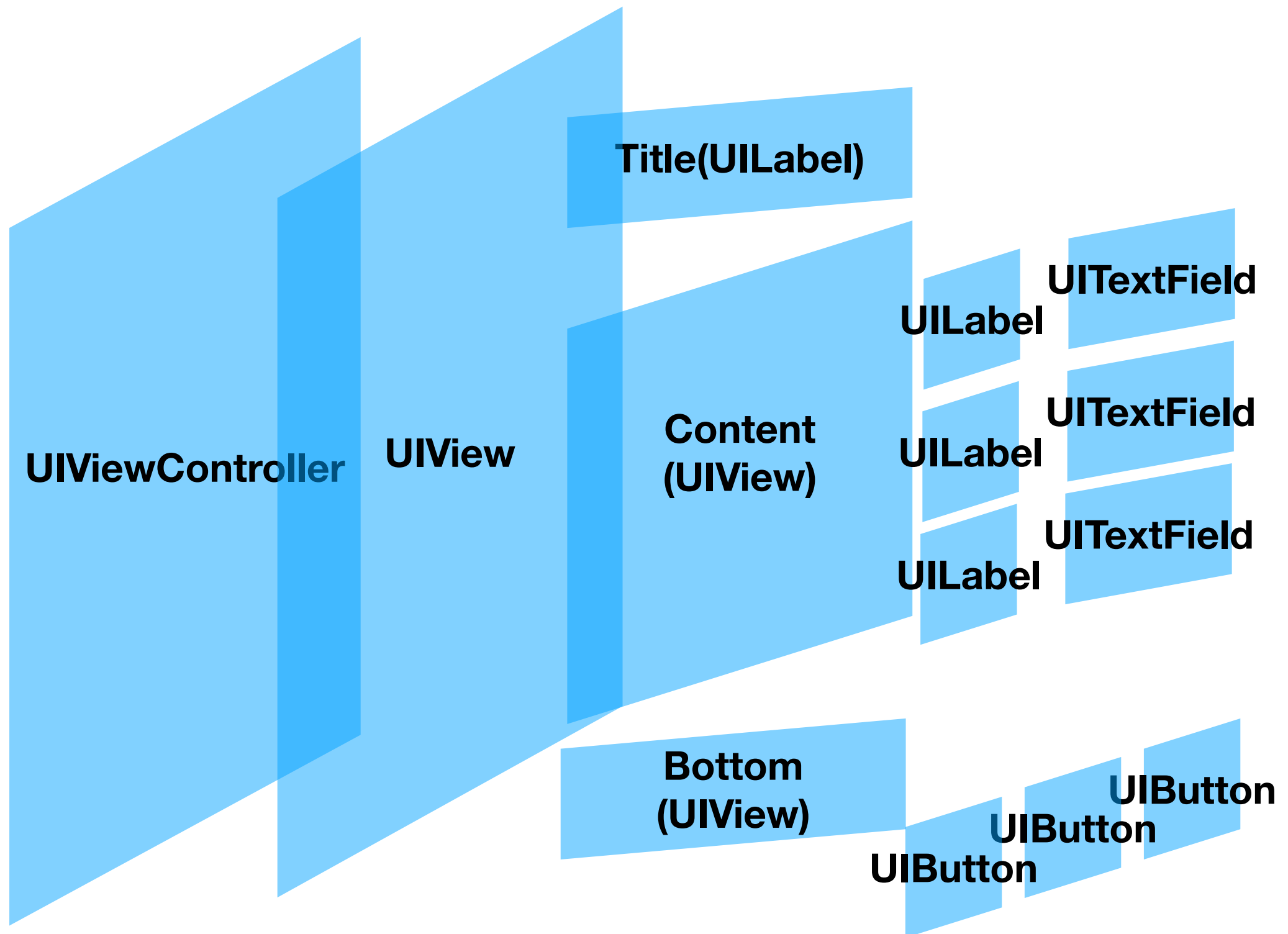
UITableView

android.widget.ListView

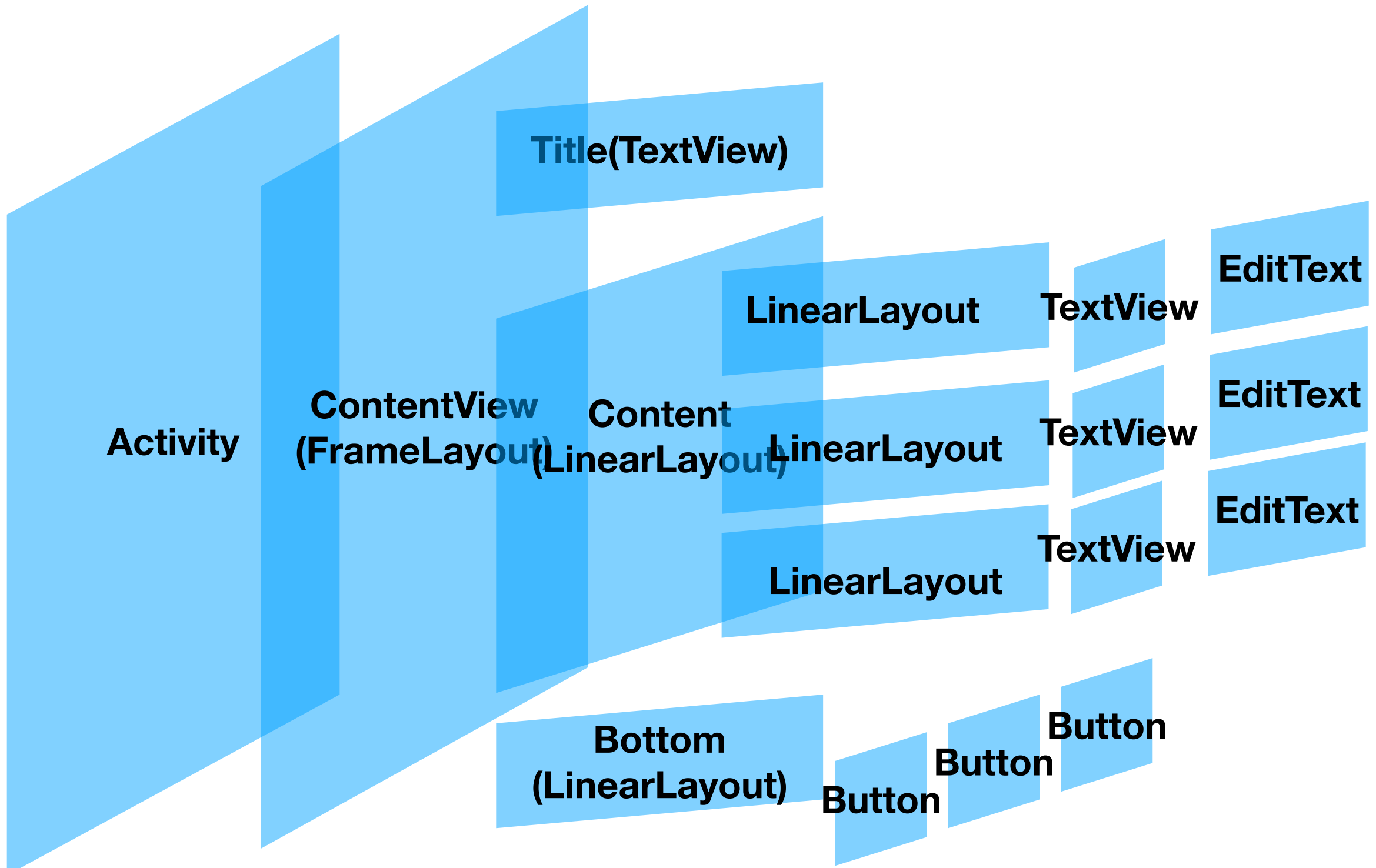
UICollectionView

androidx.recyclerview.widget.RecyclerView

# UIKit

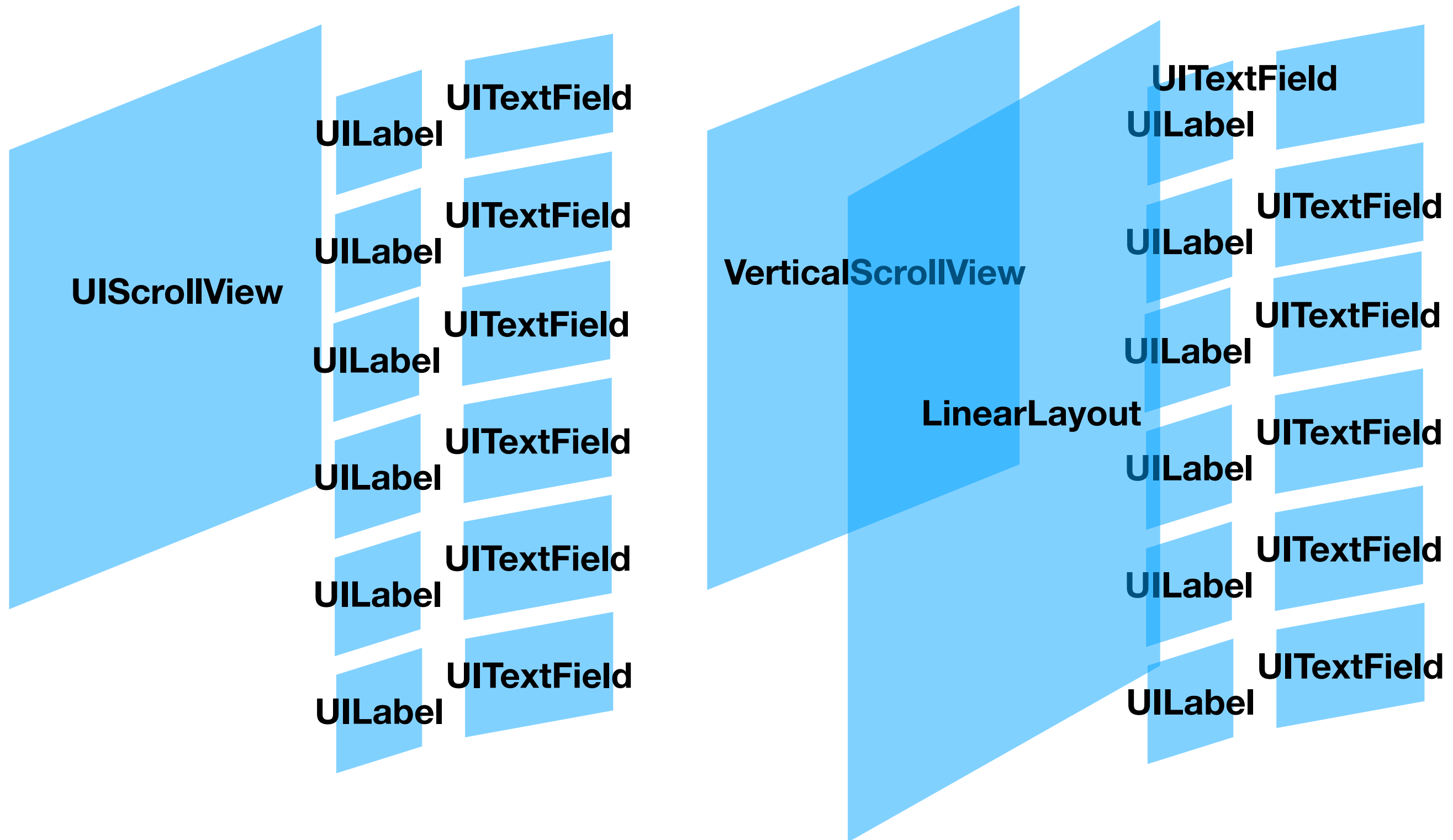


# android.widget

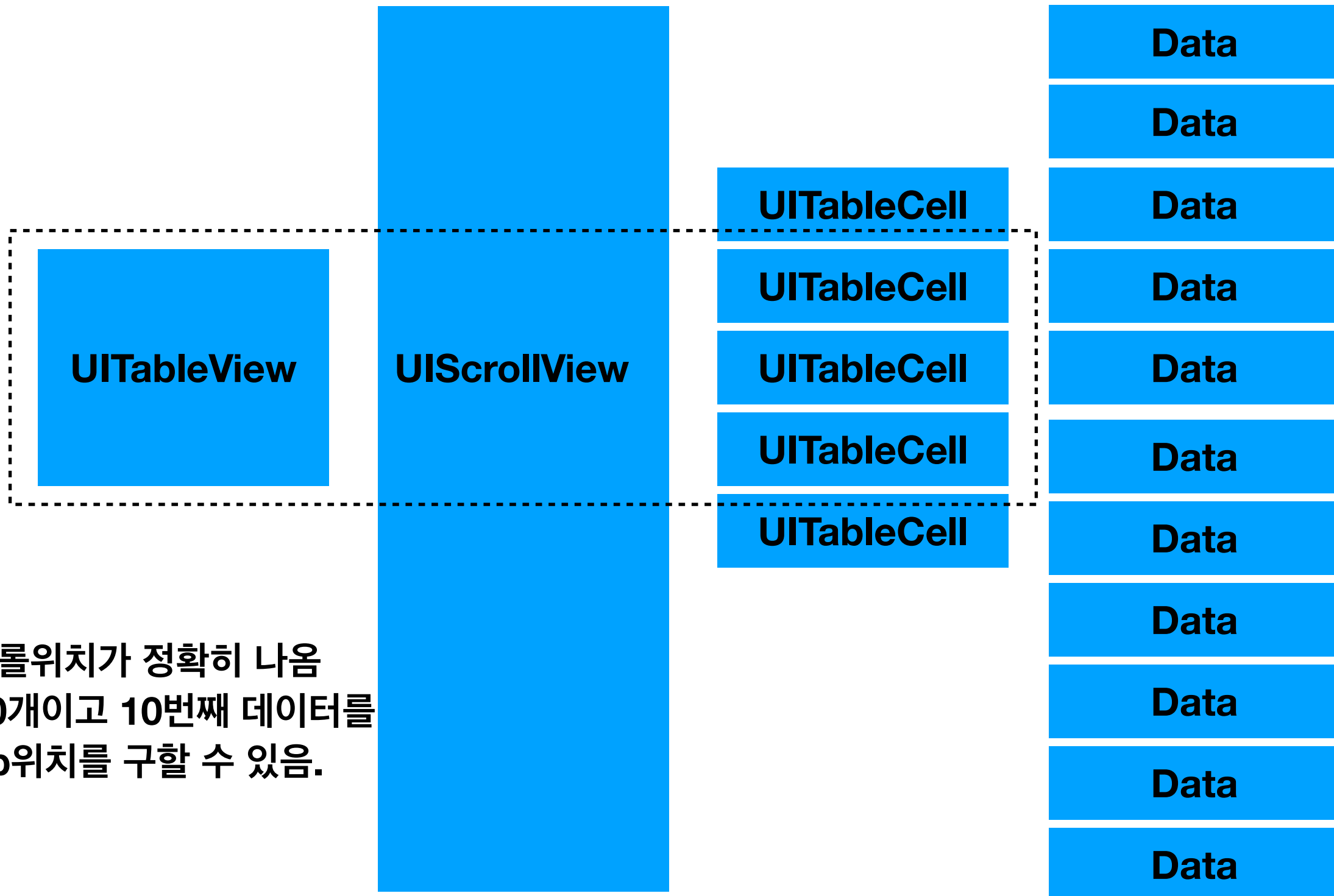




# ScrollView

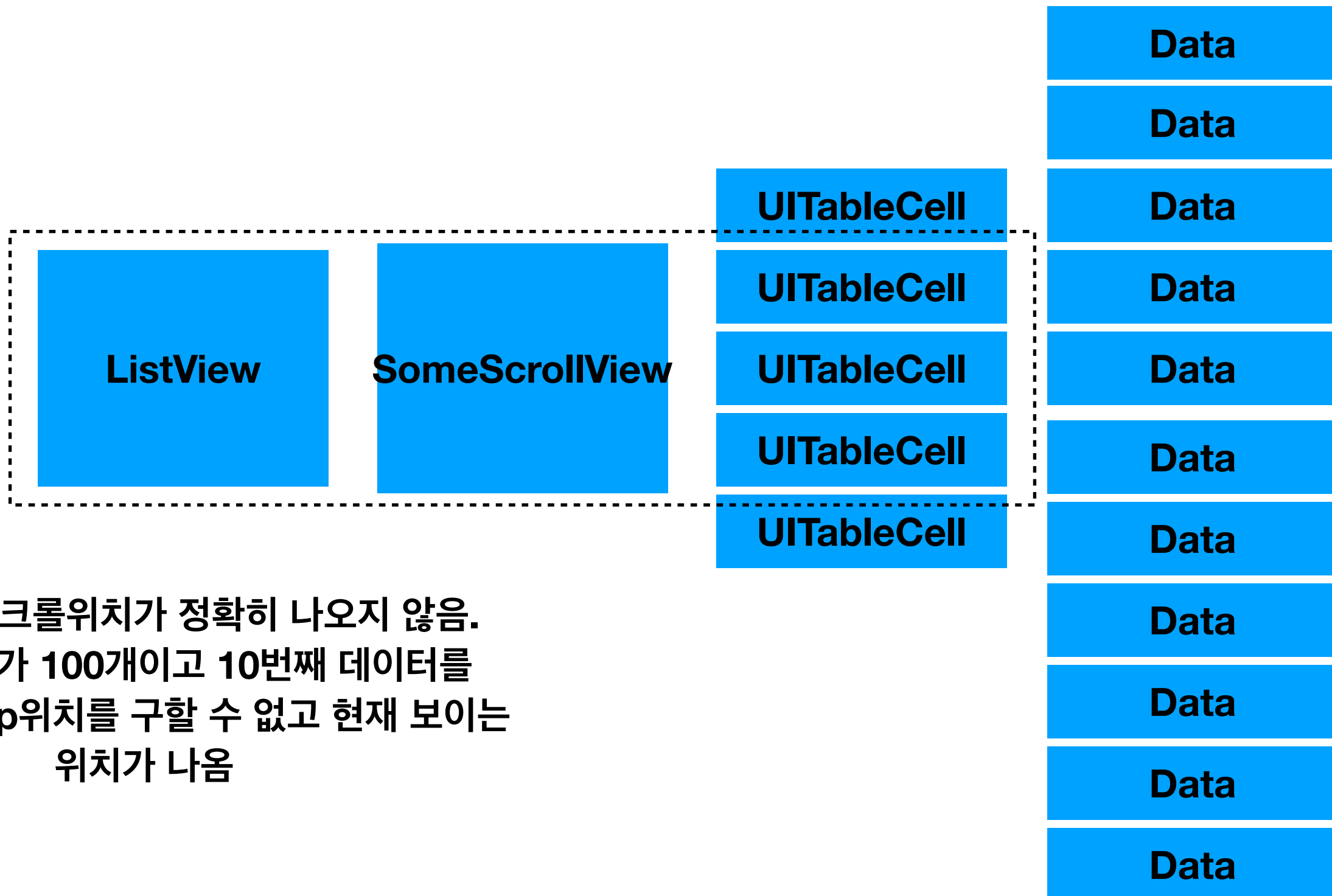


# UITableView



해당 셀의 스크롤위치가 정확히 나옴  
데이터 갯수가 100개이고 10번째 데이터를  
표시한셀의 top위치를 구할 수 있음.

# android.widget.ListView



# 다양한 해상도 대응

- Apple과 Google의 권고안
- 통상적인 해상도 통일 처리
- 반응형웹 페이지
- 과거 m.Stock과 현재 m.Stock
- interface Builder / Layout editor

# 화면 구성 기본개념.

- RelativeLayout개념(RelativeLayout은 한계가 있어서 적절히 바꾸어야 함.)
- iOS에서 하위객체 크기조정
  - 생성시 결정(viewDidLoad / viewWillAppear / init(생성자))
  - layoutSubviews / viewWillLayoutSubviews
  - Animation시 주의점
- Android에서 하위객체 크기조정
  - 생성자 / onCreate
  - onLayout / onLayout의 한계점
  - post / postDelay / iOS개념을 붙이기는 곤란