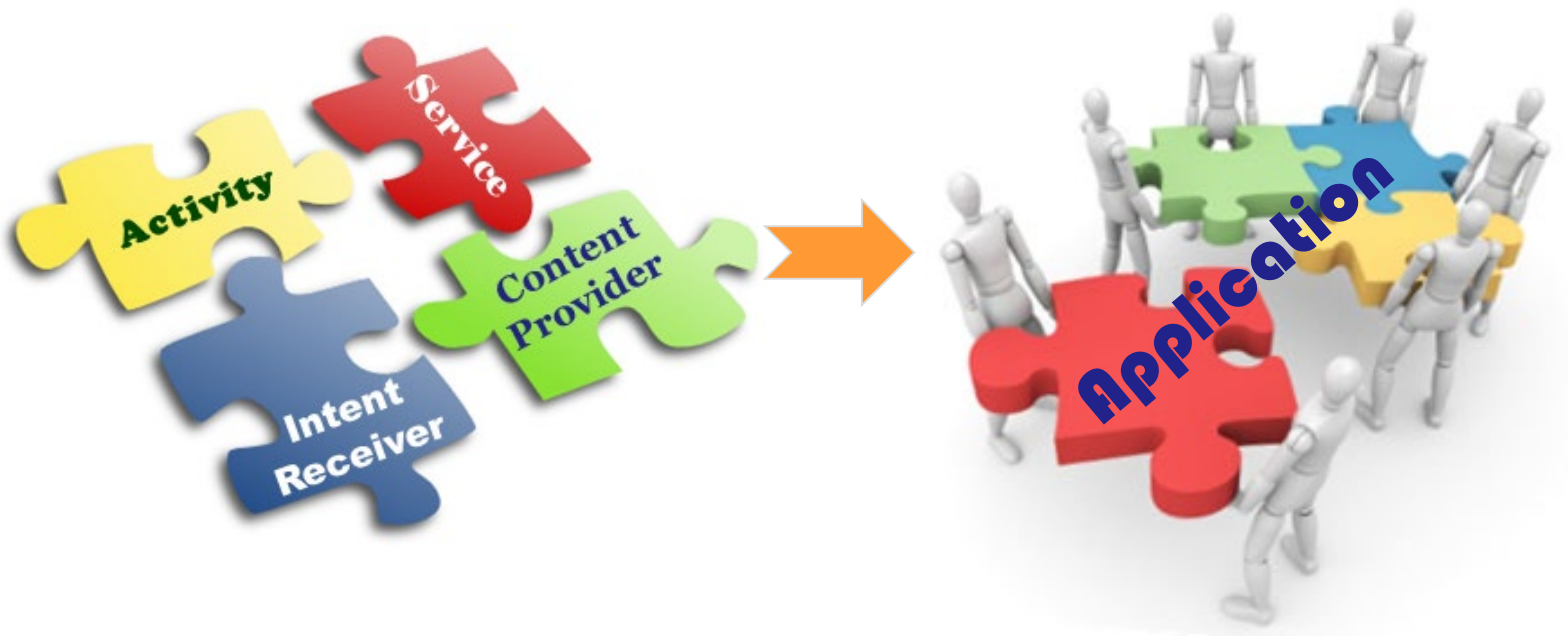


android



Application 구성요소

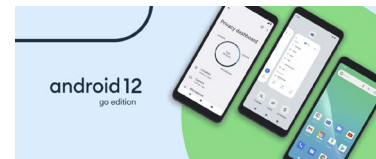
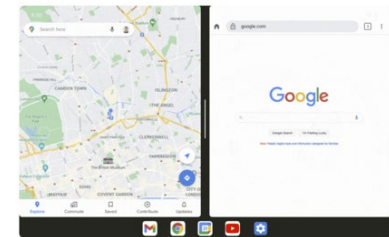
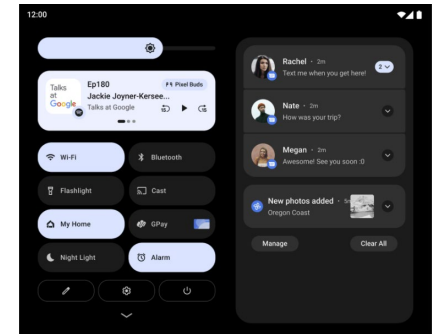
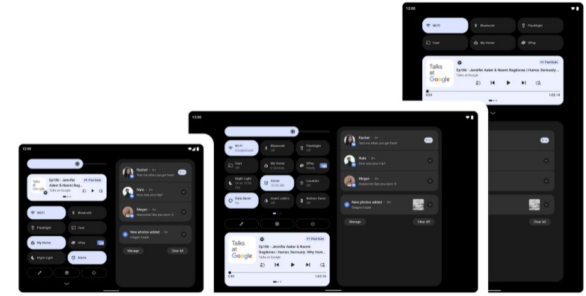




학습 목표

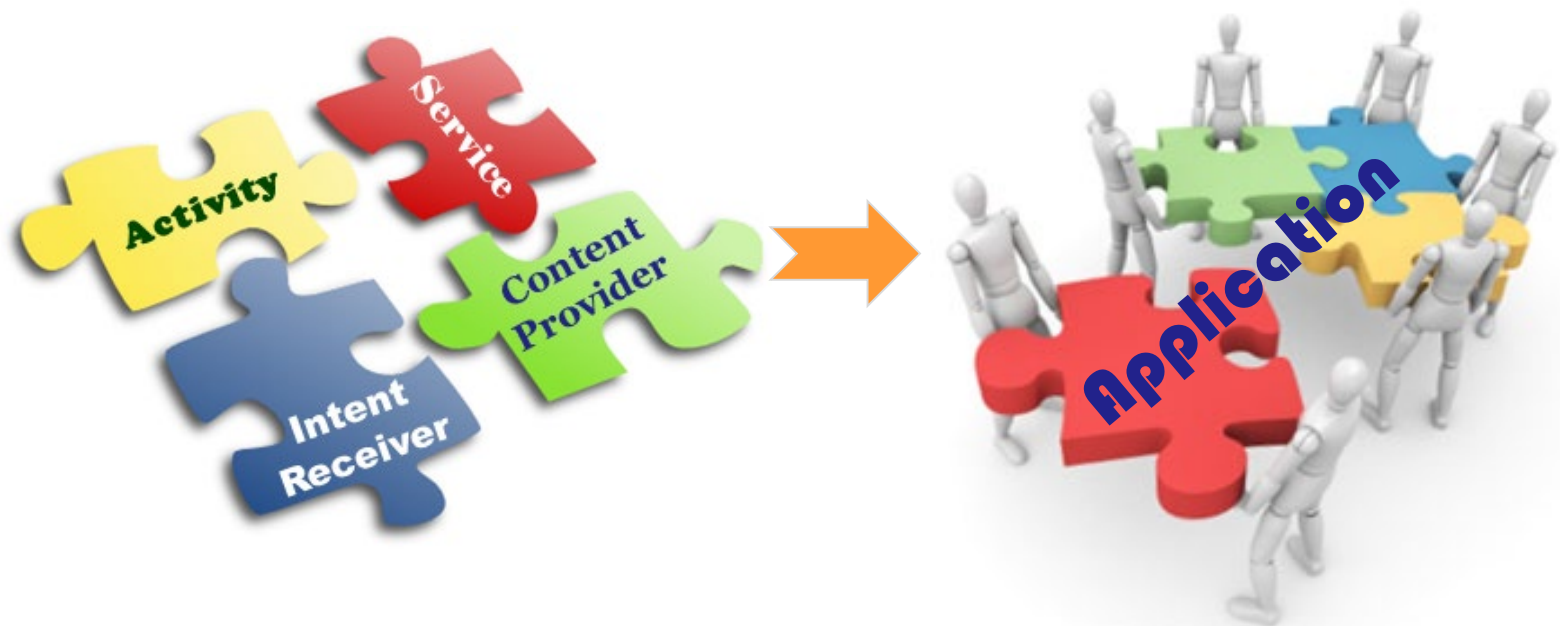
■ 교육 목표

- ❖ Android application 구성 요소
- ❖ Activity
- ❖ Layout
- ❖ UI 설계



Android application 구성 요소 (1)

■ Application 구성 요소



AndroidManifest.xml





Android application 구성 요소 (2)

구성 요소	기능
Activity	<ul style="list-style-type: none">• Application의 프리젠테이션 계층• UI component를 화면에 표시• Windows application의 Form과 동일 기능
Intent Receiver	<ul style="list-style-type: none">• 메시지 전달 framework• Event handler 기능
Service	<ul style="list-style-type: none">• UI와 관계없이 오랫동안 존재하면서 실행되는 코드• background로 실행(예: MediaPlayer)
Content Provider	<ul style="list-style-type: none">• Application data 공유• DB(SQLite) 연동





Android application 구성 요소 (3)

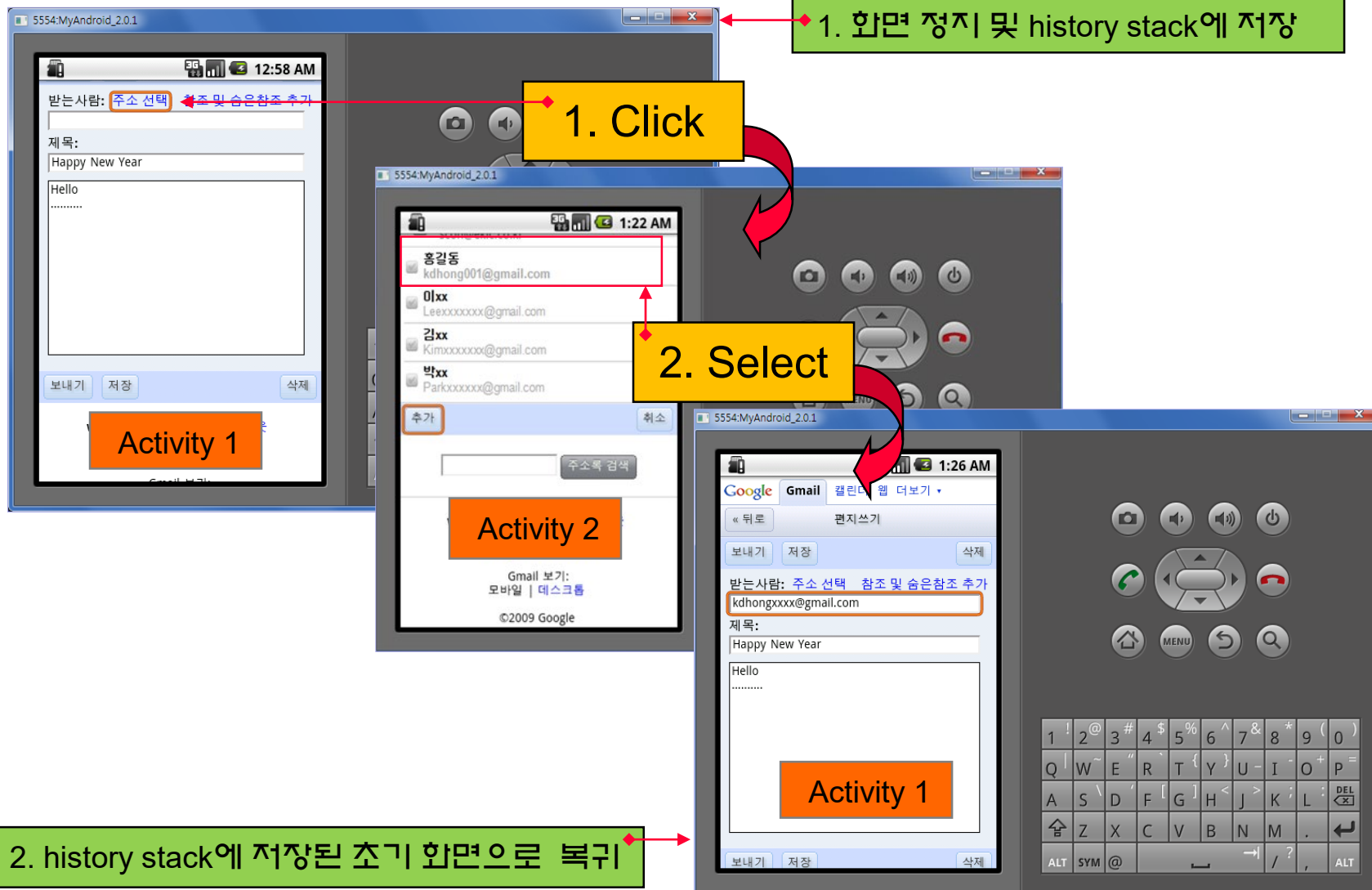
❖ AndroidManifest.xml

```
AndroidManifest.xml ×
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.inhatc.lee_won_joo.welcome">
4
5      <application
6          android:allowBackup="true"
7          android:icon="@mipmap/ic_launcher"
8          android:label="Welcome"
9          android:roundIcon="@mipmap/ic_launcher_round"
10         android:supportsRtl="true"
11         android:theme="@style/AppTheme">
12         <activity android:name=".MainActivity">
13             <intent-filter>
14                 <action android:name="android.intent.action.MAIN" />
15
16                 <category android:name="android.intent.category.LAUNCHER" />
17             </intent-filter>
18         </activity>
19     </application>
20
21 </manifest>
22
```

Text Merged Manifest

Activity (1)

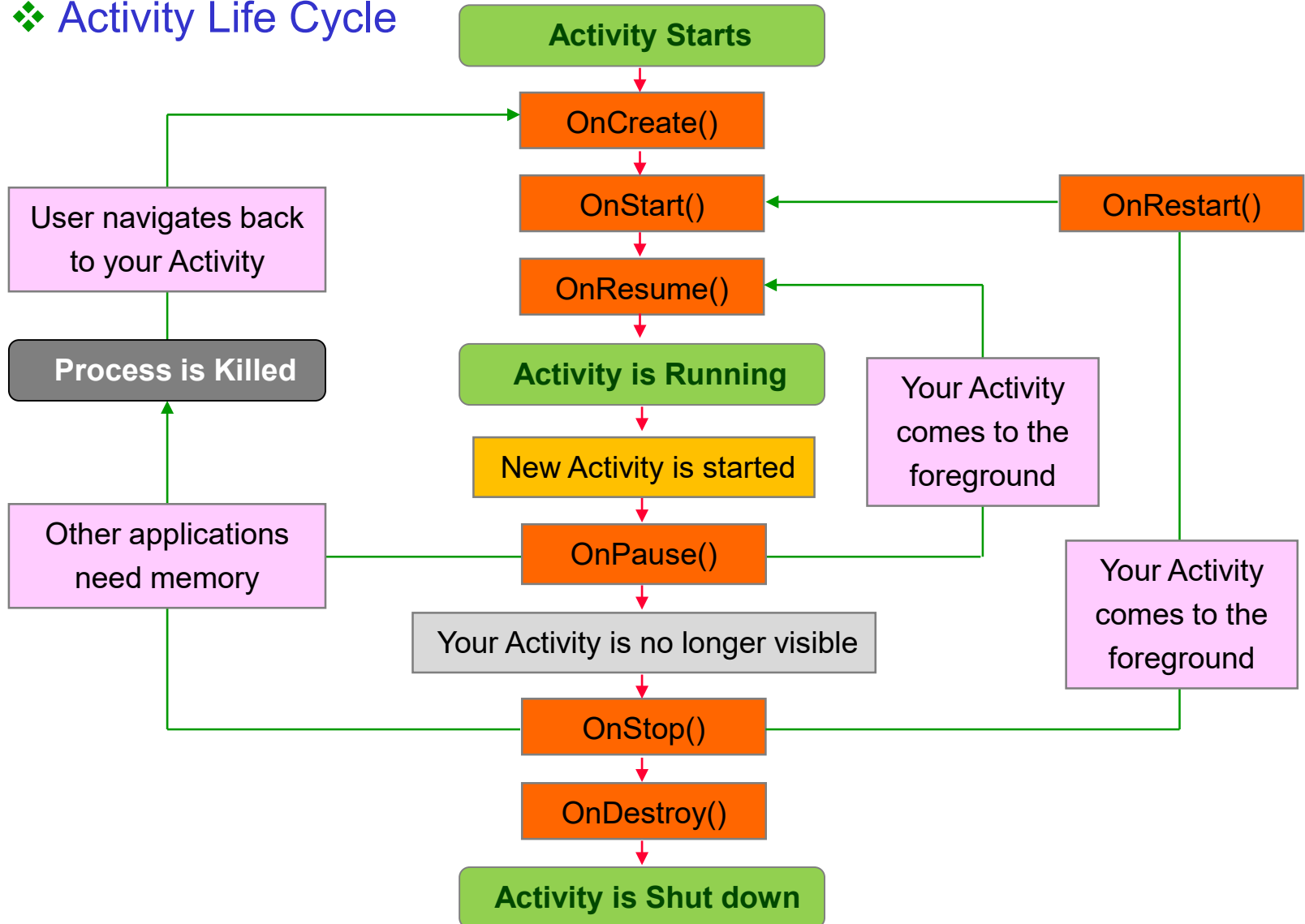
Activity





Activity (2)

❖ Activity Life Cycle





Activity (3)

❖ Activity Method

구성 요소	기능
OnCreate ()	<ul style="list-style-type: none">• Activity가 생성될 때 처음 호출• 리소스 초기화• findViewById(int) 메소드 호출 : Widget 검색• setContentView(int) 메소드 호출 : Widget 화면 출력
OnStart ()	<ul style="list-style-type: none">• Activity 화면 출력시 호출
OnResume ()	<ul style="list-style-type: none">• Activity stack의 Top에 위치• Activity 화면 출력 및 입력 처리시 호출
OnPause ()	<ul style="list-style-type: none">• Activity 1 → Activity 2 : Activity 1의 내용 저장• 데이터 저장, animation 중지, CPU 점유 작업 중단 등 수행
OnStop ()	<ul style="list-style-type: none">• 더 이상 Activity가 stack의 Top에 위치하지 않으므로 Activity 출력 안됨
OnDestroy ()	<ul style="list-style-type: none">• 시스템에서 Activity 제거





Activity (4)

MainActivity.java

```
MainActivity.java x
1  package com.inhatc.lee_won_joo.android_edittext;
2
3  import android.support.v7.app.AppCompatActivity;
4  import android.os.Bundle;
5  import android.widget.EditText;
6
7  public class MainActivity extends AppCompatActivity {
8
9      @Override
10     protected void onCreate(Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
12         setContentView(R.layout.activity_main);
13
14         EditText objET = (EditText)findViewById(R.id.editText);
15         objET.setText("Google : ");
16         objET.setGravity(0x01);
17
18         String strData = objET.getText().toString();
19         objET.setText(strData + " Android Programming.");
20
21     }
22 }
```



Activity (5)

Process Life Cycle

Foreground activity

- 최상위 화면의 activity로 가장 중요한 프로세스
- 메모리 부족 시 마지막으로 제거



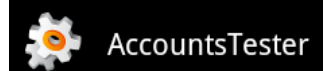
Visible activity

- 화면에 보이는 activity로 foreground activity가 아닌 activity



Background activity

- 화면에 보이지 않고 stop 상태의 activity
- 메모리 부족 시 foreground, visible activity보다 먼저 제거
- onSaveInstanceState(Bundle savedInstanceState) method 호출
– savedInstanceState : 중요 데이터 저장
- onCreate() method에서 savedInstanceState parameter를 이용하여 Background activity 제거 전의 상태로 복귀 가능



Empty Process

- Service, Broadcast Receiver 클래스와 같이 activity, application component도 없는 프로세스
- 메모리 부족 시 즉시 제거





Layout (1)

■ Layout

❖ 기능

◆ 화면 구성

❖ 분류

◆ ConstraintLayout

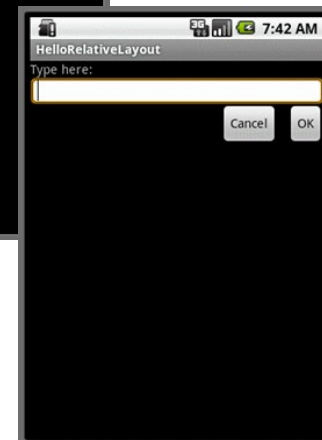
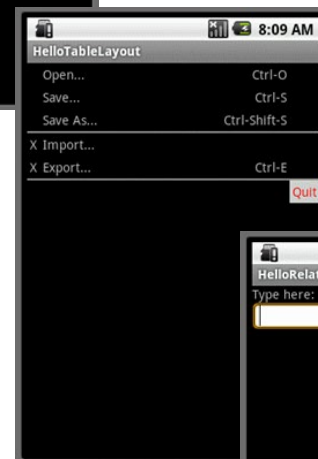
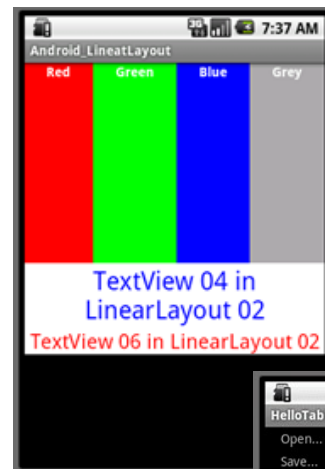
◆ FrameLayout

◆ LinearLayout

◆ TableLayout

◆ AbsoluteLayout

◆ RelativeLayout





Layout (2)

■ View 클래스

❖ 기능

- ◆ 직사각형 형태의 layout과 각종 정보 저장
- ◆ 화면 크기 조절, Layout 구성, draw, focus 변환
- ◆ Scrolling, key 처리
- ◆ Widget의 base class

❖ Widget

- ◆ Text, EditText, Button, RadioButton, Checkbox, ScrollView 등과 같이 화면에 컴포넌트처럼 동작하는 것
- ◆ Widget을 이용하면 UI를 빠르게 구현





Layout (3)

❖ View 크기 지정 단위

- ◆ px (pixels)
- ◆ dip (device independent pixels)
- ◆ sp (scaled pixels – best for text size)
- ◆ pt (points)
- ◆ in (inches)
- ◆ mm (millimeters)





Layout (4)

■ ViewGroup

❖ View 집합체

Class	설 명
ConstraintLayout	<ul style="list-style-type: none">• Android Studio 2.2의 향상된 레이아웃 디자이너와 함께 사용• 기존의 Layout 보다 쉽고 빠르게 Layout 구성
FrameLayout	<ul style="list-style-type: none">• 단일 객체를 표현하기 위한 View frame 기능• 2개 이상의 객체를 표현할 때는 마지막으로 그린 객체 화면 출력
LinearLayout	<ul style="list-style-type: none">• 수평/수직 으로 구성 요소 배치• 창의 길이가 화면 길이를 초과하면 scrollbar 자동 생성
TableLayout	<ul style="list-style-type: none">• Table• 행 : 가장 큰 구성 요소 크기에 맞게 설정• 셀 테두리 : hidden
AbsoluteLayout	<ul style="list-style-type: none">• 객체의 위치를 좌표값(x, y)으로 지정
RelativeLayout	<ul style="list-style-type: none">• 하나의 구성 요소를 기준으로 Up/Down, Left/Right 로 상대적 위치 지정
Gallery	<ul style="list-style-type: none">• image를 수평으로 scroll하면서 보여주는 클래스





Layout (5)

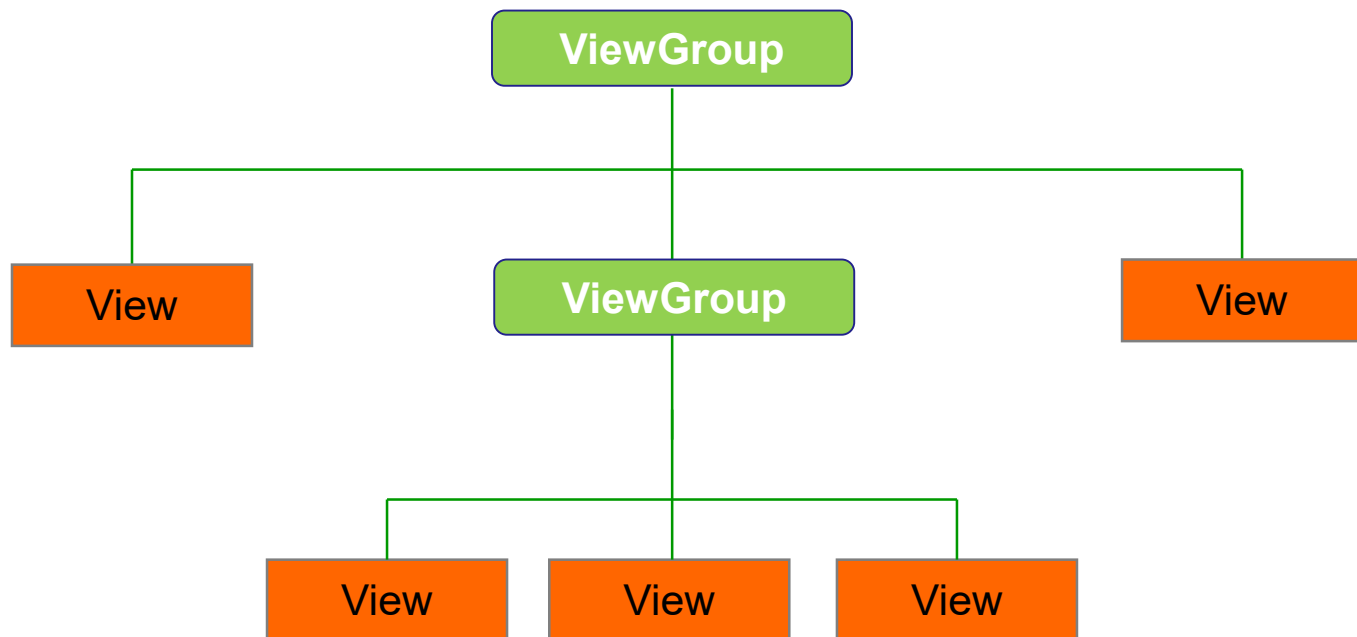
❖ ViewGroup

Class	설 명
GridView	• 눈금 구조를 보여주는 클래스
ListView	• scrolling 되는 column list를 보여줌
ScrollView	• 수직으로 구성 요소의 열을 scrolling
Spinner	• 1 line textbox에 list item을 하나씩 출력
SurfaceView	• 그리기 전용 표면에 직접 접근 가능 • 점과 line을 직접 draw 하는 low-level coding에 사용
TabHost	• click에 반응하는 Tab 목록 제공 • Tab 클릭시 application 변경
ViewFlipper	• 한번에 1 개의 item 출력 • slider show와 같이 주기적으로 item 변경 출력
ViewSwitcher	• ViewFlipper와 동일 기능



Layout (6)

■ Tree 구조의 User Interface



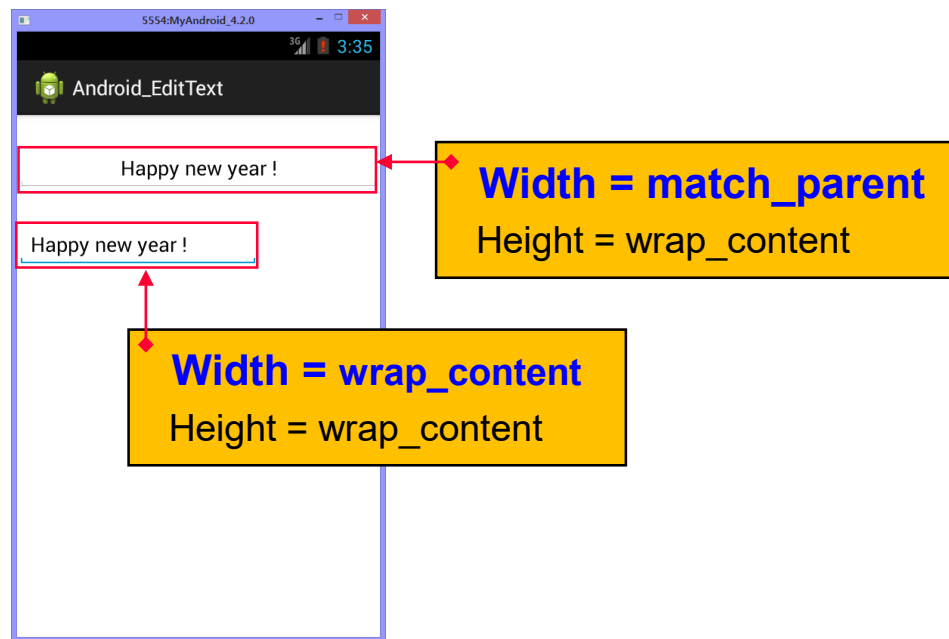


Layout (7)

■ LayoutParams 클래스

❖ 기능

- ◆ View 객체를 그리기 위한 정보를 부모 객체에 전송
- ◆ 각 객체의 Width, Height 설정
 - WRAP_CONTENT : 필요한 최소 크기
 - FILL_PARENT : 부모 객체와 여백을 제외한 모든 공간

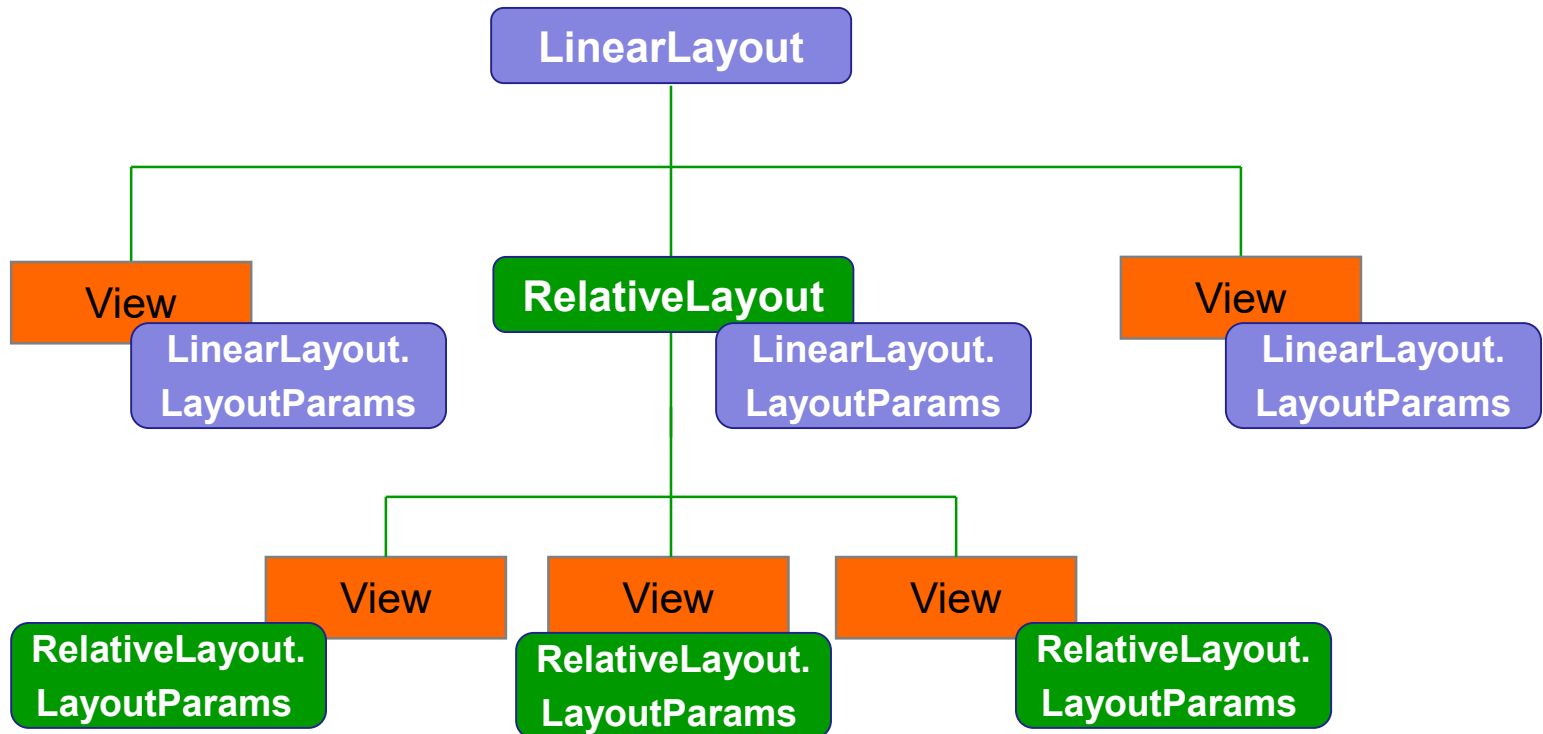


Layout (8)

LayoutParams subclass 설정

❖ 자식 객체

◆ 부모 객체의 속성에 맞는 LayoutParams 지정





UI 설계

■ UI 설계

❖ 방법 1

◆ XML 프로그래밍

- \res\layout\activity_main.xml

❖ 방법 2

◆ Java 프로그래밍

- \src\MainActivity.java

```
7      <TextView
8          android:layout_width="fill_parent"
9          android:layout_height="wrap_content"
10         android:text="@string/hello" />
11
12     <EditText
13         android:id="@+id/editText1"
14         android:layout_width="match_parent"
15         android:layout_height="wrap_content"
16         android:editable="true"
17         android:inputType="text"/>
```

Element

Attribute

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

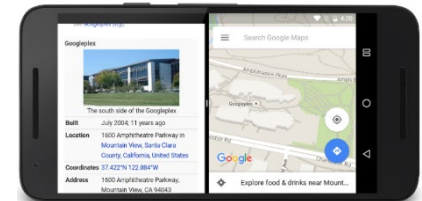
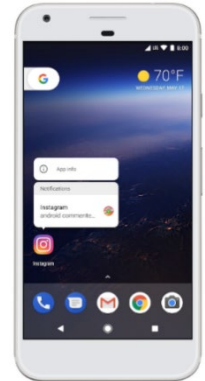
    EditText objET = (EditText)findViewById(R.id.editText1);
    objET.setText("Google : ");
    objET.setGravity(0x01);
    String strData = objET.getText().toString();
    objET.setText(strData + " Android Programming.");
}
```





학습 요약

- Android application 구성 요소
- Activity
- Layout
- UI 설계



open handset alliance