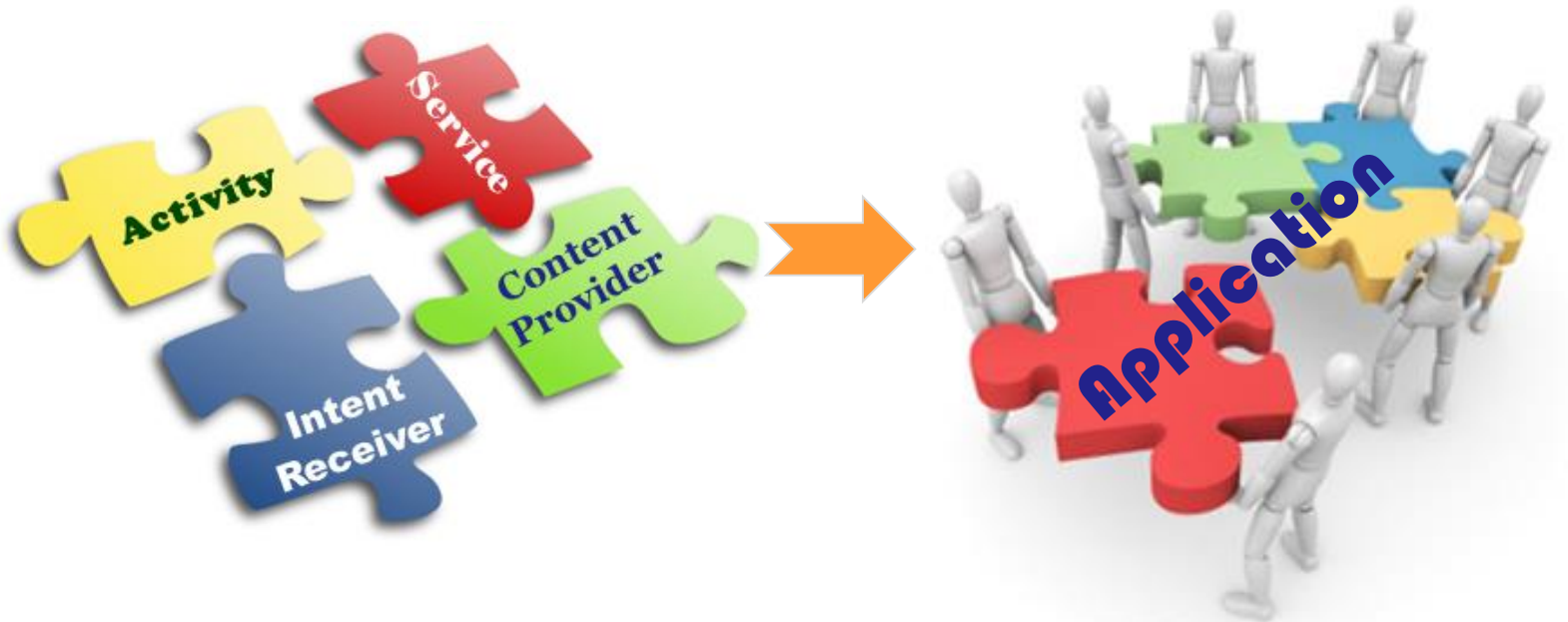




# android



## Application Component Element

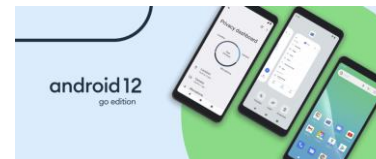
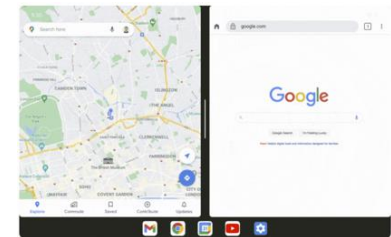
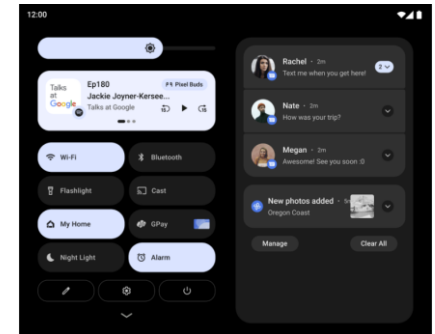




# Lesson Objectives

## ■ Contents

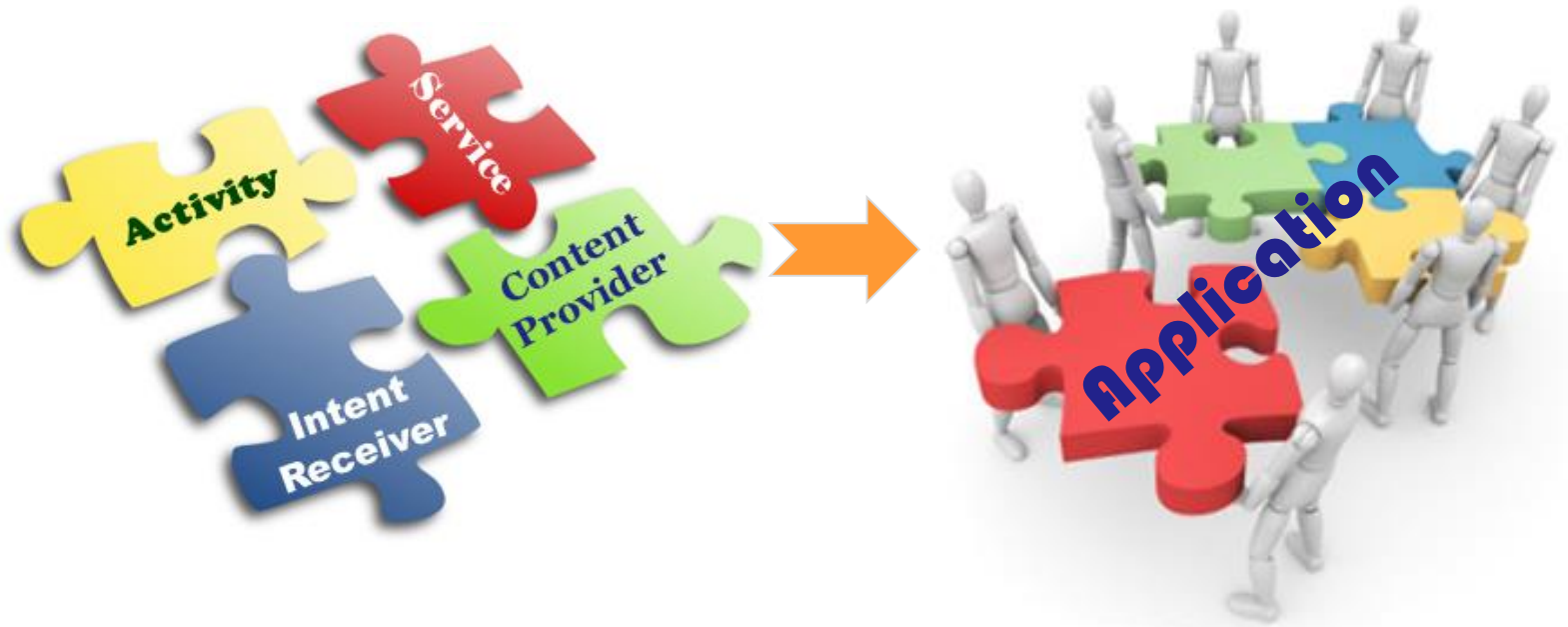
- ❖ Android application Component Element
- ❖ Activity
- ❖ Layout
- ❖ Design





# Android App Component Element (1)

■ Application 구성 요소



AndroidManifest.xml





# Android App Component Element (2)

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





# Android App Component Element (3)

## ❖ AndroidManifest.xml

```
Welcome - AndroidManifest.xml [Welcome.app.main]
AndroidManifest.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools">
4
5     <application
6         android:allowBackup="true"
7         android:dataExtractionRules="@xml/data_extraction_rules"
8         android:fullBackupContent="@xml/backup_rules"
9         android:icon="@mipmap/ic_launcher"
10        android:label="Welcome"
11        android:roundIcon="@mipmap/ic_launcher_round"
12        android:supportsRtl="true"
13        android:theme="@style/Theme.Welcome"
14        tools:targetApi="31">
15        <activity
16            android:name=".MainActivity"
17            android:exported="true">
18            <intent-filter>
19                <action android:name="android.intent.action.MAIN" />
20
21                <category android:name="android.intent.category.LAUNCHER" />
22            </intent-filter>
23        </activity>
24    </application>
```

Text Merged Manifest

CRLF UTF-8 4 spaces



# Activity (1)

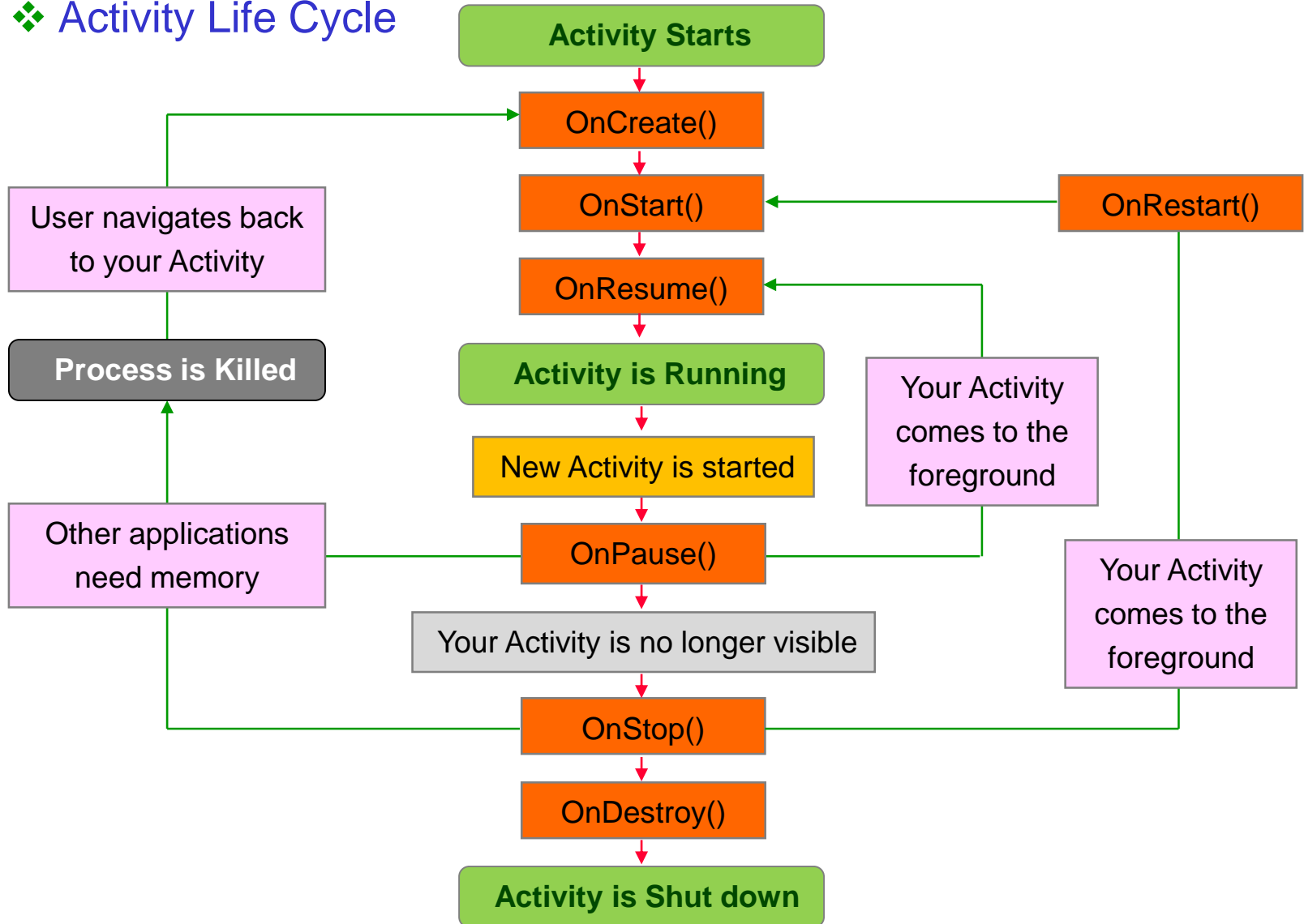
## Activity





# Activity (2)

## ❖ Activity Life Cycle







# Activity (3)

## ❖ Activity Method

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







# Activity (4)

## MainActivity.java

```
Android_EditText - MainActivity.java [Android_EditText.app.main]
MainActivity.java x
1 package com.inhatec.android_edittext;
2
3 > import ...
11
12 < > public class MainActivity extends AppCompatActivity {
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         EdgeToEdge.enable($this$enableEdgeToEdge: this);
18         setContentView(R.layout.activity_main);
19
20         EditText objET = (EditText)findViewById(R.id.editText);
21         objET.setText("Google : ");
22         objET.setGravity(0x01);
23
24         String strData = objET.getText().toString();
25         objET.setText(strData + " Android Programming.");
26
27     }
28
29     ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
30
31     }
32
33 }
```



# Activity (5)

## Process Life Cycle

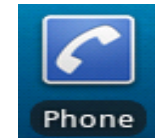
### Foreground activity

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



### Visible activity

- 화면에 보이는 activity로 foreground activity가 아닌 activity



### Background activity

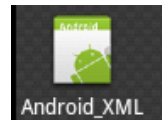
- 화면에 보이지 않고 stop 상태의 activity
- 메모리 부족 시 foreground, visible activity보다 먼저 제거
- onSaveInstanceState(Bundle savedInstanceState) method 호출  
– savedInstanceState : 중요 데이터 저장
- onCreate( ) method에서 savedInstanceState parameter를 이용하여 Background activity 제거 전의 상태로 복귀 가능



AccountsTester

### Empty Process

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





# Layout (1)

## Layout

### ❖ Function

◆ User Interface

### ❖ Layout

◆ ConstraintLayout

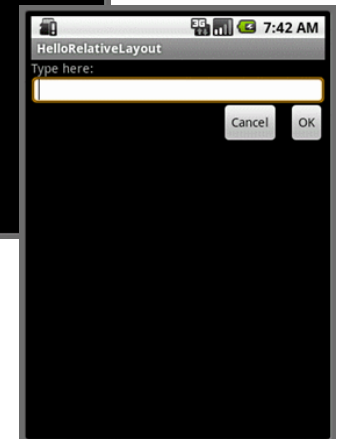
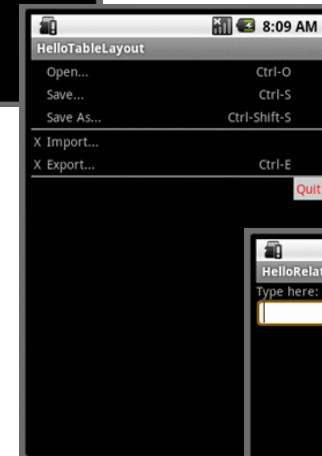
◆ FrameLayout

◆ LinearLayout

◆ TableLayout

◆ AbsoluteLayout

◆ RelativeLayout





# Layout (2)

## ■ View Class

### ❖ Function

- ◆ 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 Size Unit

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





# Layout (4)

## ■ ViewGroup

### ❖ Layout

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





# Layout (5)

## ❖ ViewGroup

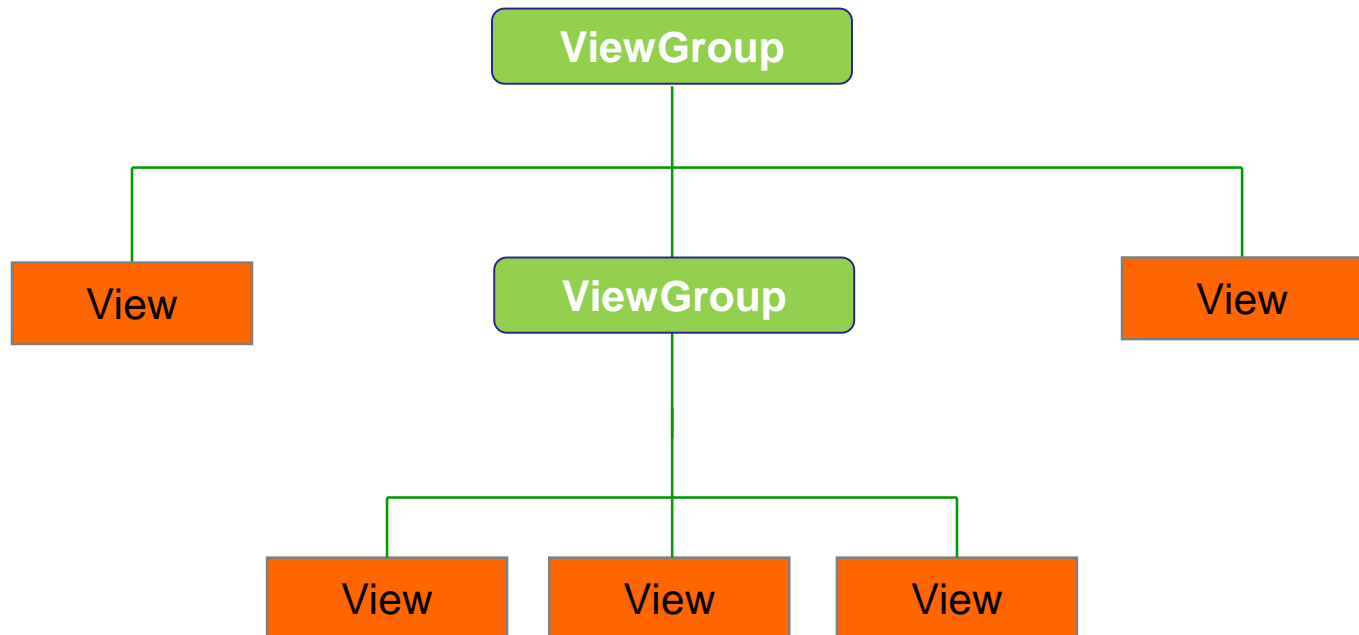
Class	Function
GridView	<ul style="list-style-type: none"><li>• 눈금 구조를 보여주는 Class</li></ul>
ListView	<ul style="list-style-type: none"><li>• scrolling 되는 column list를 보여줌</li></ul>
ScrollView	<ul style="list-style-type: none"><li>• 수직으로 구성 요소의 열을 scrolling</li></ul>
Spinner	<ul style="list-style-type: none"><li>• 1 line textbox에 list item을 하나씩 출력</li></ul>
SurfaceView	<ul style="list-style-type: none"><li>• 그리기 전용 표면에 직접 접근 가능</li><li>• 점과 line을 직접 draw 하는 low-level coding에 사용</li></ul>
TabHost	<ul style="list-style-type: none"><li>• click에 반응하는 Tab 목록 제공</li><li>• Tab 클릭시 application 변경</li></ul>
ViewFlipper	<ul style="list-style-type: none"><li>• 한번에 1 개의 item 출력</li><li>• slider show와 같이 주기적으로 item 변경 출력</li></ul>
ViewSwitcher	<ul style="list-style-type: none"><li>• ViewFlipper와 동일 기능</li></ul>





# Layout (6)

## ■ Tree 구조의 User Interface





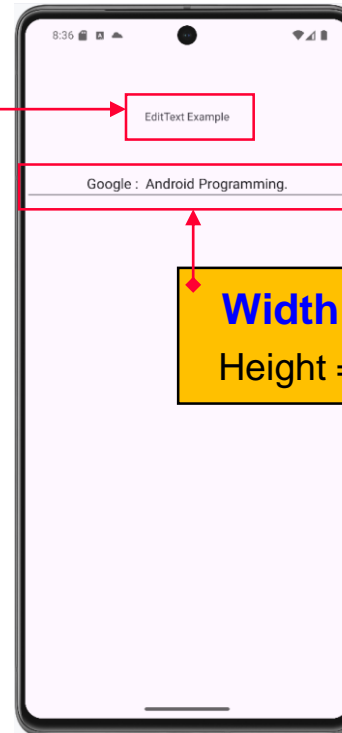
# Layout (7)

## ■ LayoutParams Class

### ❖ 기능

- ◆ View Object를 그리기 위한 정보를 부모 Object에 전송
- ◆ 각 Object의 Width, Height 설정
  - WRAP\_CONTENT : 필요한 최소 크기
  - FILLPARENT : 부모 Object와 여백을 제외한 모든 공간

**Width = wrap\_content**  
Height = wrap\_content



**Width = match\_parent**  
Height = wrap\_content



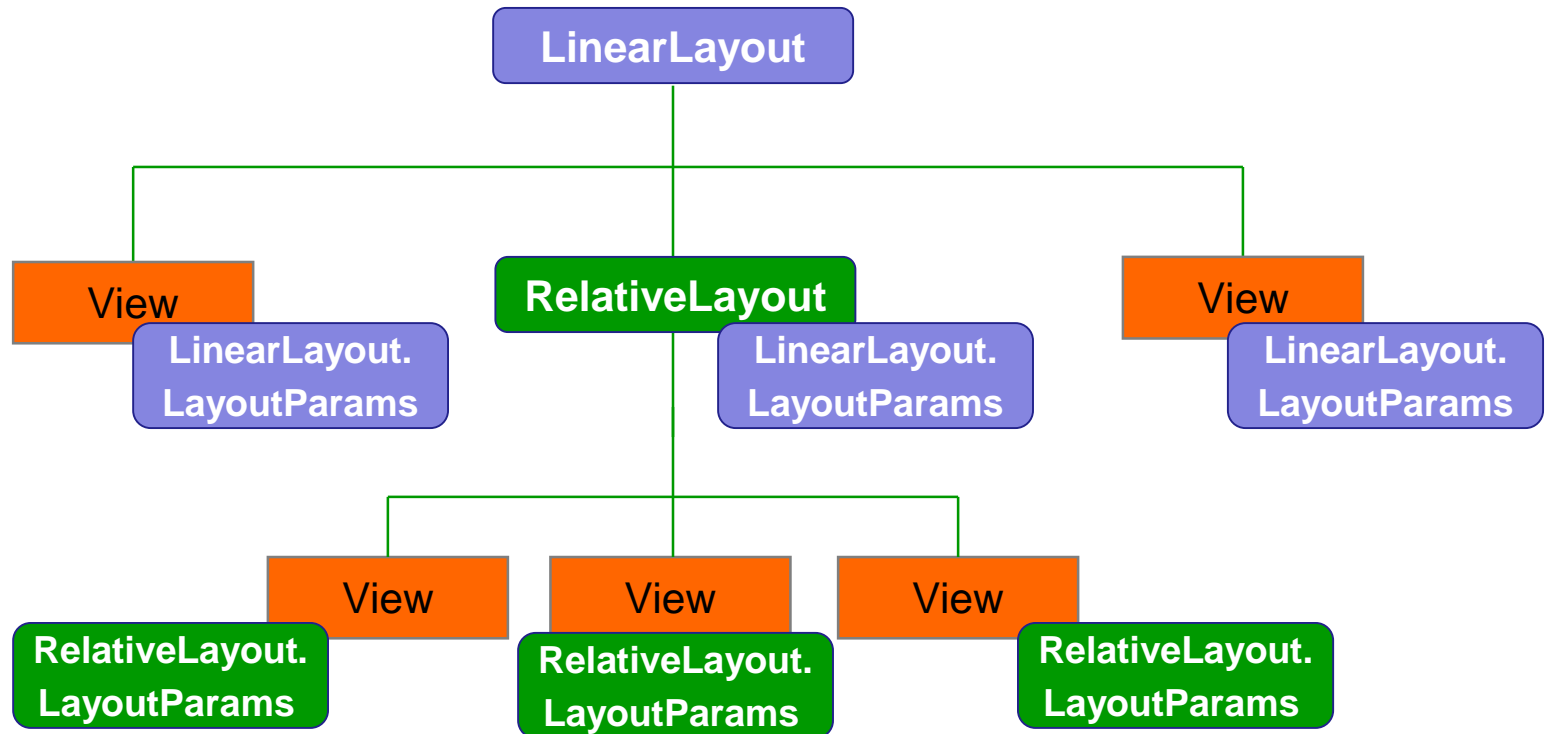


# Layout (8)

## ■ Setting LayoutParams subclass

❖ 자식 Object

◆ Setting 부모 Object의 속성에 맞는 LayoutParams





# Design UI (1)

## ■ Design UI

### ❖ 방법 1

#### ◆ XML Coding

- \res\layout\activity\_main.xml

```
22 <EditText
23     android:id="@+id/editText"
24     android:layout_width="359dp"
25     android:layout_height="41dp"
26     android:layout_marginStart="35dp"
27     android:layout_marginTop="146dp"
28     android:editable="true"
29     android:ems="10"
30     android:inputType="text"
31     android:text="Name"
32     app:layout_constraintStart_toStartOf="parent"
33     app:layout_constraintTop_toTopOf="parent" />
34
35 </androidx.constraintlayout.widget.ConstraintLayout>
```





# Design UI (2)

## ❖ 방법 2

### ◆ Java Programming

- \src\MainActivity.java

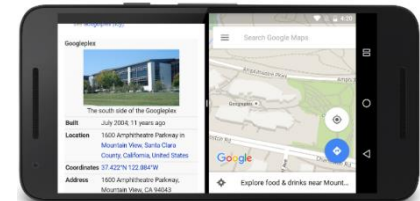
```
Android_EditText - MainActivity.java [Android_EditText.app.main]
MainActivity.java x
14      @Override
15      protected void onCreate(Bundle savedInstanceState) {
16          super.onCreate(savedInstanceState);
17          EdgeToEdge.enable($this$enableEdgeToEdge: this);
18          setContentView(R.layout.activity_main);
19
20          EditText objET = (EditText)findViewById(R.id.editText);
21          objET.setText("Google : ");
22          objET.setGravity(0x01);
23
24          String strData = objET.getText().toString();
25          objET.setText(strData + " Android Programming.");
26
27          ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
32      }
33  }
```

Coding



# Summary

- ❖ Android App Component Element
- ❖ Activity
- ❖ Layout
- ❖ Design UI



open handset alliance