02-1. UI Architecture

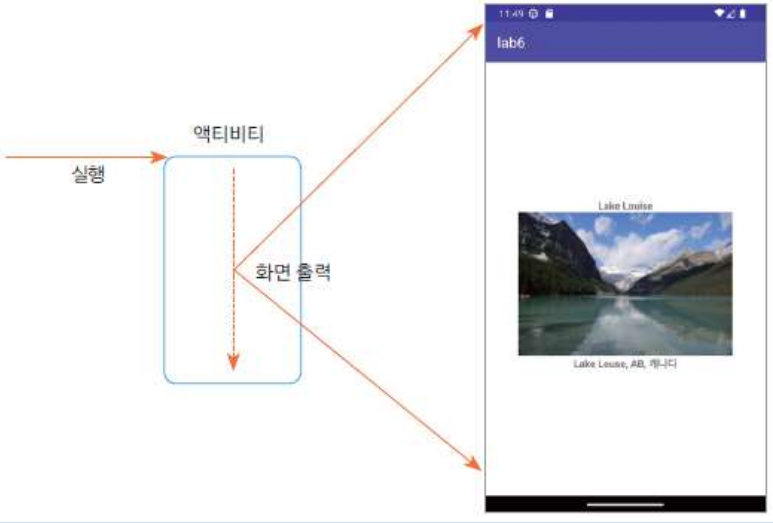
UI Programming

UI 기본 구조

액티비티-뷰 구조

- 액티비티 자체는 앱의 실행 단위인 컴포넌트
- 액티비티에 버튼, 문자열, 이미지 등을 출력
- 액티비티 내에서 화면 구성을 위한 적절한 뷰 클래스를 출력

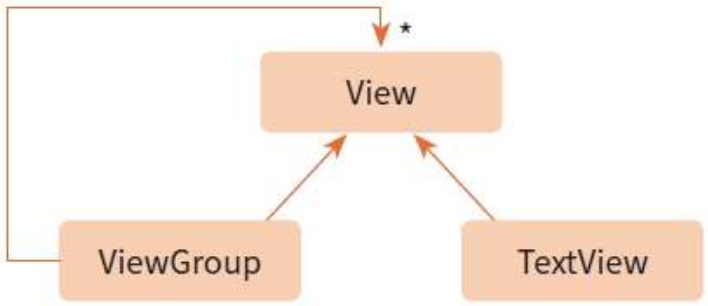
- public void setContentView(View view)
- public void setContentView(int layoutResID)



UI 기본 구조

뷰의 계층구조

- DOM(Document Object Model)을 따르고 있고, 패턴(Pattern)으로 이야기하자면 Gof 디자인 패턴의 Composite 패턴이 적용된 구조

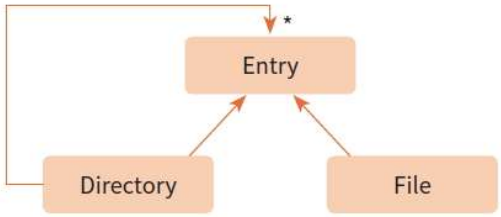


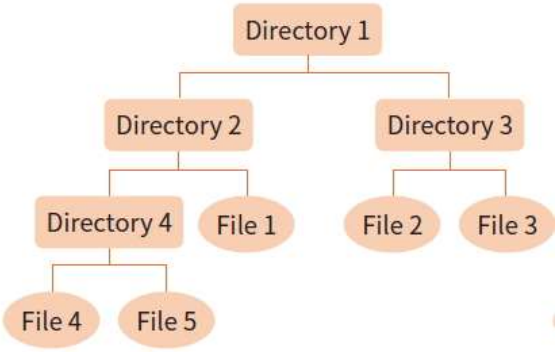
- View : 안드로이드 뷰 클래스의 최상위 클래스
- ViewGroup : 다른 뷰(Button 같은) 여러 개를 뷰그룹에 포함(Add)하여 한꺼번에 제어하기 위한 목적
- TextView : 특정 UI를 출력할 목적으로 제공되는 클래스

UI 기본 구조

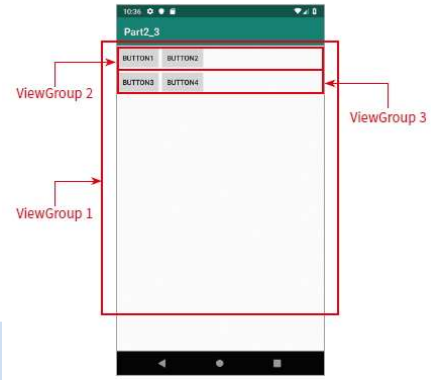
뷰의 계층구조

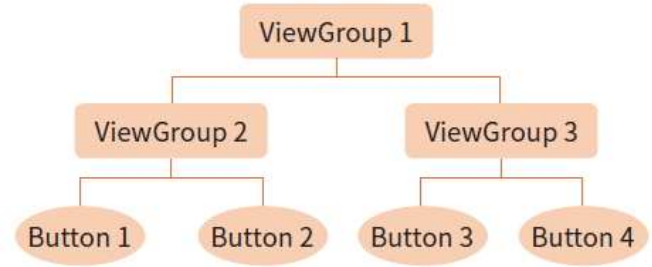
- 윈도우 탐색 창의 파일과 디렉터리의 계층구조





- 뷰 클래스의 객체들을 계층구조로 묶어서 사용





UI 기본 구조

뷰의 계층구조

- 레이아웃 XML로 계층구조 구현

```
<LinearLayout
    android:layout_width=”match_parent”
    android:layout_height=”wrap_content”
    android:orientation=”horizontal”>
    <Button
        android:layout_width=”wrap_content”
        android:layout_height=”wrap_content”
        android:text=”Button 1”/>
    <Button
        android:layout_width=”wrap_content”
        android:layout_height=”wrap_content”
        android:text=”Button 2”/>
</LinearLayout>
```

UI 기본 구조

뷰의 계층구조

- 자바 코드로 계층구조 구현

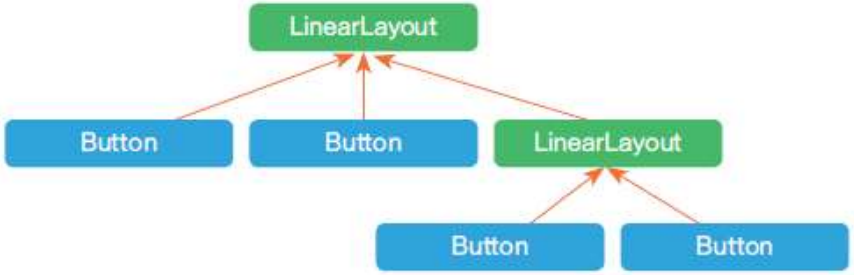
```
LinearLayout linearLayout=new LinearLayout(this);

Button bt1=new Button(this);
linearLayout.addView(bt1);
Button bt2=new Button(this);
linearLayout.addView(bt2);
```

UI 기본 구조

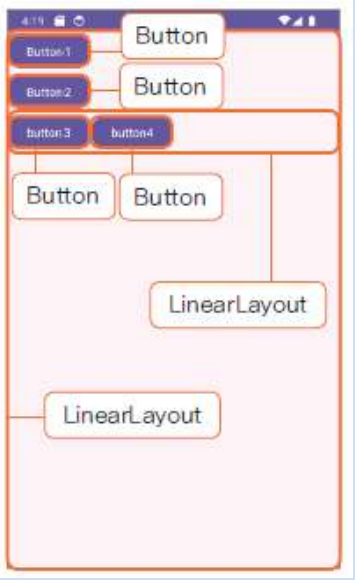
레이아웃 중첩

- 뷰의 계층 구조는 레이아웃 객체를 중첩해서 복잡하게 구성 가능



• 레이아웃 중첩

```
<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="wrap_content"
        android:layout_height="wrap_content"
        android:text="BUTTON1" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="BUTTON2" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="BUTTON3" />
        <Button
            android:layout_width="wrap_content"
```



뷰 기초 속성

- id 속성
  - 뷰의 식별자 속성, 필수 속성은 아니며 필요할 때 추가
  - 지정한 id 값은 R.java 파일에 등록
  - XML에서 등록한 id 값을 매개변수로 하여 findViewById( ) 함수로 획득

```
<TextView
    android:id="@+id/myText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="hello"/>
```

```
public static final class id {
    ...
    public static final int myText=0x7f0b0059;
    ...
}
```

```
TextView myTextView=findViewById(R.id.myText);
```

뷰 기초 속성

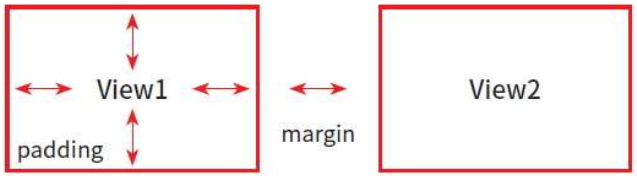
- layout_width, layout_height
  - 뷰의 크기를 지정하기 위한 속성
  - match_parent, fill_parent, wrap_content, 100px
  - match_parent와 fill_parent는 의미상 동일, 뷰의 크기를 부모 계층의 뷰가 지정한 크기에 꽉 들어차게 자동으로 결정
  - wrap_content는 해당 뷰의 내용을 화면에 보이기 위한 적절한 크기를 계산해서 결정

뷰 기초 속성

- layout_width, layout_height

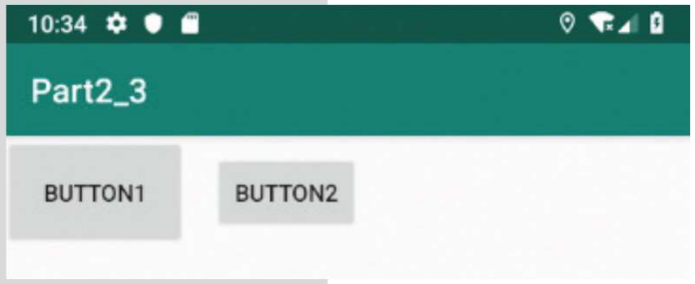
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_lab3_2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#F7FB08">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="hello"
        android:background="#FF0000"
        android:textColor="#FFFFFF"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="world"
        android:background="#0000FF"
        android:textColor="#FFFFFF"/>
</LinearLayout>
```

뷰 기초 속성



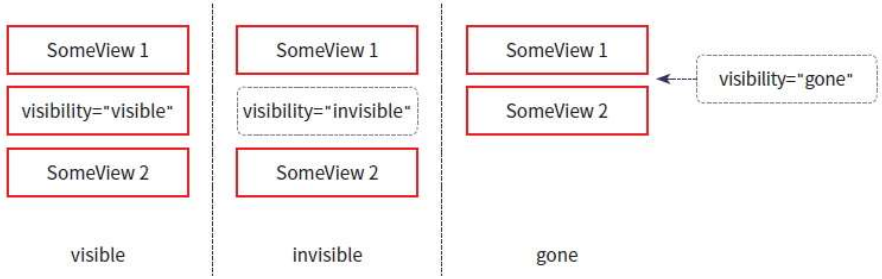
- margin, padding 속성
  - margin은 뷰와 뷰 사이의 간격을 지정하는 속성이며, padding은 뷰 내부에서 내용과 뷰의 테두리 간의 간격을 지정하는 속성
- 단일 방향 margin 속성
  - layout_marginLeft, layout_marginRight, layout_marginTop, layout_marginBottom
- 단일 방향 padding 속성
  - paddingLeft, paddingRight, paddingTop, paddingBottom

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button1"
    android:padding="24dp" />
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button2"
    android:layout_marginLeft="16dp" />
```



뷰 기초 속성

- clickable 속성
  - clickable 속성은 TextView, ImageView 등에 클릭 이벤트가 발생하게 하는 속성
- visibility 속성
  - 기본값은 "true"이며, "invisible", "gone" 으로 지정하여 화면에 안 보이게 함




```
public void onClick(View v) {

  if(v==trueBtn){
    targetBtn.setVisibility(View.VISIBLE);
  }else if(v==fasleBtn){
    targetBtn.setVisibility(View.INVISIBLE);
  }
}
```

감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.

윌리엄 셰익스피어
William Shakespeare