



CHAP 10. 액티비티와 인텐트(1)

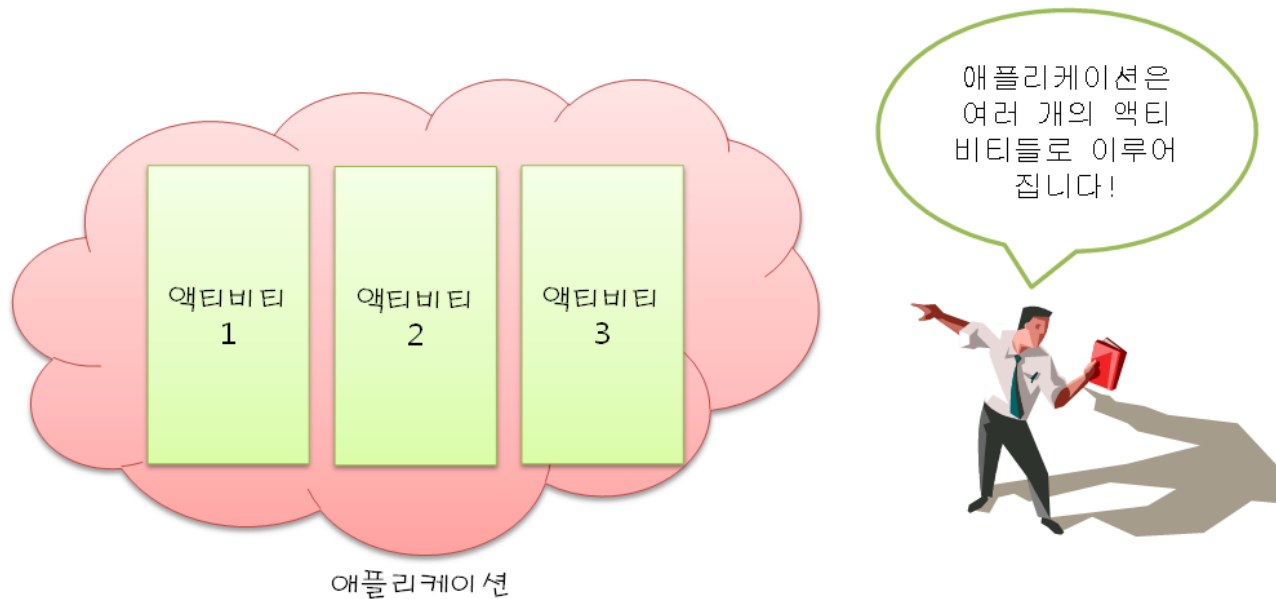
1

4가지의 중요한 개념

- 애플리케이션(application)
- 액티비티(activities)
- 액티비티 스택(activity stack)
- 태스크(task)

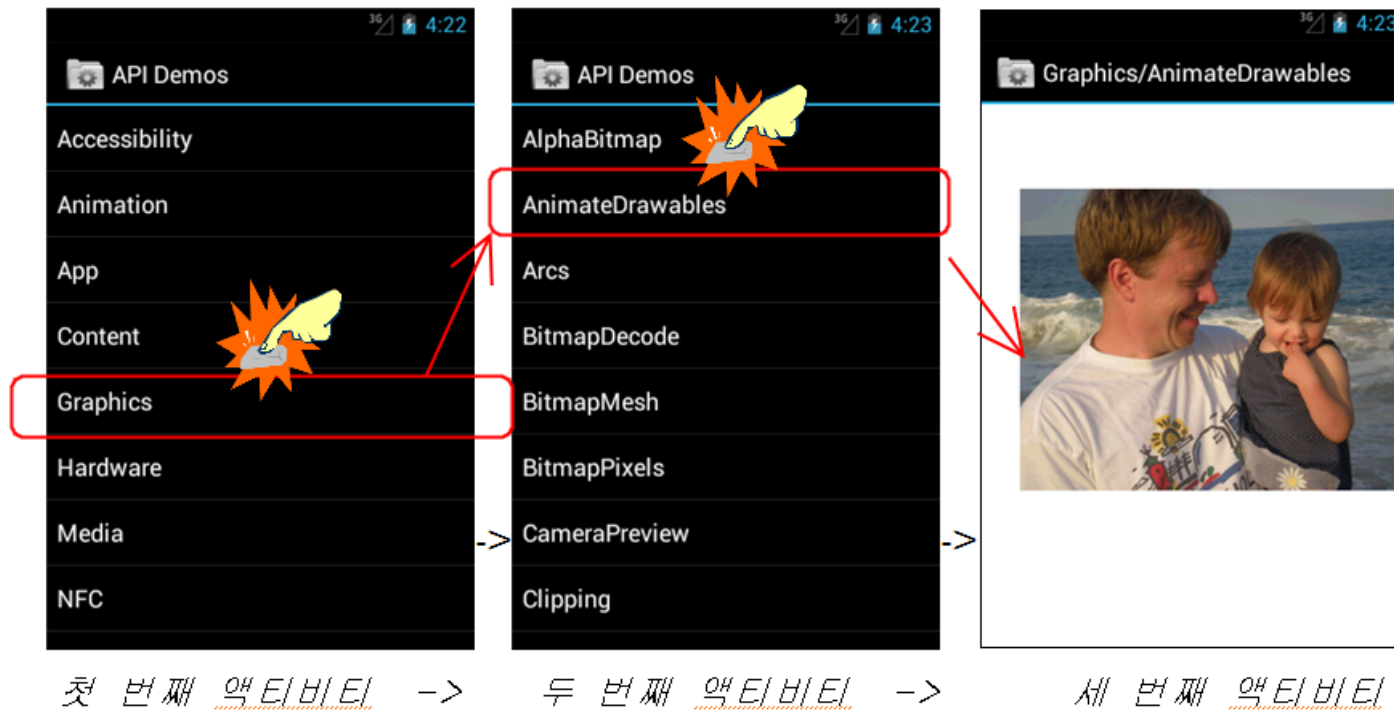
애플리케이션

- 한 개 이상의 액티비티들로 구성된다.
- 액티비티들은 애플리케이션 안에서 느슨하게 묶여 있다.



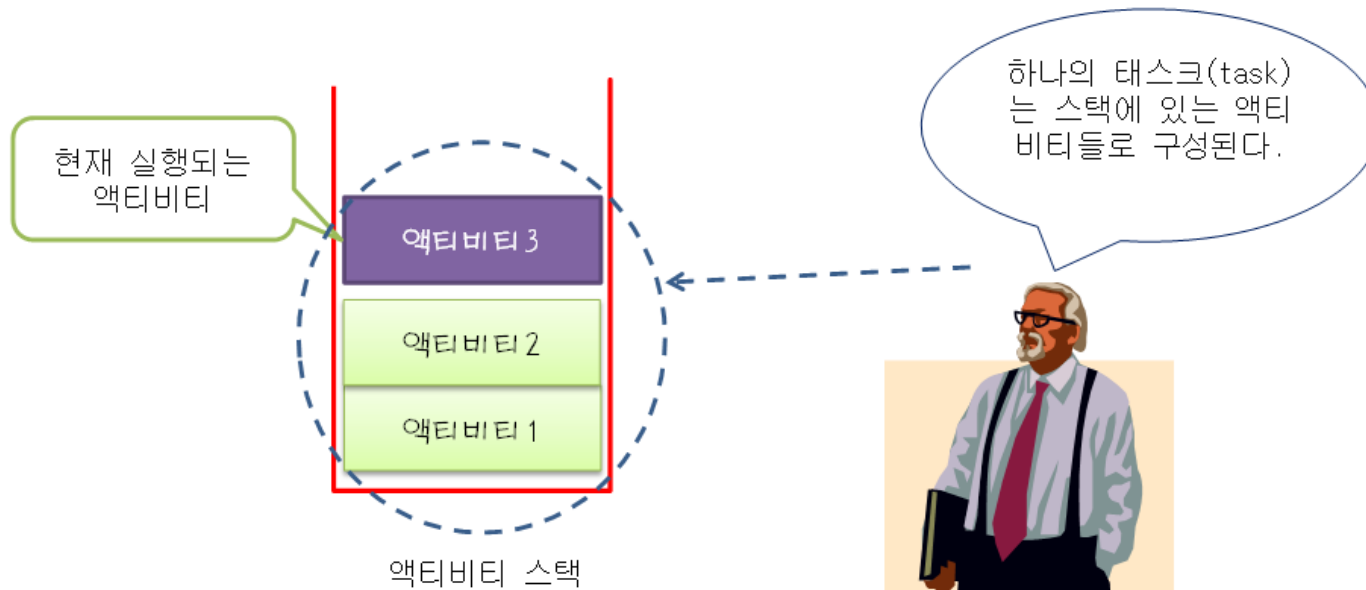
액티비티

○ 애플리케이션을 구성하는 빌딩 블록



태스크

○ 스택에 있는 액티비티

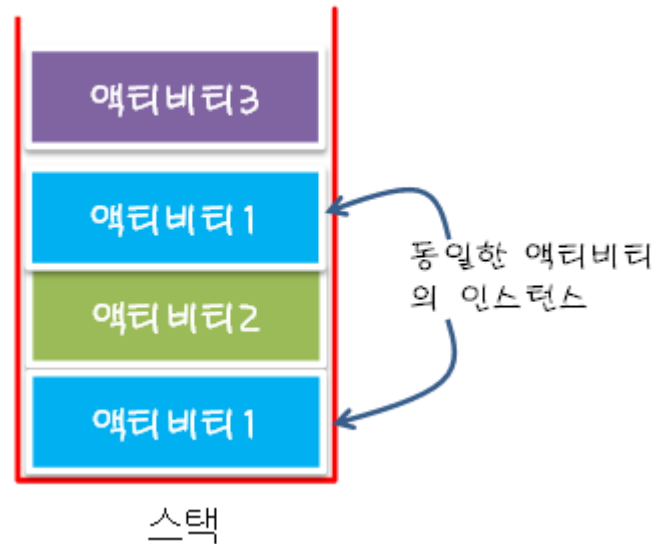


액티비티 스택

- Back 키를 누르면 현재 액티비티를 제거하고 이전 액티비티로 되돌아 간다.
- 사용자가 방문한 액티비티들은 어딘가에 기억



다음과 같은 상태도 가능하다



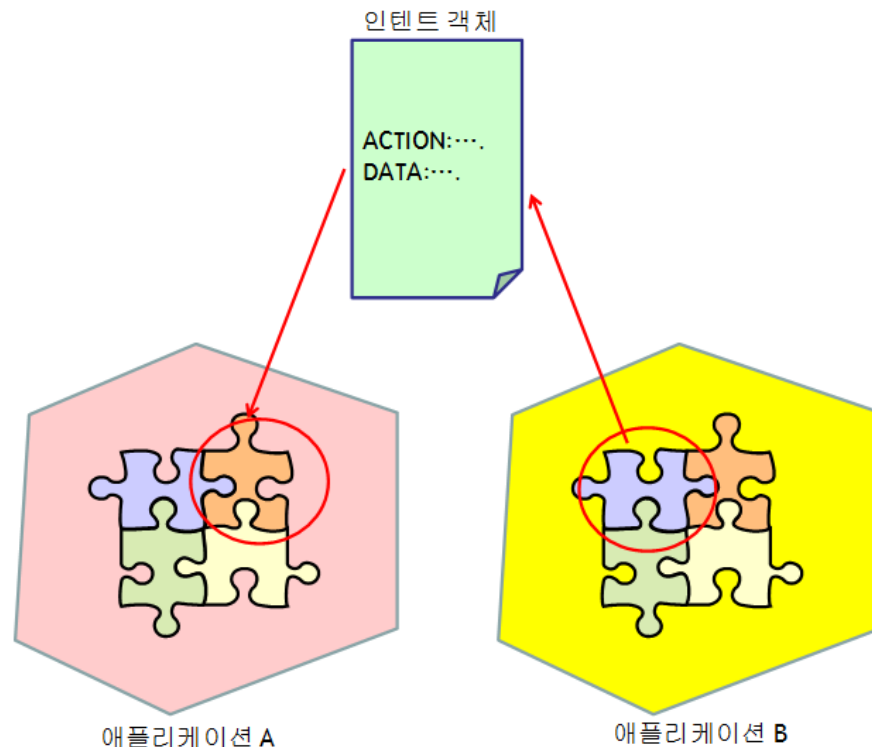
인텐트

- 각각의 화면은 별도의 액티비티로 구현된다.
- 하나의 액티비티(화면)에서 다른 액티비티(화면)로 전환하려면 어떻게 하여야 하는가?



인텐트

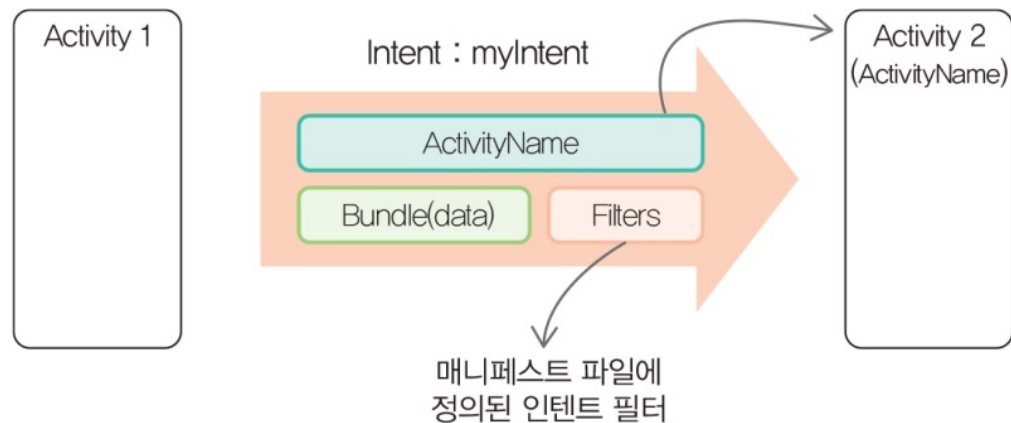
- 다른 액티비티를 시작하려면 액티비티의 실행에 필요한 여러 가지 정보들을 보내주어야 한다.
- 정보를 인텐트에 실어서 보낸다.



인텐트의 종류

명시적 인텐트(explicit intent)

- “애플리케이션 A의 컴포넌트 B를 구동시켜라”와 같이 명확하게 지정



암시적 인텐트(implicit intent)

- “지도를 보여줄 수 있는 컴포넌트이면 어떤 것이라도 좋다”

명시적인 인텐트

- 실행하고자 하는 액티비티의 이름을 적어 준다.

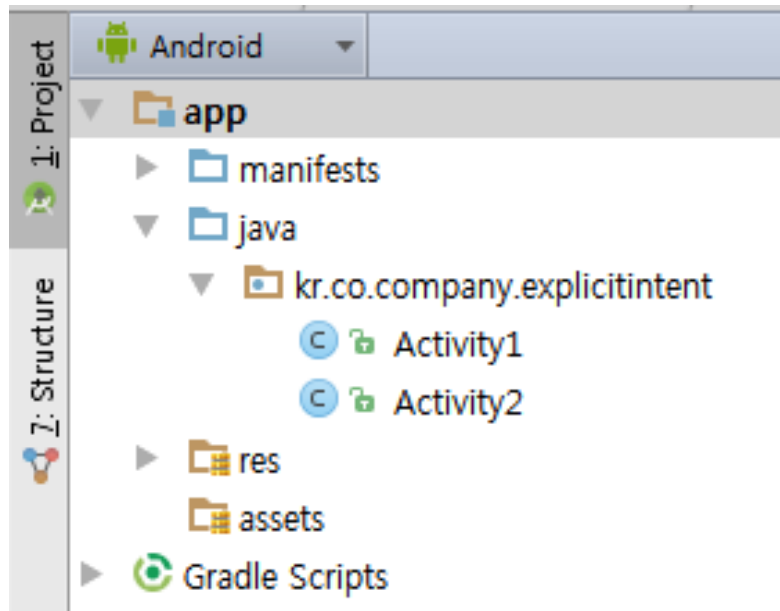
```
Intent intent = new Intent(this, NextActivity.class);  
startActivity(intent);
```

intent 객체에 기술된 액티비티를 시작한다.

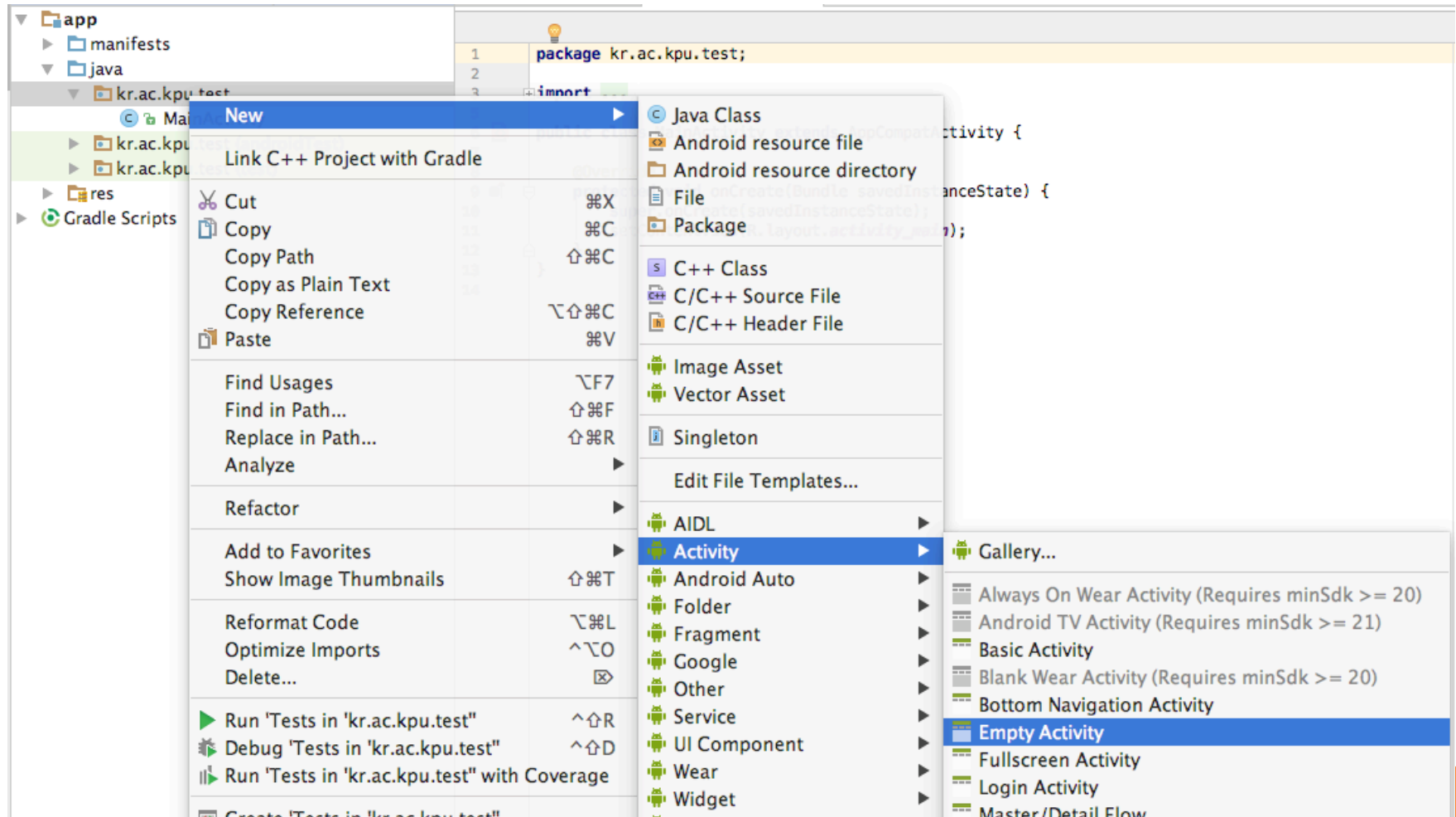
인텐트 객체에 실행하고 싶은 액티비티의 클래스 이름인 NextActivity를 지정한다.

명시적인 인텐트 예제

- 여기서 두 개의 액티비티로 이루어진 애플리케이션을 작성하여 보자. 첫 번째 액티비티는 Activity1, 두 번째 액티비티는 Activity2라고 하자.



ANDROIDSTUDIO에서 ACTIVITY 추가하기



레이아웃 파일 LAYOUT1.XML

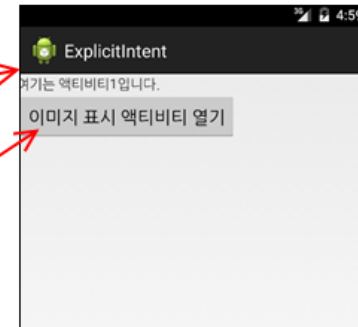
layout1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="여기는 액티비티1입니다." />
```

```
<Button
    android:id="@+id/Button01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="이미지 표시 액티비티 열기" />
```

```
</LinearLayout>
```



레이아웃 파일 LAYOUT2.XML

layout2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="여기는 액티비티2입니다" />
```

```
<ImageView
    android:id="@+id/imageView1"
    android:layout_width="135dp"
    android:layout_height="248dp"
    android:src="@drawable/ic_launcher" />
```

```
<Button
    android:id="@+id/Button01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="닫기" />
```

```
</LinearLayout>
```



ACTIVITY1.JAVA

Activity1.java

```
package kr.co.company.explicitintent;  
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.
```

```
public class Activity1 extends ActionBarActivity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.layout1);  
        Button b = (Button)findViewById(R.id.Button01);  
        b.setOnClickListener(new OnClickListener() {  
            public void onClick(View v) {  
                Intent intent = new Intent(Activity1.this,  
                                           Activity2.class);  
                startActivity(intent);  
            }  
        });  
    }  
}
```

버튼이 클릭되면 Activity2
를 시작한다.

두 번째 액티비티를 시작하려면 어떻게 하여야 하는가? 먼저 인텐트 객체를 생성한다. 우리는 두 번째 액티비티의 이름을 알고 있으므로 두 번째 액티비티의 클래스 이름을 인수로 주어서 인텐트 객체를 생성하면 된다. 즉 명시적인 인텐트를 사용하는 것이다.

ACTIVITY2.JAVA

Activity2.java

```
package kr.co.company.explicitintent;
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.

public class Activity2 extends ActionBarActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout2);
        Button b = (Button)findViewById(R.id.Button01);
        b.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                finish();
            }
        });
    }
}
```

이벤트 리스너에서는 버튼이 클릭되면 finish() 메소드를 호출하여서 현재의 액티비티를 종료한다.

메니페스트 파일

AndroidManifest.xml

...

```
<activity
    android:name="kr.co.company.explicitintent.Activity1"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name="Activity2" android:label="Activity2"></activity>
```

...

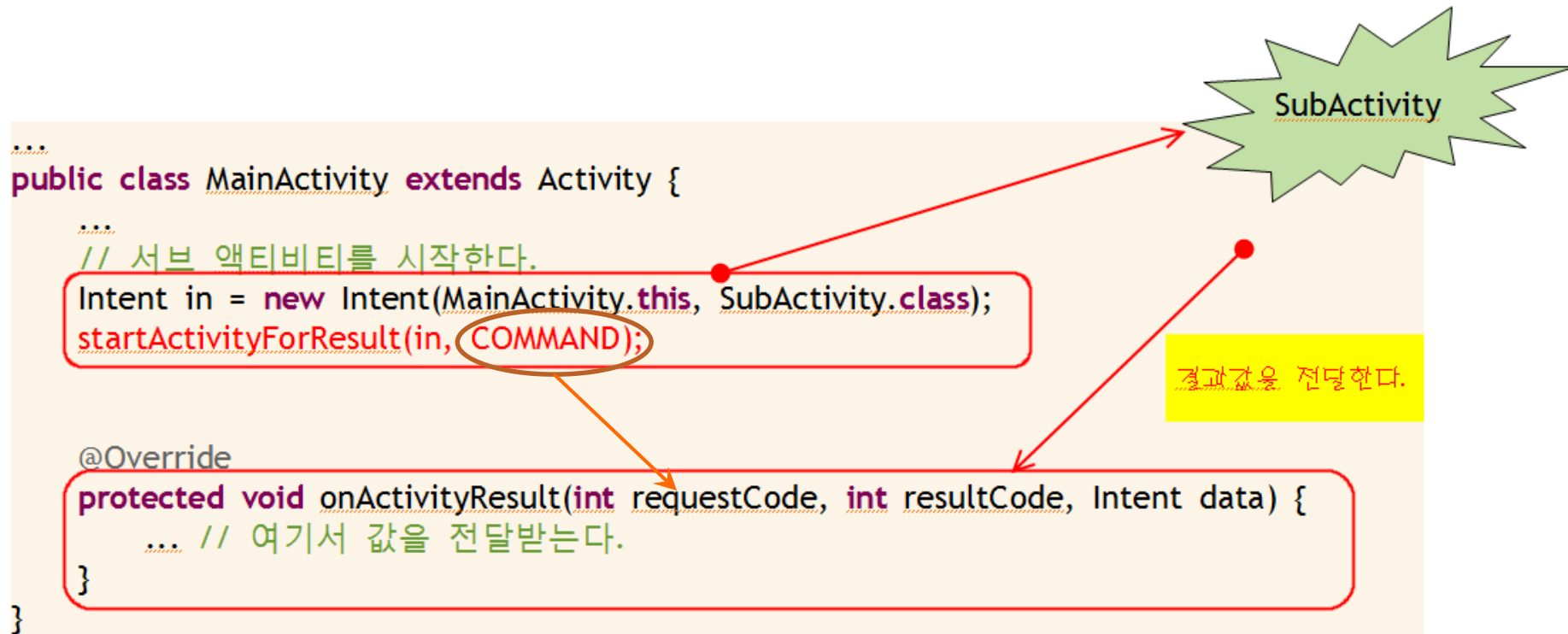
android:name에는 액티비티의 클래스 이름을 적어준다. 같은 패키지에 있는 경우에는 앞에 .을 찍거나 아니면 단순히 클래스 이름만 적어준다. 다른 패키지라면 패키지 이름을 포함한 완전한 경로 이름을 적어야 한다. android:label에는 타이틀바에 나타나는 텍스트를 적어준다.

안드로이드 스튜디오가 자동으로 수정해 줌

실행 결과

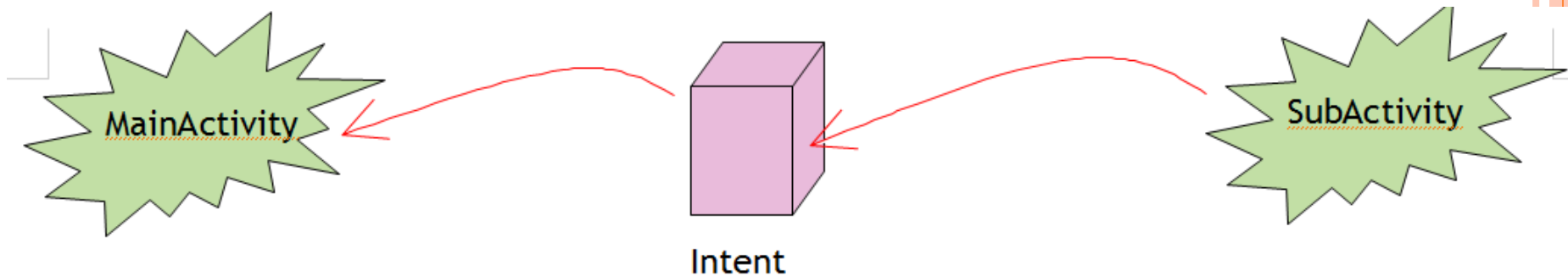


액티비티에서 결과받기



resultCode: RESULT_OK 또는 RESULT_CANCELLED 등

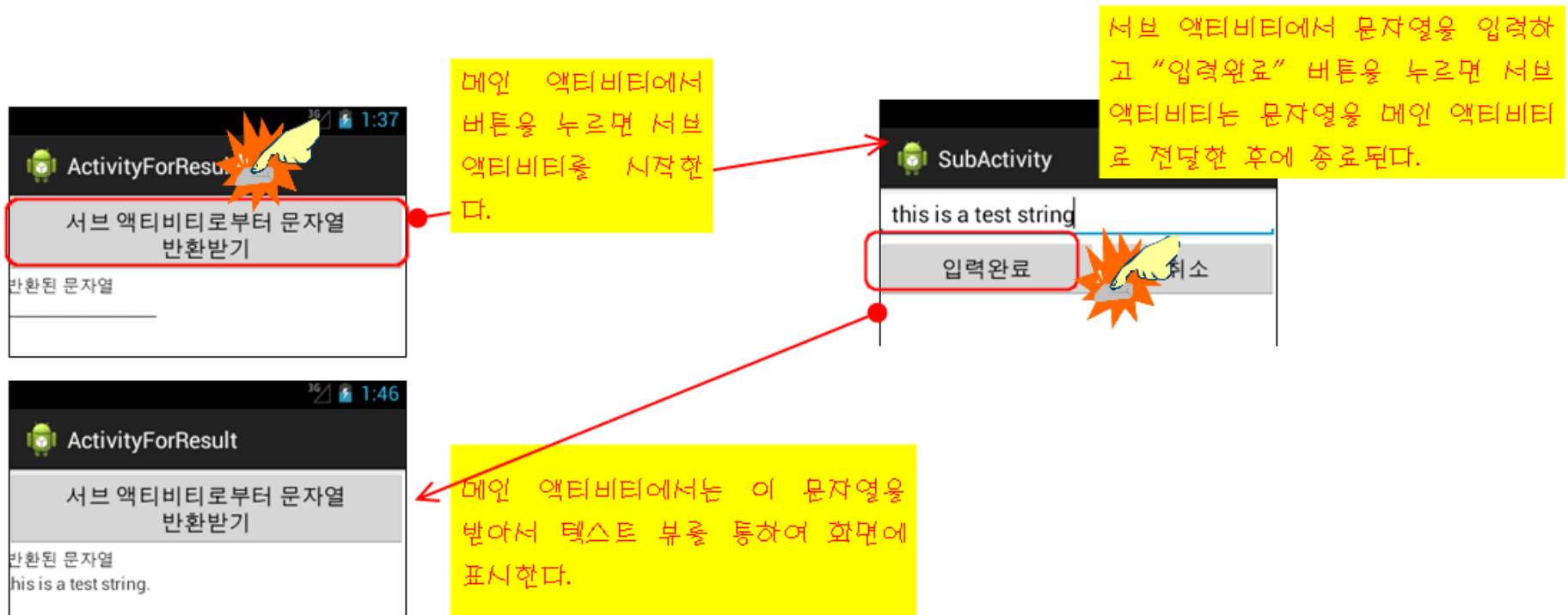
값을 저장하고, 값을 읽는 메소드



```
int  getIntExtra(String name, int defaultValue)
float  getFloatExtra(String name, float defaultValue)
int[]  getIntArrayExtra(String name)
String getStringExtra(String name)
```

```
putExtra(String name, int value)
putExtra(String name, float value)
putExtra(String name, int[] value)
putExtra(String name, String value)
```

예제



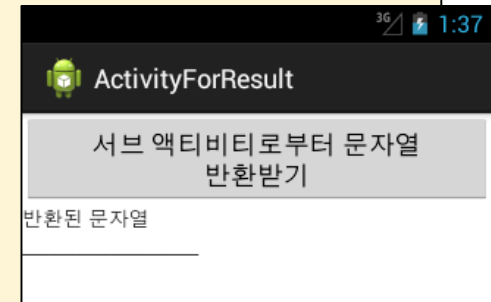
레이아웃 파일 MAIN.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button    android:layout_width="match_parent"
        android:text="서브 액티비티로부터 문자열 반환받기"
        android:layout_height="wrap_content"
        android:id="@+id/button">

    </Button>
    <TextView  android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="반환된 문자열"
        android:id="@+id/textView1">

    </TextView>
    <TextView  android:layout_height="wrap_content"
        android:text="_____"
        android:id="@+id/text"
        android:layout_width="match_parent">

    </TextView>
</LinearLayout>
```

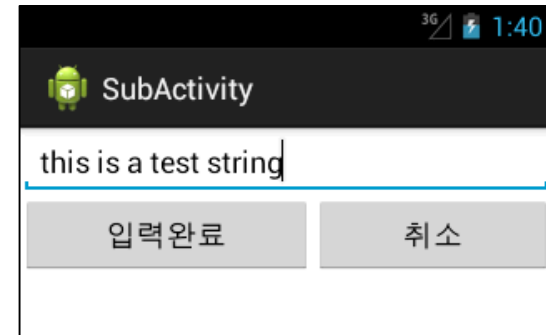


레이아웃 파일 SUB.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

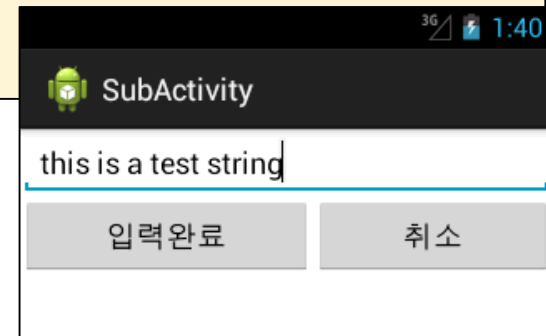
    <EditText
        android:id="@+id/edit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
        <requestFocus></requestFocus>
    </EditText>

    ...
```



레이아웃 파일 SUB.XML

```
<LinearLayout
    android:id="@+id/linearLayout1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center" >
    <Button
        android:id="@+id/button_ok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="입력완료" >
    </Button>
    <Button
        android:id="@+id/button_cancel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="취소" >
    </Button>
</LinearLayout>
```



매니페스트 파일 수정

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
...
    <application
        ...
        <activity
            android:name="kr.co.company.activityforresult.MainActivity"
            android:label="@string/app_name" >
                ...
        </activity>

        <activity
            android:name=".SubActivity"
            android:label="SubActivity" >
        </activity>
    </application>

</manifest>
```

MAINACTIVITY.JAVA

```
...
public class MainActivity extends Activity {
    static final int GET_STRING = 1;
    TextView text;
    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        setContentView(R.layout.main);
        Button button = (Button) findViewById(R.id.button);
        text = (TextView) findViewById(R.id.text);
        button.setOnClickListener(new OnClickListener() {
            // @Override
            public void onClick(View arg0) {
                Intent in = new Intent(MainActivity.this, SubActivity.class);
                startActivityForResult(in, GET_STRING);
            }
        });
    }
}
```

MAINACTIVITY.JAVA

```
...  
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    if (requestCode == GET_STRING) {  
        if (resultCode == RESULT_OK) {  
            text.setText(data.getStringExtra("INPUT_TEXT"));  
        }  
    }  
}  
}
```

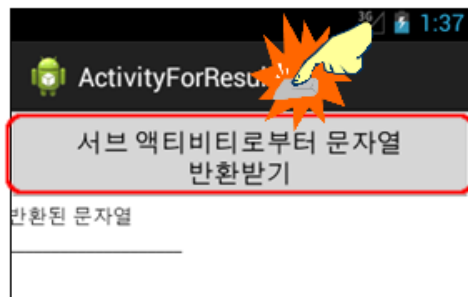
SUBACTIVITY.JAVA

```
...
public class SubActivity extends Activity {
    EditText edit;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.sub);
        edit = (EditText) findViewById(R.id.edit);
        Button button_ok = (Button) findViewById(R.id.button_ok);
        button_ok.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                Intent intent = new Intent();
                intent.putExtra("INPUT_TEXT", edit.getText().toString());
                setResult(RESULT_OK, intent);
                finish();
            }
        });
    }
}
```

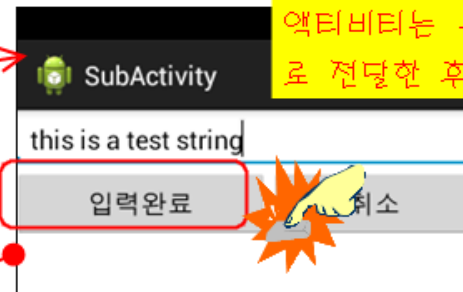
SUBACTIVITY.JAVA

```
Button button_cancel = (Button) findViewById(R.id.button_cancel);
    button_cancel.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            setResult(RESULT_CANCELED);
            finish();
        }
    });
}
```

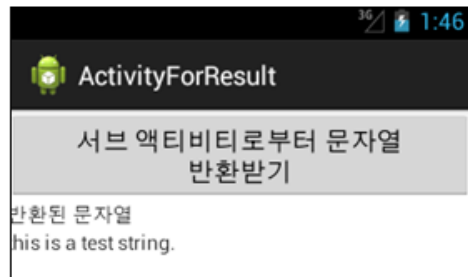
실행 결과



메인 액티비티에서 버튼을 누르면 서브 액티비티를 시작한다.



서브 액티비티에서 문자열을 입력하고 "입력완료" 버튼을 누르면 서브 액티비티는 문자열을 메인 액티비티로 전달한 후에 종료된다.



메인 액티비티에서는 이 문자열을 받아서 텍스트 뷰를 통하여 화면에 표시한다.

실습(1)

- 날짜를 받는 앱을 작성하시오.

Practice10_1

이름: Hong

생일: 생일 날짜

생일날짜 설정

Practice10_1

Hong 생일을 설정

April 2017

	S	M	T	W	T	F	S
13	26	27	28	29	30	31	1
14	2	3	4	5	6	7	8
15	9	10	11	12	13	14	15
16	16	17	18	19	20	21	22
17	23	24	25	26	27	28	29
18	30	1	2	3	4	5	6

Practice10_1

이름: Hong

생일: 2017년 4월 9일

생일날짜 설정

실습(1) 힌트

- 사용한 위젯: Button, EditText, TextView, CalendarView
- 주요 API:
 - getIntent() : 두번째 Activity에서 Intent를 받는 API
 - 두번째 Activity의 onCreate()에서 Intent intent = getIntent() 수행
 - setOnDateChangeListener(): CalendarView에서 변경된 날짜 받는 Listener 설정하는 API
 - getText().toString(): EditText 위젯에서 String 객체를 받는 API