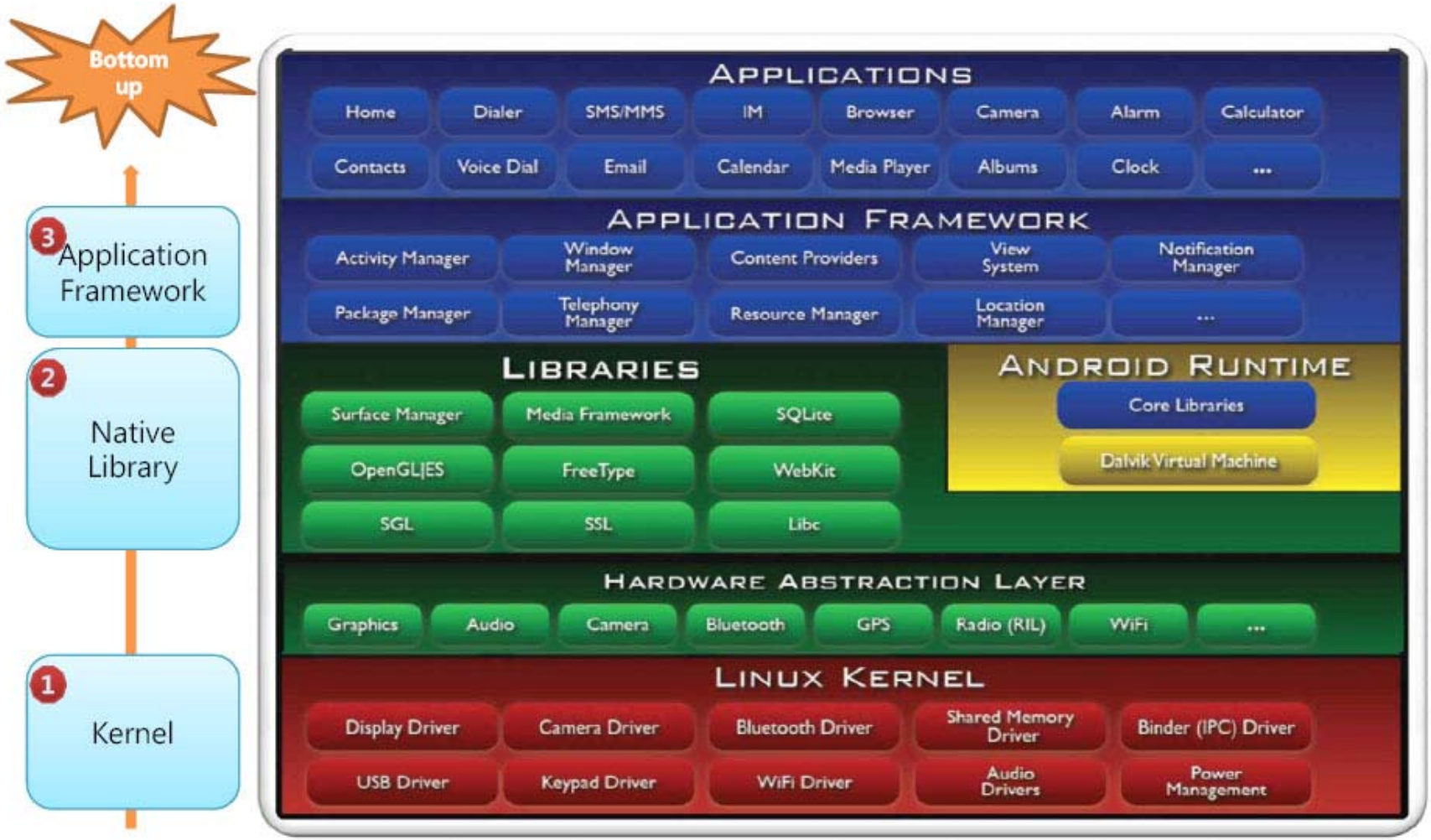
군산대학교
KUNSAN NATIONAL UNIVERSITY

Android Application Framework

모바일 시스템SW

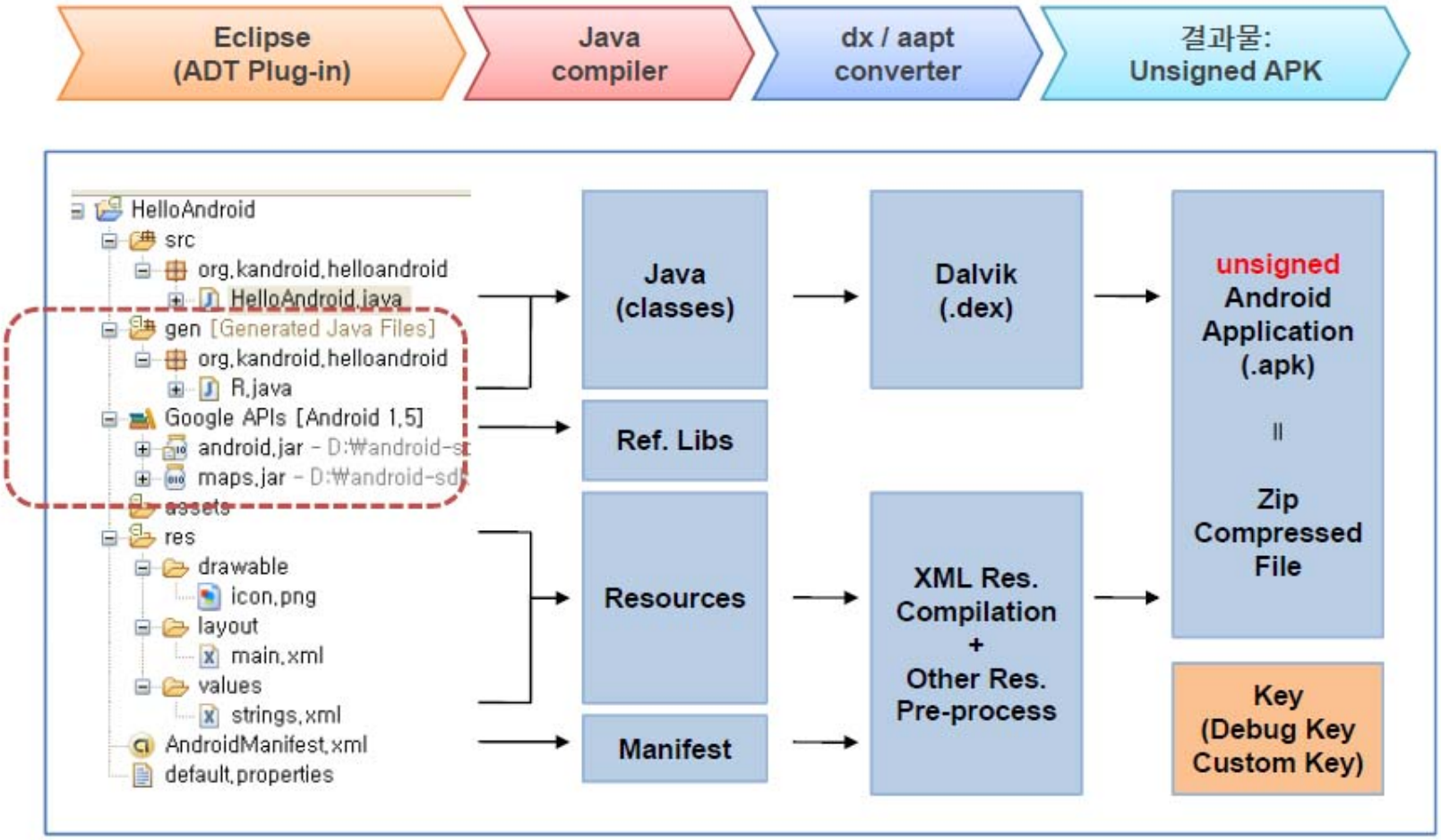
남 광 우

주요 강의자료 인용 : www.kandroid.org 양정수(yangjeongsoo@gmail.com)


Bottom up
3
Application Framework
2
Native Library
1
Kernel
APPLICATIONS
Home
Dialer
SMS/MMS
IM
Browser
Camera
Alarm
Calculator
Contacts
Voice Dial
Email
Calendar
Media Player
Albums
Clock
...
APPLICATION FRAMEWORK
Activity Manager
Window Manager
Content Providers
View System
Notification Manager
Package Manager
Telephony Manager
Resource Manager
Location Manager
...
LIBRARIES
Surface Manager
Media Framework
SQLite
OpenGL|ES
FreeType
WebKit
SGL
SSL
Libc
ANDROID RUNTIME
Core Libraries
Dalvik Virtual Machine
HARDWARE ABSTRACTION LAYER
Graphics
Audio
Camera
Bluetooth
GPS
Radio (RIL)
WiFi
...
LINUX KERNEL
Display Driver
Camera Driver
Bluetooth Driver
Shared Memory Driver
Binder (IPC) Driver
USB Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

안드로이드 개발 프로세스(Android 1.5 이후 Feature)

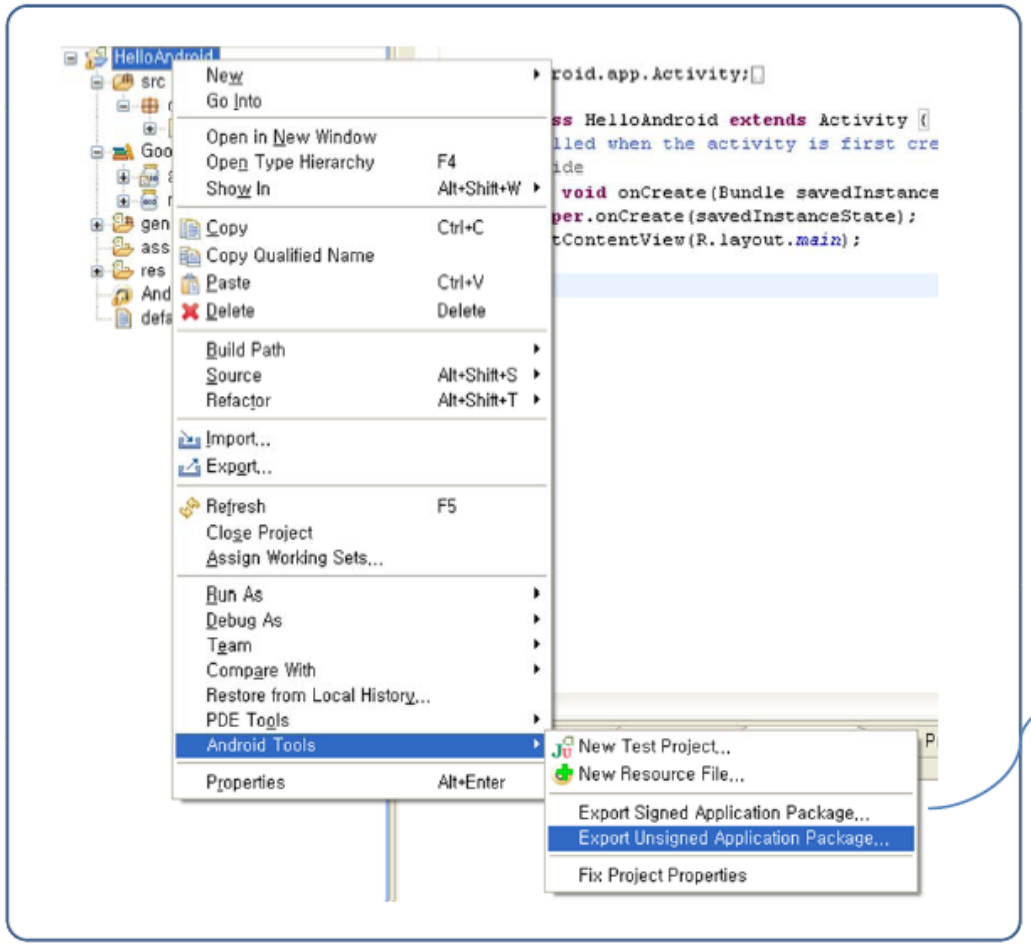
❖ 안드로이드 애플리케이션 빌드 프로세스(1)

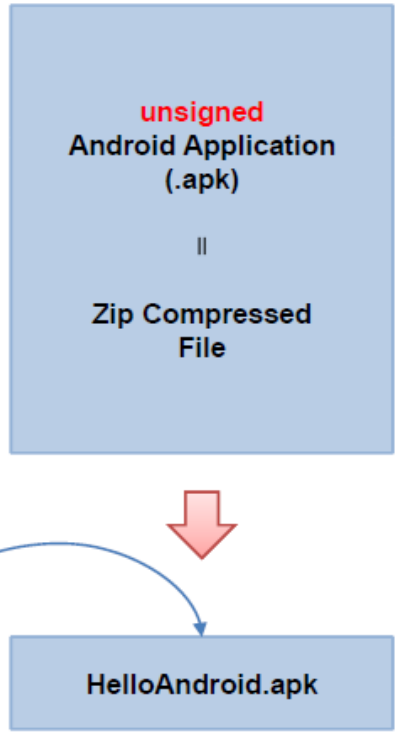


안드로이드 개발 프로세스(Android 1.5 이후 Feature)

안드로이드 개발 프로세스 : 배포하기

❖ android app. 만들기(apk 만들기)





안드로이드 개발 프로세스 : 배포하기

❖ Sign하고 adb를 이용하여 설치하기

adb = android debug bridge
(설명 보기)

```
>adb.exe install HelloAndroid.apk
424 KB/s (6789 bytes in 0.015s)
        pkg: /data/local/tmp/HelloAndroid.apk
Failure [INSTALL_PARSE_FAILED_NO_CERTIFICATES]
```

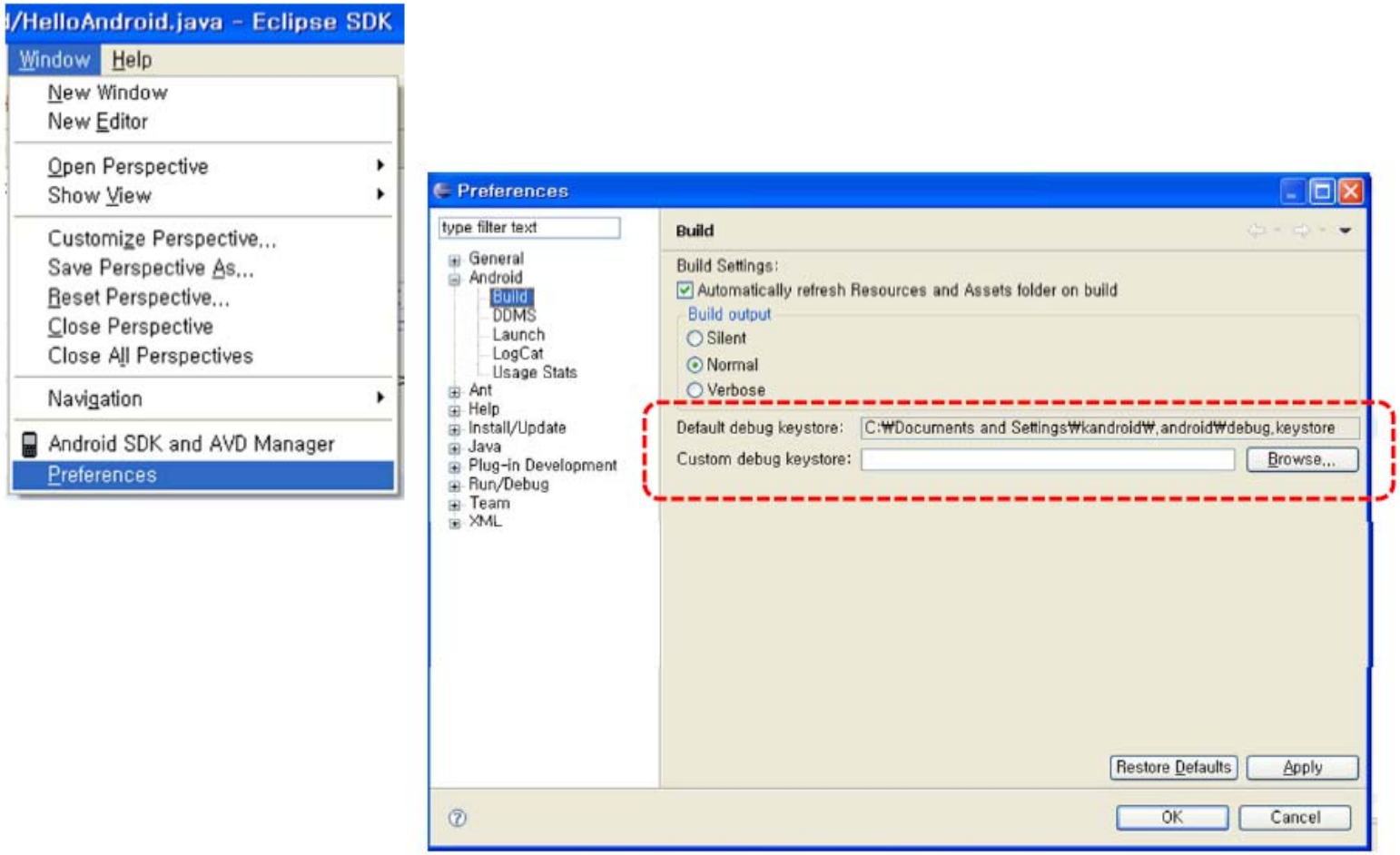
```
> jarsigner.exe -verbose -keystore "debug.keystore" HelloAndroid.apk androiddebugkey
Enter Passphrase for keystore: android
   adding: META-INF/MANIFEST.MF
   adding: META-INF/ANDROIDD.SF
   adding: META-INF/ANDROIDD.RSA
  signing: res/drawable/icon.png
  signing: res/layout/main.xml
  signing: AndroidManifest.xml
  signing: resources.arsc
  signing: classes.dex
```

다음페이지에 설명됨

```
>adb.exe install HelloAndroid.apk
526 KB/s (8428 bytes in 0.015s)
        pkg: /data/local/tmp/HelloAndroid.apk
Success
```

안드로이드 개발 프로세스 : 배포

- **debug.keysotre**



안드로이드 개발 프로세스 : 배포

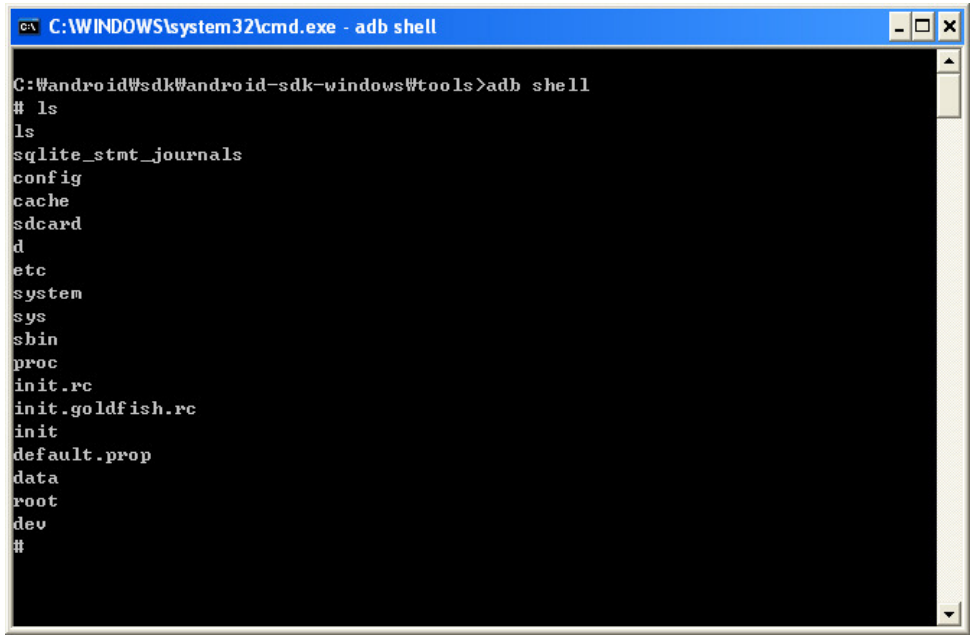
❖ uninstall

```
>adb.exe uninstall org.kandroid.helloandroid
Success
```

안드로이드 개발 프로세스 : 배포

❖ adb 란?

- ✓ adb = android debug brige
  - ADB 안드로이드의 전용 명령어 인터페이스 서비스이다. 명령어 기반으로 애플리케이션 인스톨, 언인스톨, 파일 추가 및 삭제, 리눅스 셸 접속 등 다양한 오퍼레이션을 수행할 수 있다.
  - ADB 실행 파일은 안드로이드 SDK home 밑의 Tools 폴더에 존재한다.
  - 'C:\android\sdk\android-sdk-windows\tools' 가 시스템 경로에 추가되었다면, CMD 명령으로 프롬프트 창을 열고 adb shell 이라고 명령을 내려본다.




```
> adb.exe shell
# cd /data/app
# ls -l

-rw-r--r-- system   system       8336 2009-09-27 15:50 org.kandroid.helloandroid.apk
```

안드로이드 개발 프로세스 : apk 파일의 안쪽 보기

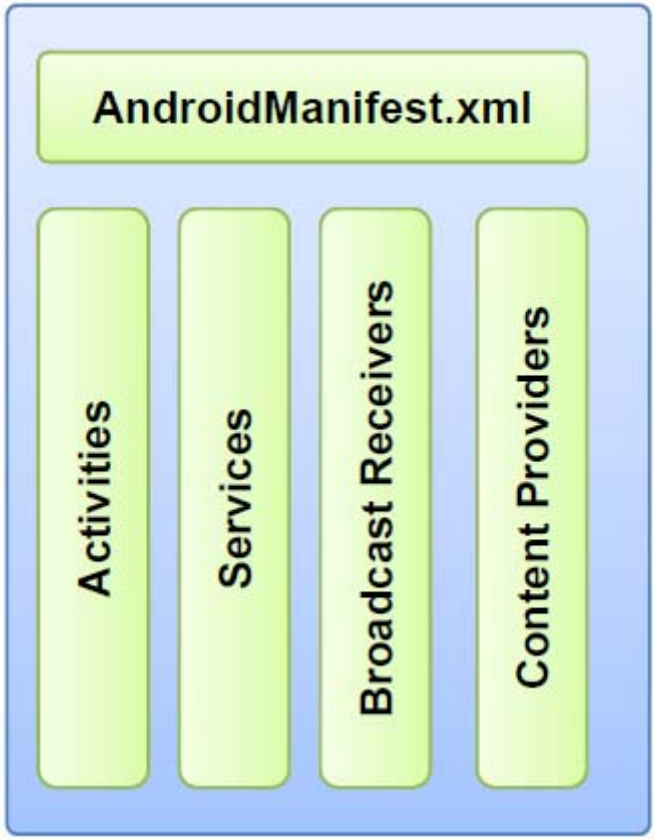
- **안드로이드 마켓**
  - ✓ 공식 안드로이드 마켓 : http://www.android.com/market/
  - ✓ 독립마켓 : http://slideme.org

- **android app. 다운로드**
  - ✓ slideme에서 application을 하나 다운로드
  - ✓ slideme => load monitor 다운로드
  - ✓ *.apk에서 확장명을 zip으로 변경하고 풀기

이름	수정한 날짜	유형	크기
META-INF	2010-03-15 오후 ...	파일 폴더	
res	2010-03-15 오후 ...	파일 폴더	
AndroidManifest.xml	2010-03-12 오전 ...	XML 문서	3KB
classes.dex	2010-03-12 오전 ...	DEX 파일	12KB
resources.arsc	2010-03-12 오전 ...	ARSC 파일	4KB

Application Fundamental

❖ 주요 구성 요소



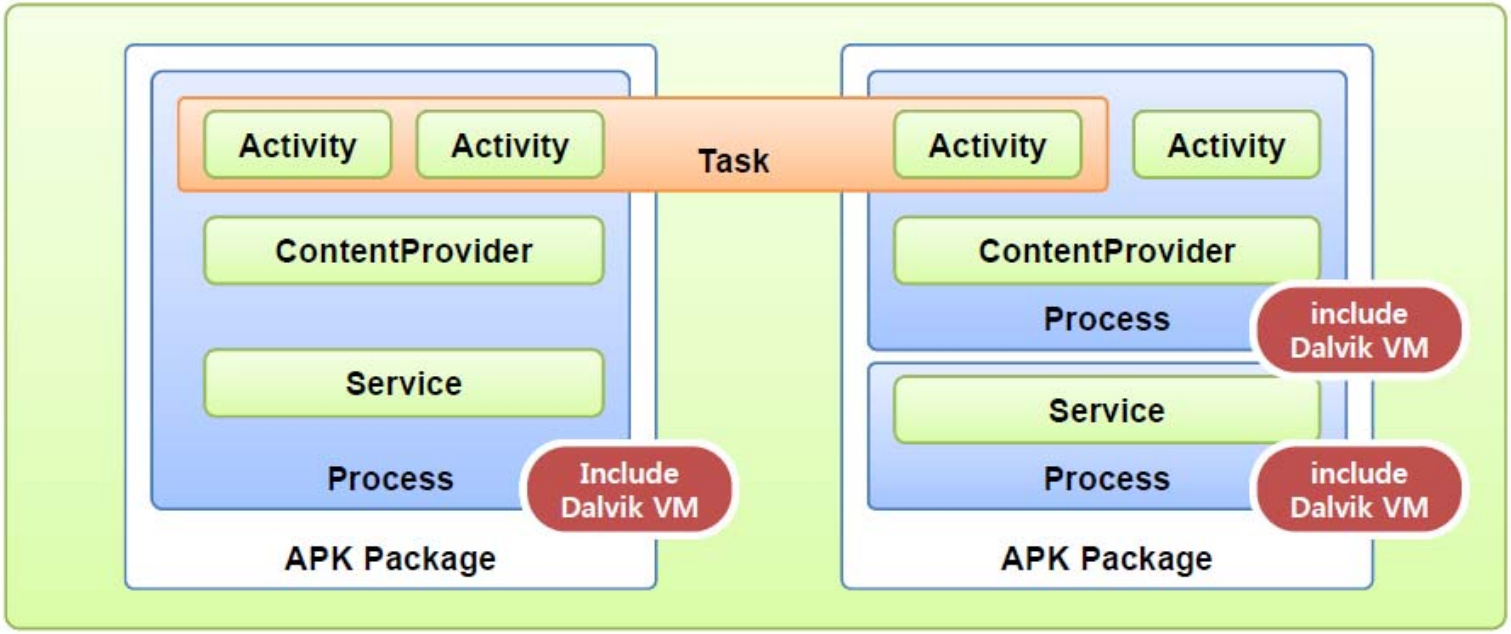
Application Fundamental

❖ 구조

Application Building Block

- AndroidManifest.xml
- Activity [User Interaction] :
- ContentProvider [Data Provider]
- Service [Service Provider]
- BroadcastReceiver

Intent : Bundle of Informations
- Explicit Intent : Call Class
- Implicit Intent : IntentFilter
  - Action, Data, Category
  - Declared at AndroidManifest.xml



Application Fundamental

❖ manifest file

✓ 디렉토리 root에 AndroidManifest.xml 로 생성

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="org.kandroid.helloandroid"
      android:versionCode="1"
      android:versionName="1.0.0">

  <application android:icon="@drawable/icon"
          android:label="@string/app_name">

    <activity android:name=".HelloAndroid"
              android:label="@string/app_name">

      <intent-filter>
         <action android:name="android.intent.action.MAIN" />
         <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>

    </activity>
  </application>
</manifest>
```

Application Fundamental

❖ Activity

- ✓ Activity는 하나의 가상의 사용자 인터페이스에 대한 표현이다.

- Can be faceless
- Can be in a floating window
- Can return a value
- Can be embedded

Application Fundamental

```
package com.android.myactivity;

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

public class MyActivity extends Activity
{
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.main);

    }
}
```

Application Fundamental

❖ Service


```
package com.android.myservice;

import android.app.Service;

public class MyService extends Service
{
    public void onCreate() {
        Thread st = new Thread() {
            void run() { /* ... */ }
        };
        st.start();
    }
    public void onDestroy() {
        /* ... */
    }
}
```

Application Fundamental

❖ BroadcastReceiver

```
package org.kandroid.helloandroid;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;

public class MyReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context arg0, Intent arg1) {

    }

}
```

Application Fundamental

❖ Content Provider

- Enables sharing of data across applications

  Examples : address book, photo gallery, etc.

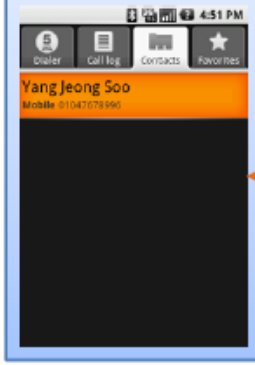
- Provides uniform API for :

  Querying (return a Cursor)

  delete, update, and insert rows

- Content is represented by URI and MIME type

다양한 Content Provider 가 존재함.



CalendarProvider
ContactsProvider
DownloadProvider
DrmProvider
GoogleContactsProvider
GoogleSubscribedFeedsProvider
ImProvider
MediaProvider
TelephonyProvider

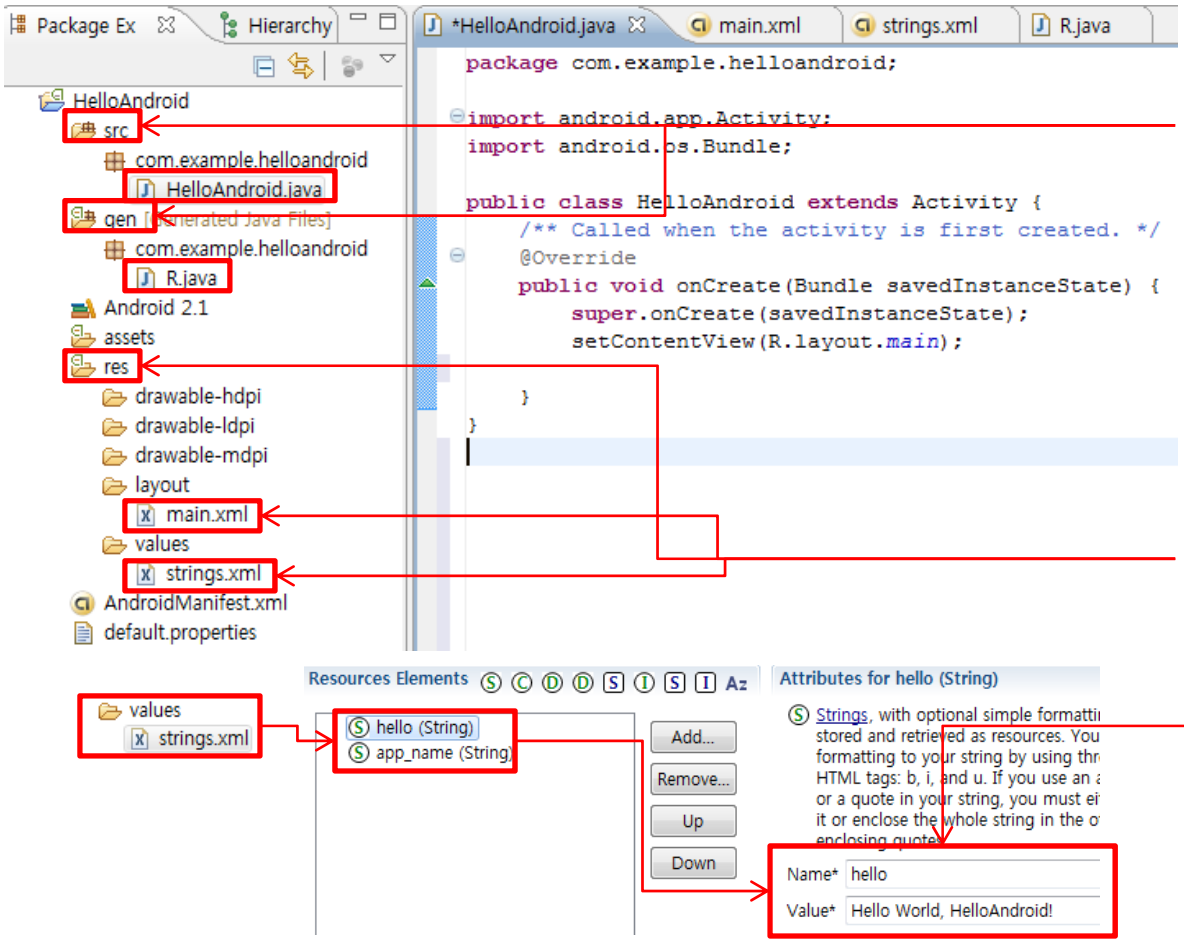
Application Fundamental

```
package org.kandroid.helloandroid;

import android.content.*;
import android.database.Cursor;
import android.net.Uri;

public class MyProvider extends ContentProvider {

    @Override public int delete(Uri arg0, String arg1, String[] arg2)
                            { return 0; }

    @Override public String getType(Uri arg0) {return null;}

    @Override public Uri insert(Uri arg0, ContentValues arg1)
                            { return null;}
    @Override public boolean onCreate() {return false;}

    @Override public Cursor query(Uri arg0, String[] arg1, String arg2,
                            String[] arg3, String arg4) {return null;}

    @Override public int update(Uri arg0, ContentValues arg1, String arg2,
                            String[] arg3) {return 0;}
}
```

예 1: Hello World

1. “Hello Android” (1)

- Project 구성요소에 대한 설명



src 폴더는 생성시 입력한 Package(폴더) 안에 java소스파일이 있는 구조로 되어있으며 밑에 gen폴더 역시 같은 Package로 구성되어 각각 다른폴더지만 묵시적으로 서로 연결된 상태로 볼 수 있다. Gen 폴더는 자동적으로 리소스(res)를 src쪽에서 쉽게 사용할 수 있도록 정의한 R.java파일을 생성한다.

res폴더는 리소스파일들을 관리하는데 Drawable은 이미지에 관한 리소스를 담고 있으며 Layout은 화면에 표시할 객체들을에 대한 정보를 가지는 main.xml 파일을 가지고 있다. Values에는 string.xml을 가지고 있는데, 문자에 대한 정보 값들을 가지고 있으며 프로젝트 생성시 자동으로 찍히는 “hello world, HelloAndroid!” 란 문자는 여기에 저장되어 있다. 여기에 value 값을 수정하여 화면에 표시될 문자 값을 변경할 수 있다.

XML로 Layout을 지정하는 이유는 GUI를 프로그램에서 좀더 쉽게 생성하고 관리가 가능하기 때문이다.

예 1 : Hello World

1. 실습 "Hello Android" (2)

1. 이제는 소스코드를 입력하여 Text를 화면에 직접 뛰워보자!
2. 먼저 setContentView(R.layout.main); 부분을 주석/삭제 하여 아무것도 프로그램에 뜨지 않게 한다.

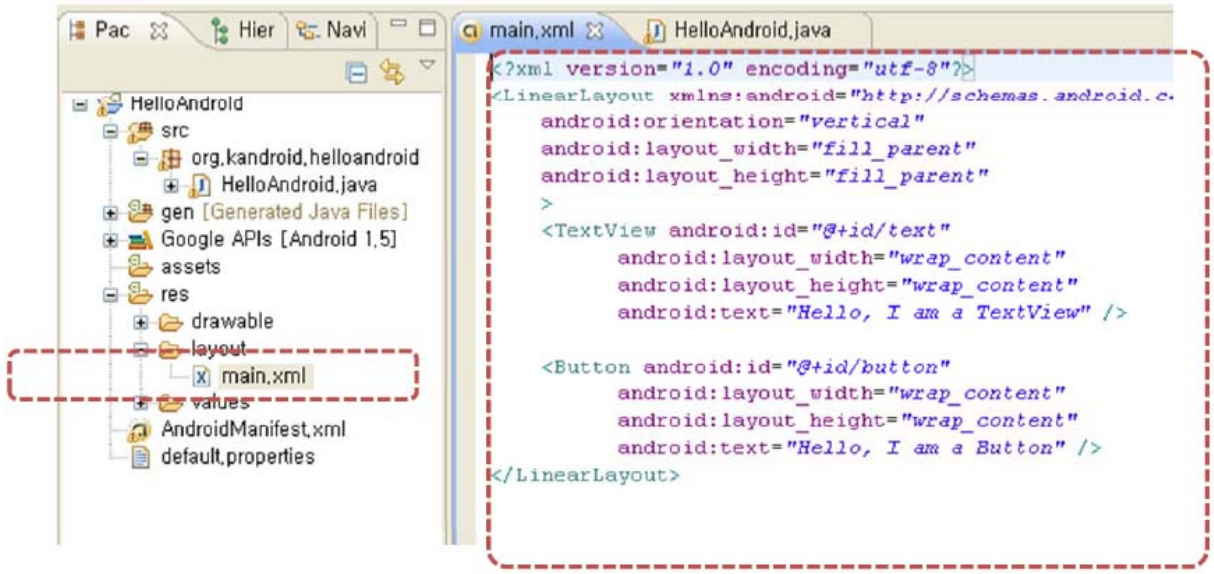
*HelloAndroid.java | main.xml | strings.xml

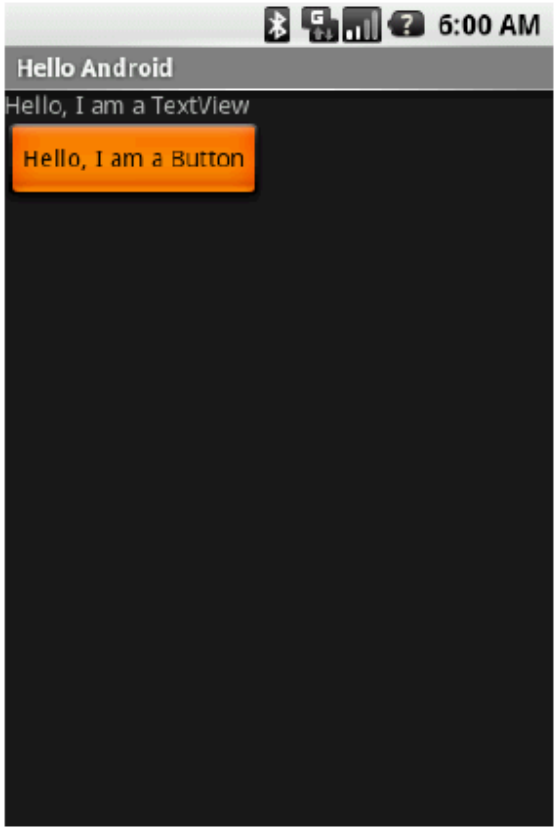
```
package com.example.helloandroid;

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

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
//        setContentView(R.layout.main);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
    }
}
```

3. TextView객체를 사용하기 위해 다음과 같이 TextView관련 라이브러리를 import 한다.
4. TextView tv = new TextView(this); 로 동적으로 'tv' 란 TextView객체를 생성한다.
5. tv.setText("Hello, Android");로 tv 객체 안에 setText함수를 통해 "Hello Android" 란 문자를 입력받는다.
6. 마지막으로 setContentView(tv);로 tv 객체를 화면에 보일 수 있게 설정한다.
7. Run을 통해 가상장치로 실행하면 다음과 같이 화면상에 입력한 문자가 뜨는것을 확인할 수 있다.

예 2: Layout 이용하기

❖ **Layout을 이용하여 버튼과 Text를 함께 보이는 프로그램을 작성하시오**





예 2: Layout 이용하기

❖ 프로그램 구현 소스

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
       xmlns:android="http://schemas.android.com/apk
/res/android"
       android:layout_width="fill_parent"
       android:layout_height="fill_parent"
       android:orientation="vertical" >

    <TextView android:id="@+id/text"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:text="Hello, I am a TextView" />

    <Button android:id="@+id/button"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:text="Hello, I am a Button" />

</LinearLayout>
```

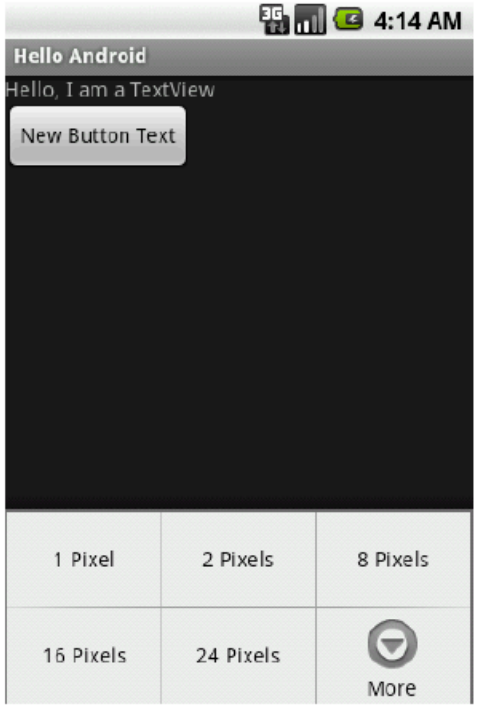
```
package org.kandroid.helloandroid;

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

public class HelloAndroid extends Activity {

    /** Called when the activity is first created. */

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

        setContentView(R.layout.main);

    }

}
```

예 3: Menu 이용하기

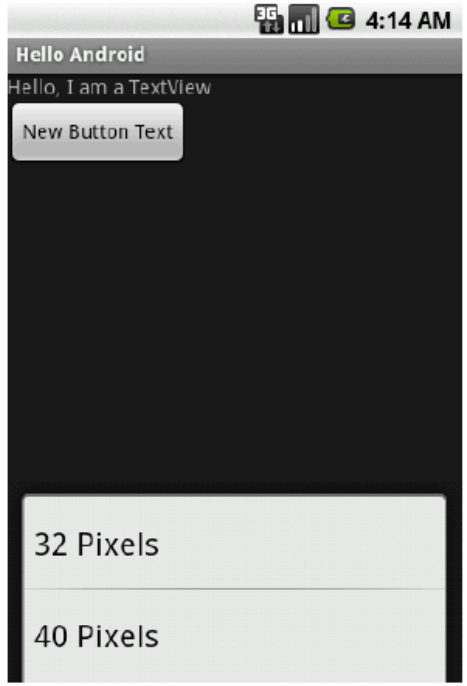
❖ **Hello World 프로그램에 Option Menu를 추가하시오**

```
package org.kandroid.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.widget.Button;

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button myButton = (Button) findViewById(R.id.button);
        myButton.setText("New Button Text");

    }
    public boolean onCreateOptionsMenu(Menu menu)
    {
            menu.add(Menu.NONE, 1,  Menu.NONE, "1 Pixel");
            menu.add(Menu.NONE, 2,  Menu.NONE, "2 Pixels");
            menu.add(Menu.NONE, 3,  Menu.NONE, "8 Pixels");
            menu.add(Menu.NONE, 4,  Menu.NONE, "16 Pixels");
            menu.add(Menu.NONE, 5,  Menu.NONE, "24 Pixels");
            menu.add(Menu.NONE, 6,  Menu.NONE, "32 Pixels");
            menu.add(Menu.NONE, 7,  Menu.NONE, "40 Pixels");
            return(super.onCreateOptionsMenu(menu));
    }
}
```

Icon Menu

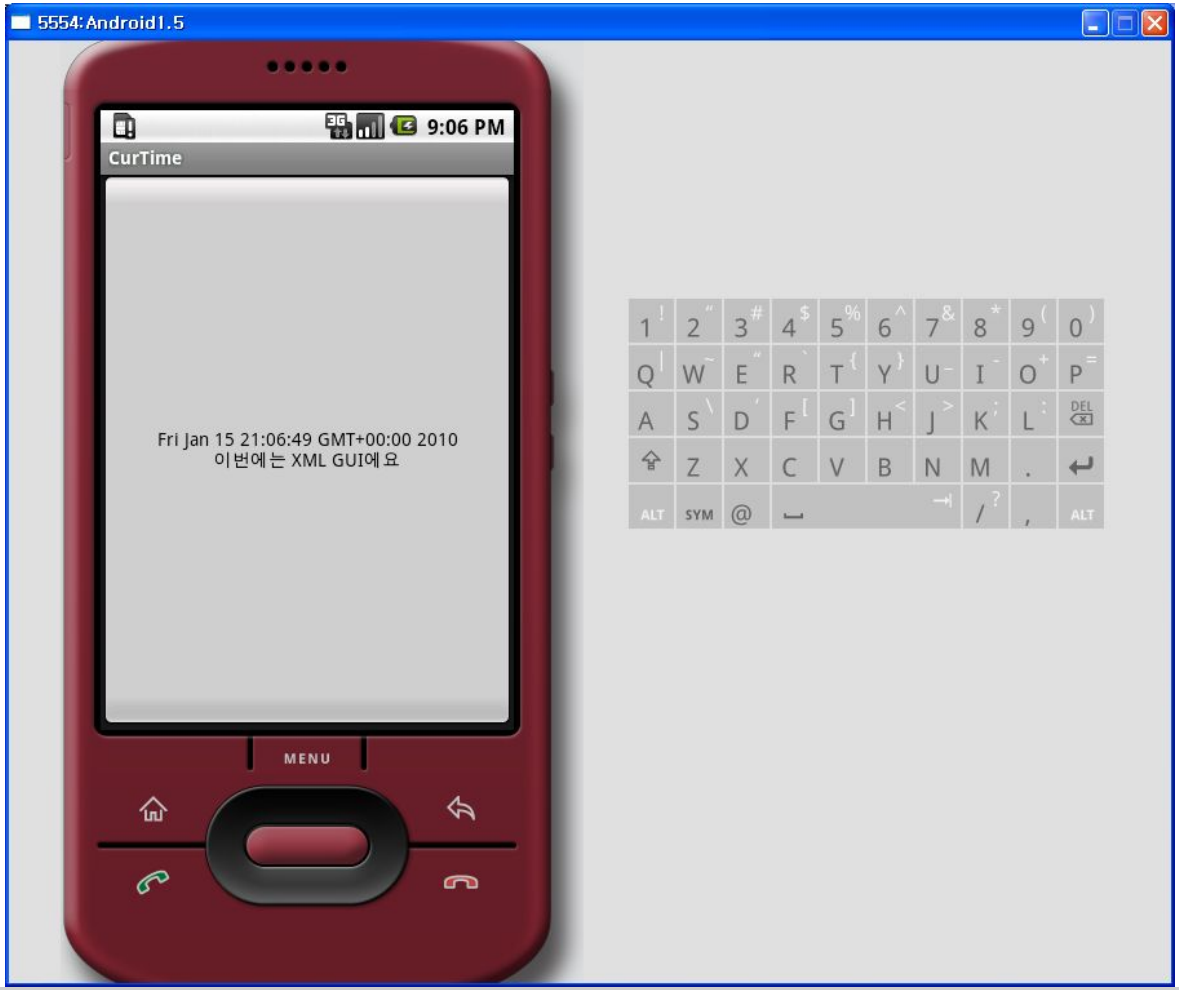


Expanded Menu



예 4 : 버튼 눌러 시간 보기

❖ 버튼을 눌러 시간을 표시하는 프로그램을 작성하시오.



예 4 : 버튼 눌러 시간 보기

❖ main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- xml 에서 버튼부분의 gui를 관리한다 -->
<Button
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/button"
android:text=""
android:layout_width="fill_parent"
android:layout_height="fill_parent"/>
```

예 4 : 버튼 눌러 시간 보기

❖ HelloButton.java

```
public class HelloButton extends Activity implements View.OnClickListener{
        Button btn;
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        //자바코드와 xml 레이아웃 연결
        //ContentView를 main.xml로 설정한다는 의미
        setContentView(R.layout.main);
        //xml 에서 버튼의 정보를 가져옴
        //findView() 는 view에서 객체를 찾는것
        btn = (Button)findViewById(R.id.button);
        btn.setOnClickListener(this);
        //시간 갱신하여 화면에 출력
        updateTime();
}
public void onClick(View view){
        updateTime();
}
//시간을 갱신하여 화면에 출력하는 메소드
private void updateTime(){
        btn.setText(new Date().toString()
        +"\n이번에는 XML GUI에요");
}
}
```

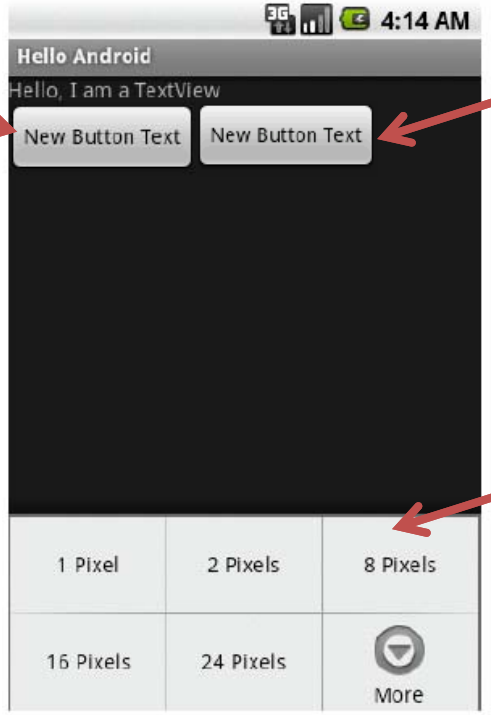
```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
// Handle item selection
                switch (item.getItemId()) {
                case R.id.new_game:

                case R.id.help:
                                showHelp();
                                return true;
                default:
return super.onOptionsItemSelected(item);
}
}
```

실습 : options menu를 지원하는 버튼 눌러 시간 보기

❖ **구현 프로그램**

- ✓ 목표 : 예2, 예3, 예4 의 프로그램을 합하는 프로그램 구현하고 실제 에뮬과 폰에 설치
- ✓ 구현 내용

필수 1: 이 버튼을 누르면 현재 시간 표시



옵션1 : 이 버튼(btn2)을 누르면 누를때마다 1씩 증가

옵션2 : 누를때마다 btn2의 숫자를 해당 숫자만큼씩 증가

실습 : options menu를 지원하는 버튼 눌러 시간 보기

- **구현 프로그램 설치**
  - ✓ 필수 2 : 프로그램을 apk로 만들어 인증하고,
    adb를 이용하여 에뮬레이터에 설치하고 실행시킬 것
  - ✓ 옵션 3 : 실제 안드로이드폰에 설치하고 실행해보기