

● 안드로이드 뷰 객체의 계층구조는 소프트웨어 모델로 이용하자면 DOM(Document Object Model)을 따르고 있고, 패턴(Pattern)으로 이야기 하자면, Gof 디자인 패턴의 Composite 패턴이 적용된 구조이다. 뷰 객체 여러 개를 뷰그룹에 포함하는 식으로 뷰 클래스의 객체들을 계층구조로 묶어서 사용한다. 이렇게 완성된 계층의 최상위 뷰그룹을 화면에 출력하면 전체가 함께 출력된다.

▶ View: 안드로이드 뷰 클래스의 최상위 클래스로, 액티비티에 출력되는 클래스는 모두 이 View의 서브 클래스여야 한다. 최상위 클래스 명이 View이기 때문에, 안드로이드에서는 Button과 같은 UI 위젯 클래스를 뷰라고 부른다.

▶ ViewGroup: 뷰의 서브 클래스여서 화면에 출력되지만, 뷰그룹 자체만의 UI는 가지지 않는다. 뷰그룹의 역할은 UI출력이 아니라 다른 뷰 여러 개를 뷰그룹에 포함(Add)하여 한꺼번에 제어하기 위한 목적으로, 일반적으로 컨테이너(Container)기능을 담당한다. 실제 뷰그룹도 이용되지만, 뷰그룹의 서브 클래스인 레이아웃 클래스들이 사용된다.

1) 레이아웃 XML로 계층구조 구현: 태그의 상하 관계를 이용하면 된다. 서브 태그로 등록한 뷰 객체가 상위 태그 객체에 포함된다.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    >

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="버튼1"
        android:gravity="center"
        android:id="@+id/button1" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="버튼2"
        android:gravity="center"
        android:id="@+id/button2" />

</LinearLayout>
```

2) 자바 코드로 계층구조 구현: addView() 함수를 이용하면 된다.

```
package com.example.practicexml;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        LinearLayout linear = new LinearLayout(this);
        linear.setOrientation(LinearLayout.Vertical);

        Button button1 = new Button(this);
        button1.setText("버튼 1");
        button1.setGravity(Gravity.CENTER);
        linear.addView(button1);

        Button button2 = new Button(this);
        button2.setText("버튼 2");
        button2.setGravity(Gravity.CENTER);
        linear.addView(button2);

        setContentView(linear);
    }
}
```

Resources: 강쌤의 안드로이드 프로그래밍
