

Google Android 심화 학습

Dae-Ki Kang

개발 도구들(Development Tools)

- Android Emulator – 에뮬레이터
- Hierarchy Viewer – 구조 뷰어
- Draw 9-patch
- Android Development Tools Plugin for the Eclipse IDE – 개발 툴
- Dalvik Debug Monitor Service (ddms) – 디버그 모니터 서비스
- Android Debug Bridge (adb) – 디버그 브릿지
- Android Asset Packaging Tool (aapt) – 패키징 툴
- Android Interface Description Language (aidl) – 인터페이스 기술 언어
- sqlite3
- Traceview
- mkshcard
- dx
- UI/Application Exerciser Monkey
- activitycreator



안드로이드 응용 프로그램 모델(Android Application Model: Applications, Tasks, Processes, and Threads)

- An android package (패키지) (or .apk for short) is the file containing an application's code and its resources – 어플리케이션의 코드와 자원을 가지고 있는 파일
- A task (태스크, 작업) is generally what the user perceives as an “application” that can be launched – 사용자가 실행할 수 있는 “응용 프로그램” 으로 지각하고 있는 것들
- A process (프로세스) is a low-level kernel process in which an application's code is running – 응용 프로그램 코드가 실행되고 있는 저수준의 커널 프로세스

안드로이드 응용 프로그램의 생명 주기

Life Cycle of an Android Application

1. A foreground process is one that is required for what the user is currently doing. (전경 프로세스)
 - It is running an Activity at the top of the screen that the user is interacting with (its onResume() method has been called).
 - It has a BroadcastReceiver that is currently running (its BroadcastReceiver.onReceive() method is executing).
 - It has a Service that is currently executing code in one of its callbacks (Service.onCreate(), Service.onStart(), or Service.onDestroy()).
2. A visible process is one holding an Activity that is visible to the user on-screen but not in the foreground (its onPause() method has been called). (가시적인 프로세스, 보이는 프로세스)
3. A service process is one holding a Service that has been started with the startService() method. (서비스)
4. A background process (배경 프로세스) is one holding an Activity that is not currently visible to the user (its onStop() method has been called).
5. An empty process is one that doesn't hold any active application components. (비어있는 프로세스)



차례

- 무엇이 안드로이드인가
 - What is Android?
- 시작
 - Getting Started
- 응용 프로그램 개발
 - **Developing Applications**
- 참조 문서 정보
 - Reference Information
- 샘플 코드
 - Sample Code



응용 프로그램 개발 Developing Applications

- 사용자 인터페이스 구현
 - Implementing a UI
- 응용 프로그램의 세부 구성
 - Building Blocks
- 데이터 저장
 - Storing and Retrieving Data
- 보안 모델
 - Security Model
- 자원과 세계화
 - Resources and i18n

사용자 인터페이스 구현

Implementing a UI

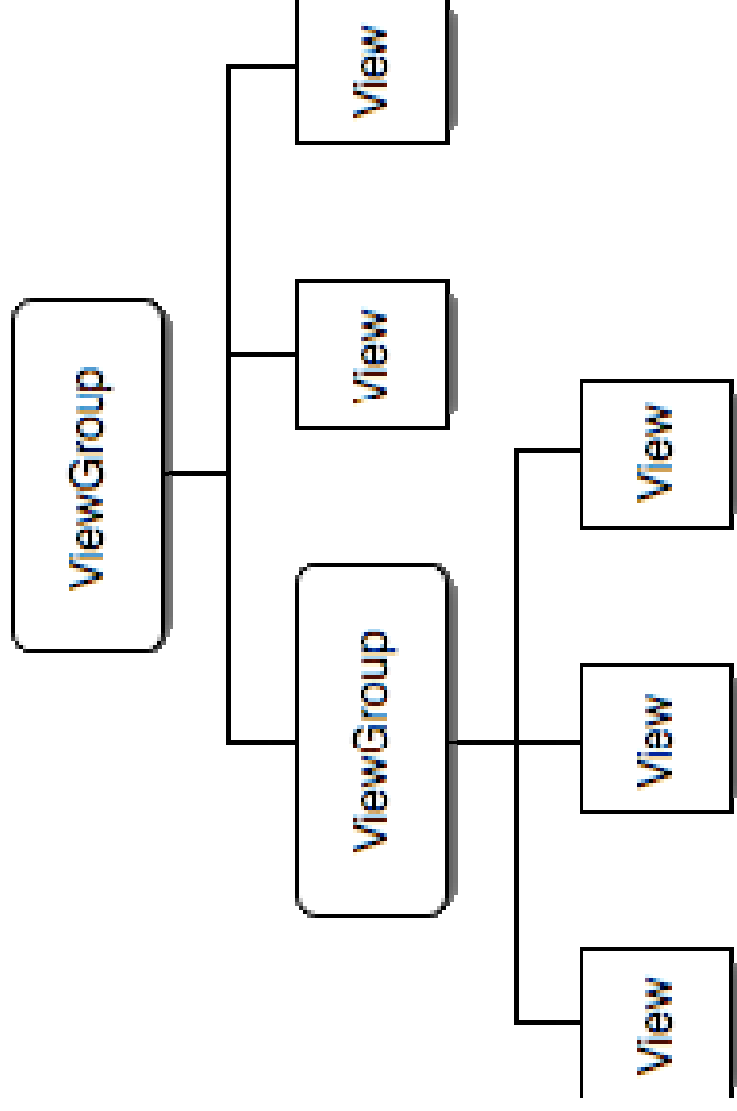
- View
 - `android.view.View` 를 기본 클래스로 가지는 객체
 - A view is an object (객체) of base class `android.view.View`.
 - 레이아웃과 속성들을 저장하고 있는 자료 구조
 - It's a data structure whose properties store the layout and content for a specific rectangular area of the screen.

사용자 인터페이스 구현

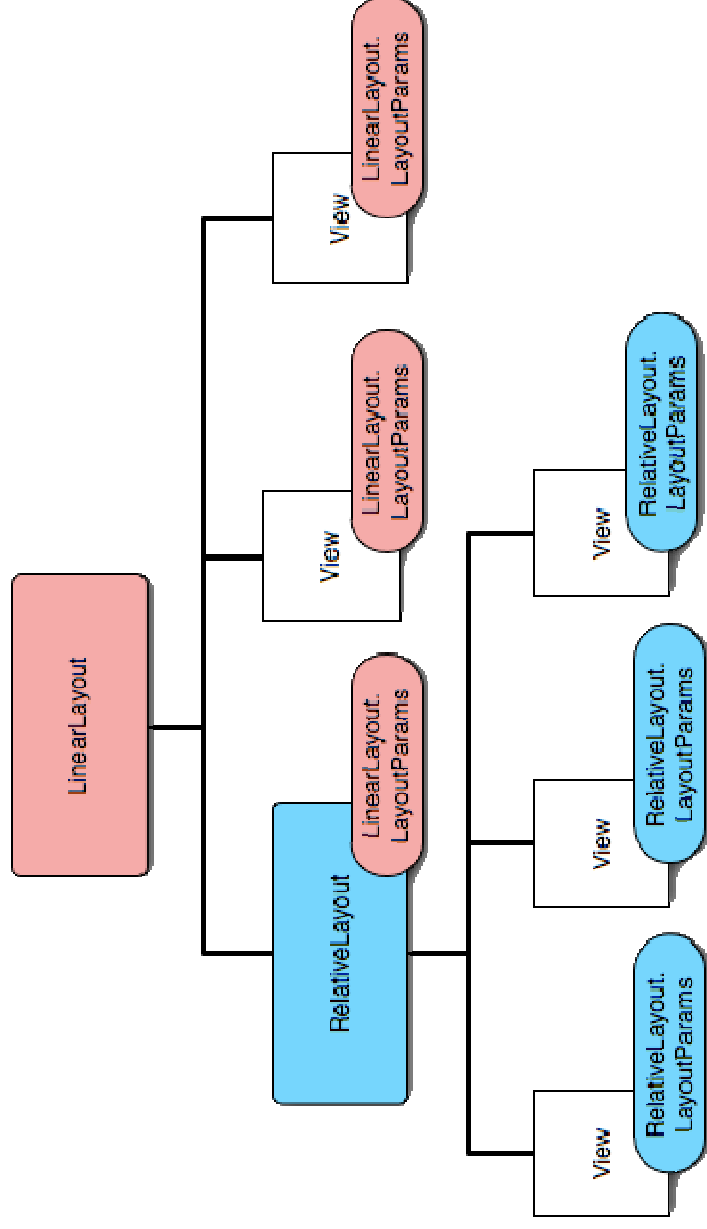
Implementing a UI

- **View**
 - A view is an object of base class `android.view.View`.
 - `android.view.View` 를 기본 클래스로 가지는 객체
 - It's a data structure whose properties store the layout and content for a specific rectangular area of the screen.
- **Viewgroups**
 - A viewgroup is an object of class `android.view.ViewGroup`.
- 뷰그룹과 뷰를 저장하는 컨테이너
 - A viewgroup is a special type of view object whose function is to contain and manage a subordinate set of views and other viewgroups.

트리 구조의 사용자 인터페이스 A Tree-Structured UI



레이아웃 파라미터) LayoutParams: How a Child Specifies Its Position and Size

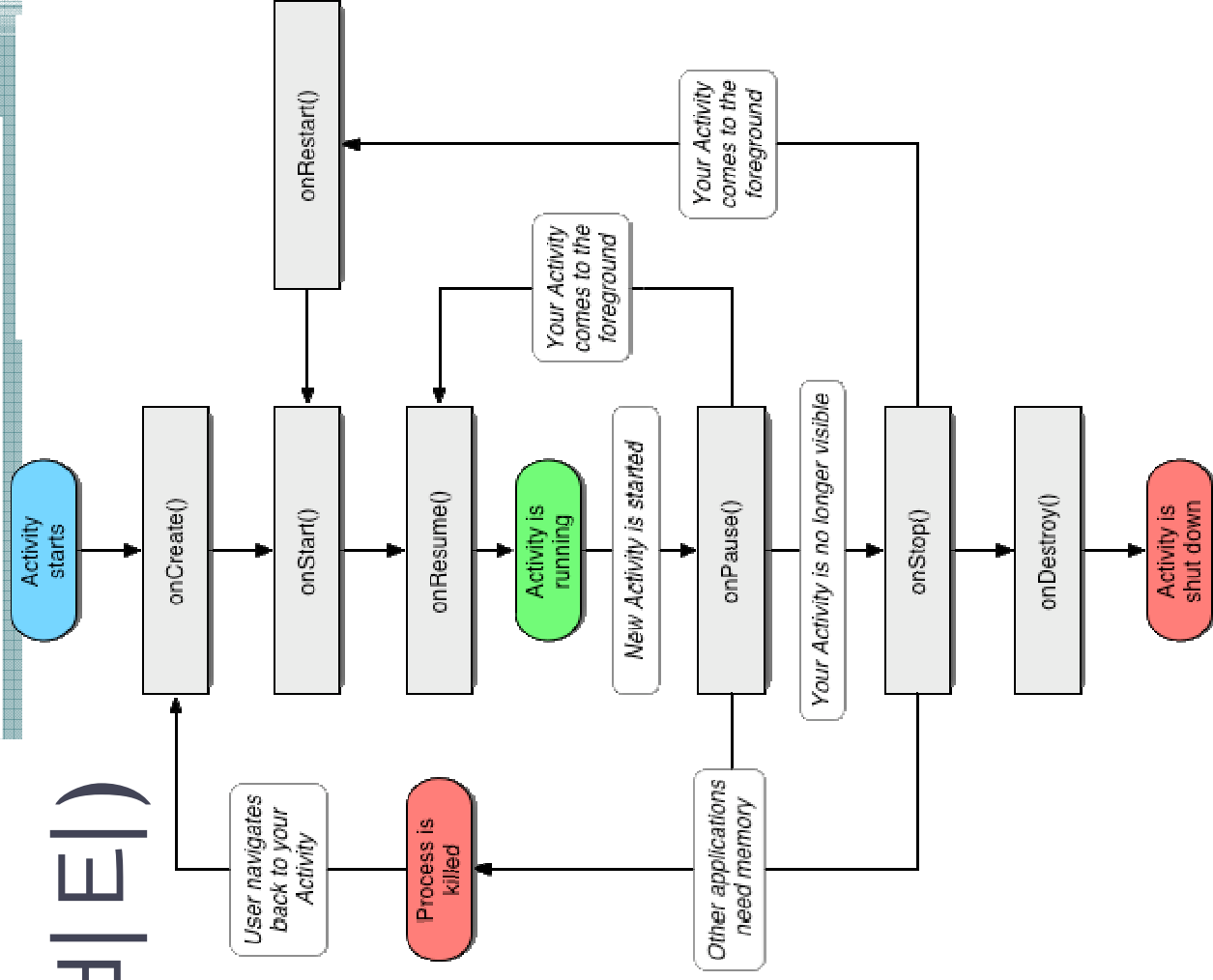


Building Blocks

- **AndroidManifest.xml (안드로이드 응용 프로그램 제어 파일)**
 - The control file that tells the system what to do with all the top-level components you've created – 탭-레벨 요소들을 어떻게 사용하고 연결할 것인지 지정하는 제어 파일
 - <http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- **Activities (액티비티)**
 - An Activity is, fundamentally, an object that has a life cycle – 수명 주기를 가지고 있는 객체로 하나의 화면과 대응됨
- **Views (뷰)**
 - A View is an object that knows how to draw itself to the screen. – 화면에 쓰여지는 그래픽 객체로, 윈도우 프로그래밍에서 콘트롤과 동일함
- **Intents (인텐트 – 의도라는 뜻)**
 - An Intent is a simple message object that represents an “intention” to do something. – 액티비티들 간에 주고받는 메시지 객체
- **Services (서비스)**
 - A Service is a body of code that runs in the background – 유닉스의 데몬같이 운영체제의 배경에서 오랫동안 작동되는 프로그램
- **Notifications (통지)**
 - A Notification is a small icon that appears in the status bar – 상태 바에 나타나는 조그만 아이콘 (스마트폰 운영체제들은 대부분 필수적으로 가지고 있음)
 - Users can interact with this icon to receive information.
- **ContentProviders (컨텐츠 프로바이더)**
 - A ContentProvider is a data storehouse that provides access to data on the device
 - 디바이스의 데이터에 접근하기 위해 사용되는 요소

Activities (액티비티)

- <http://developer.android.com/reference/android/app/Activity.html>



- View (뷰)
 - <http://developer.android.com/reference/android/view/View.html>
- Intents (인텐트, 의도)
 - <http://developer.android.com/reference/android/content/Intent.html>
- Services (서비스)
 - <http://developer.android.com/reference/android/app/Service.html>
- Notifications (통지)
 - <http://developer.android.com/reference/android/app/NotificationManager.html>
- ContentProviders (컨텐츠 프로바이더)
 - <http://developer.android.com/reference/android/content/ContentProvider.html>

데이터 저장

Storing and Retrieving Data

- Preferences (프레퍼런스)
 - A lightweight mechanism to store and retrieve key/value pairs of primitive data types – 키/밸류 기반의 데이터 타입을 저장하기 위한 메커니즘
- Files (파일)
 - You can store your files on the device or on a removable storage medium – 디바이스나 외부저장장치에 파일을 저장
- Databases (데이터베이스)
 - The Android APIs contain support for SQLite – 데이터베이스
- Content Providers (컨텐츠 프로바이더)
 - A content provider is an optional component of an application that exposes read/write access to an application's private data, subject to whatever restrictions it wants to impose – 응용프로그램의 개인적인 데이터를 읽고 쓰기 위한 컴포넌트
- Network (네트워크)
 - You can also use the network to store and retrieve data – 네트워크

보안 모델 (Security Model)

- <http://developer.android.com/guide/topics/security/security/security.html>
- Security Architecture
- Application Signing – 어플리케이션 사이닝
- User IDs and File Access – 사용자 아이디와 파일 접근 권한
- Using Permissions - 퍼미션
- Declaring and Enforcing Permissions – 퍼미션의 선언과 인코딩
 - Enforcing Permissions in AndroidManifest.xml
 - Enforcing Permissions when Sending Broadcasts
 - Other Permission Enforcement
- URI Permissions



차례

- 무엇이 안드로이드인가
 - What is Android?
- 시작
 - Getting Started
- 응용 프로그램 개발
 - Developing Applications
- 참조 문서 정보
 - Reference Information
- 샘플 코드
 - Sample Code



참조 정보

Reference Information

- [http://developer.android.com/reference/packag
es.html](http://developer.android.com/reference/packag
es.html)
-

HelloWorld.java

```
• package com.example.android.apis.app;

// Need the following import to get access to the app resources, since this
// class is in a sub-package.
import com.example.android.apis.R;

import android.app.Activity;
import android.os.Bundle;

public class HelloWorld extends Activity
{
    /**
     * Initialization of the Activity after it is first created. Must at least
     * call {@link android.app.Activity#setContentView setContentView()} to
     * describe what is to be displayed in the screen.
     */
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        // Be sure to call the super class.
        super.onCreate(savedInstanceState);

        // See assets/res/any/layout/hello_world.xml for this
        // view layout definition, which is being set here as
        // the content of our screen.
        setContentView(R.layout.hello_world);
    }
}
```

Examples

- API Demos
 - <http://developer.android.com/guide/samples/ApiDemos/index.html>
- Lunar Lander
 - <http://developer.android.com/guide/samples/LunarLander/index.html>
- Note Pad
 - <http://developer.android.com/guide/samples/NotePad/index.html>



차례

- 무엇이 안드로이드인가
 - What is Android?
- 시작
 - Getting Started
- 응용 프로그램 개발
 - Developing Applications
- 참조 문서 정보
 - Reference Information
- 샘플 코드
 - Sample Code

샘플 코드 학습 순서

1. 회색의 구글 안드로이드의 모바일 플레이스 예제들 실습 (http://www.mobileplace.co.kr/android_dev_info/2389)
2. 한백 전자 교육의 구글 안드로이드 예제들 실습
3. NotePad 응용 프로그램 및 LunarLander 와 APIDemo 들 (<http://developer.android.com/guide/samples>)

Installation (설치)

- Install Java SDK
 - <http://java.sun.com/javase/downloads/index.jsp>
- Install Eclipse
 - <http://www.eclipse.org/downloads/>
- Install Android SDK
 - <http://code.google.com/android/download.html>
- Install Android Plugin for Eclipse
 - Eclipse → Menu → Help → Software Update → Available Software
 - Add Site : <https://dl-ssl.google.com/android/eclipse/>



안드로이드 SDK를 위한 이클립스 설치 Setup Eclipse for Android SDK

- Menu → Windows → Preference
- Android on the left panel
- Choose Android SDK directory
- Apply and OK

이클립스 개발환경 사용법 및 프로 젝트 생성

- 참고 - <http://www.mobileplace.co.kr/641>



새 프로젝트

New Project

- Menu → File → New → Project
- Choose Android
- Project Name – Hello Android
- Package Name – dsu.android
- Activity Name – Main
- Application Name – HelloAndroid

프로젝트 실행

Run the Project

- Menu → Run → Run Configuration
- Right mouse click → New
- Name → AndroidConfiguration1
- Browse HelloAndroid Project
- Apply and Run

새 액티비티

New Activity

- Add New Class
- MyActivity
- Subclass of Activity → Browse
- `android.app.Activity`



레이아웃 XML

Layout XML

- under /res/layout folder
- New File
- Name the new file `myactivity.xml` (lower case!)
- Copy `main.xml` to `myactivity.xml`
- Change
 - `android:text="@string/hello"`
 - to whatever you like!



AndroidManifest 변경 (Update AndroidManifest)

- Whenever new activity → Update AndroidManifest
- Click AndroidManifest.xml
- Copy one more activity tags
- Change
 - `android:name, android:label (titlebar!)`

메소드 오버라이드

Override Methods

- MyActivity.java
- Right-click
- Source
- Override/Implement Methods
- Click onCreate(Bundle)
 - setContentView(R.layout.myactivity);
 - import android.util.Log;
 - Log.d("MyTag", "Print Test Log");



실행 컨피규레이션 Run Configuration

- Run Menu
 - Run Configuration
 - Launch `smartphone.android.MyActivity`
 - Apply
 - Run
-
- Click Menu on the emulator



안드로이드 강좌 4 - 액티비티 (Activity), 뷰(View), 레이아웃 (Layout)

- <http://www.mobileplace.co.kr/1050>

레이아웃 XML

Layout XML

1. `<?xml version="1.0" encoding="utf-8"?>`
2. `<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
>`
3. `<TextView`
4. `android:layout_width="fill_parent"`
5. `android:layout_height="wrap_content"`
6. `android:text="Hello, Oman"`
7. `</LinearLayout>`

뷰와 레이아웃

Views and Layouts

- View : TextView, Button, ImageView, ListView, EditText, etc.
- Layout : LinearLayout, RelativeLayout, FrameLayout, AbsoluteLayout, etc.
- View Attribute
 - layout_width, layout_height, background, visibility, id
 - <http://code.google.com/intl/ko-KR/android/reference/android/view/View.html>

뷰와 레이아웃

Views and Layouts

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout
3.     xmlns:android="http://schemas.android.com/apk/res/android"
4.     android:orientation="vertical"
5.     android:layout_width="fill_parent"
6.     android:layout_height="fill_parent"
7.     android:background="#FF888888"
8. >
9. <TextView
10.     android:layout_width="fill_parent"
11.     android:layout_height="wrap_content"
12.     android:text="파랑"
13.     android:background="#FF0000FF"
14. </LinearLayout>
```

뷰와 레이아웃

Views and Layouts

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     android:orientation="vertical"
4.     android:layout_width="fill_parent"
5.     android:layout_height="fill_parent"
6. >
7. <TextView
8.     android:layout_width="fill_parent"
9.     android:layout_height="wrap_content"
10.    android:text="Red"
11.    android:background="#FFFFFF0000"
12. />
13. <TextView
14.     android:layout_width="wrap_content"
15.     android:layout_height="50dp"
16.     android:text="Green"
17.     android:background="#FF00FF00"
18. />
19. <TextView
20.     android:layout_width="fill_parent"
21.     android:layout_height="wrap_content"
22.     android:text="Blue"
23.     android:background="#FF0000FF"
24. />
25. </LinearLayout>
```



안드로이드 강좌 5 - XML에서의 TextView, ImageView, LinearLayout

- <http://www.mobileplace.co.kr/2198>

TextView attribute

- 그 전의 예에서 (In the previous example)
- TextView
 - width – wrap_content
 - height – 50dp
 - android:visibility="invisible"
 - android:visibility="gone"

TextView attribute

1. <TextView
2. android:layout_width="fill_parent"
3. android:layout_height="fill_parent"
4. android:text = "Hello, Oman"
5. android:textColor = "#FF0000FF"
6. android:textSize = "30sp"
7. android:textStyle = "italic"
8. android:gravity = "right|center_vertical"
9. android:singleLine = "true"
10. />



TextView attribute

- `android:text`
- `android:textColor`
- `android:textSize`
- `android:textStyle` – bold, italic, etc.
- `android:gravity` – top, bottom, left, right, center, center_vertical, center_horizontal
- `android:singleLine`

ImageView

1. `<ImageView`
2. `android:layout_width="fill_parent"`
3. `android:layout_height="fill_parent"`
4. `android:src="@drawable/icon"`
5. `android:scaleType="center"`
6. `/>`
7. `android:scaleType="fillCenter"`
8. For your pic, copy it to res/drawable

LinearLayout

```
1. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2.     android:orientation="vertical"
3.     android:layout_width="fill_parent"
4.     android:layout_height="fill_parent"
5. >
6. <TextView
7.     android:layout_width="fill_parent"
8.     android:layout_height="odp"
9.     android:layout_weight="1"
10.    android:background="#FF880000"
11. />
12. <TextView
13.     android:layout_width="fill_parent"
14.     android:layout_height="odp"
15.     android:layout_weight="1"
16.     android:background="#FF008800"
17. />
18. <TextView
19.     android:layout_width="fill_parent"
20.     android:layout_height="odp"
21.     android:layout_weight="1"
22.     android:background="#FF000088"
23. />
24. </LinearLayout>
```



LinearLayout

- 그 전의 예에서 (In the previous example,)
- `orientation` → `vertical`
- `android:layout_width="00"`
- `android:layout_height="fill_parent"`
- `layout_weight` → `1, 2, 1`

안드로이드 강좌 6 - Java 코드 (Code)에서 뷰(View) 다루기

- <http://www.mobileplace.co.kr/2353>

Java Code and View

- id
 1. `<TextView`
 2. `android:id="@+id/text"`
 3. `android:layout_width="fill_parent"`
 4. `android:layout_height="wrap_content"`
 5. `/>`
- Code
 1. `TextView t = (TextView)findViewById(R.id.text);`
 2. `t.setText("Hello");`
 3. `t.setBackgroundColor(0xFFFF0000);`
 4. `t.setGravity(Gravity.LEFT);`



View.setOnClickListener

1. Button button =
(Button)findViewById(R.id.button);
2. button.setOnClickListener(new
View.OnClickListener() {
3. public void onClick(View v) {
4. //Code
5. }
6. });

myactivity.xml

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     android:orientation="vertical"
4.     android:layout_width="fill_parent"
5.     android:layout_height="fill_parent"
6.     >
7. <TextView
8.     android:id="@+id/text"
9.     android:layout_width="fill_parent"
10.    android:layout_height="wrap_content"
11.    android:text=""
12.    />
13. <Button
14.     android:id="@+id/button"
15.     android:layout_width="fill_parent"
16.     android:layout_height="wrap_content"
17.     android:text="버튼"
18.     />
19. </LinearLayout>
```

MyActivity.java

```
1. package dsu.android;
2. import android.app.Activity;
3. import android.os.Bundle;
4. import android.view.Gravity;
5. import android.view.View;
6. import android.widget.Button;
7. import android.widget.TextView;
8. public class MyActivity extends Activity {
9.     @Override
10.    protected void onCreate(Bundle savedInstanceState) {
11.        super.onCreate(savedInstanceState);
12.        setContentView(R.layout.myactivity);
13.        Button button = (Button)findViewById(R.id.button);
14.        button.setOnClickListener(new View.OnClickListener() {
15.            public void onClick(View v) {
16.                TextView t = (TextView)findViewById(R.id.text);
17.                t.setText("Clicked~");
18.                t.setBackgroundColor(oxFFFF0000);
19.                t.setGravity(Gravity.LEFT); }
20.        });
21.    }
22. }
```




한백 전자 교육의 구글 안드로이드 예제들 실습