

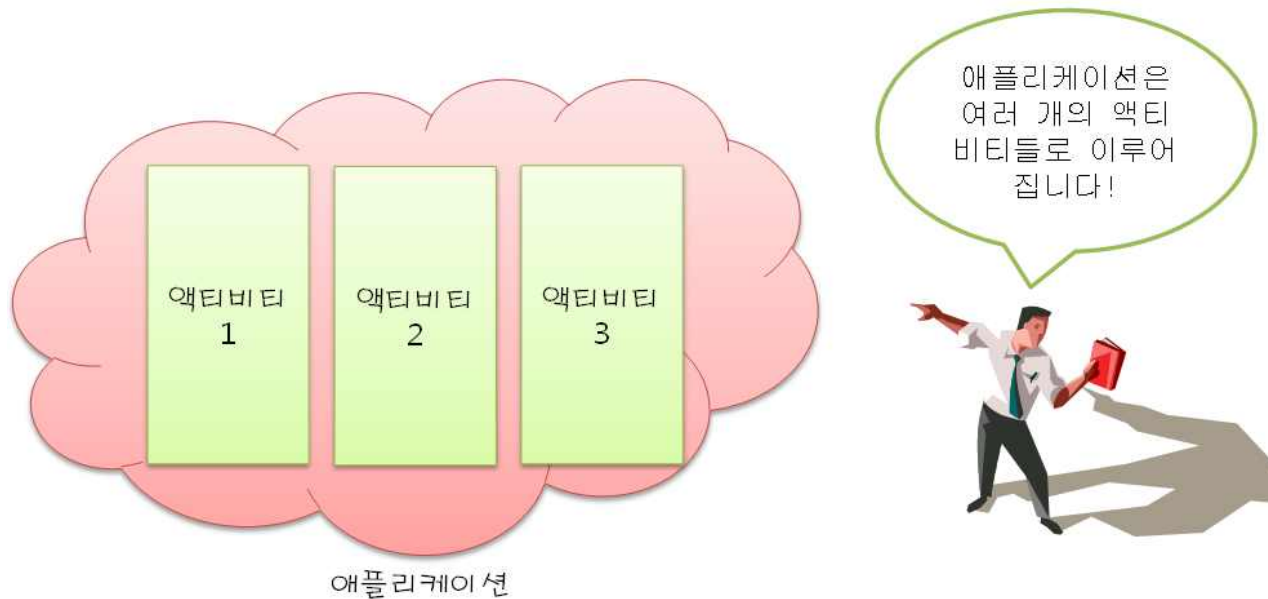
CHAP 8. 액티비티와 인텐트

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.



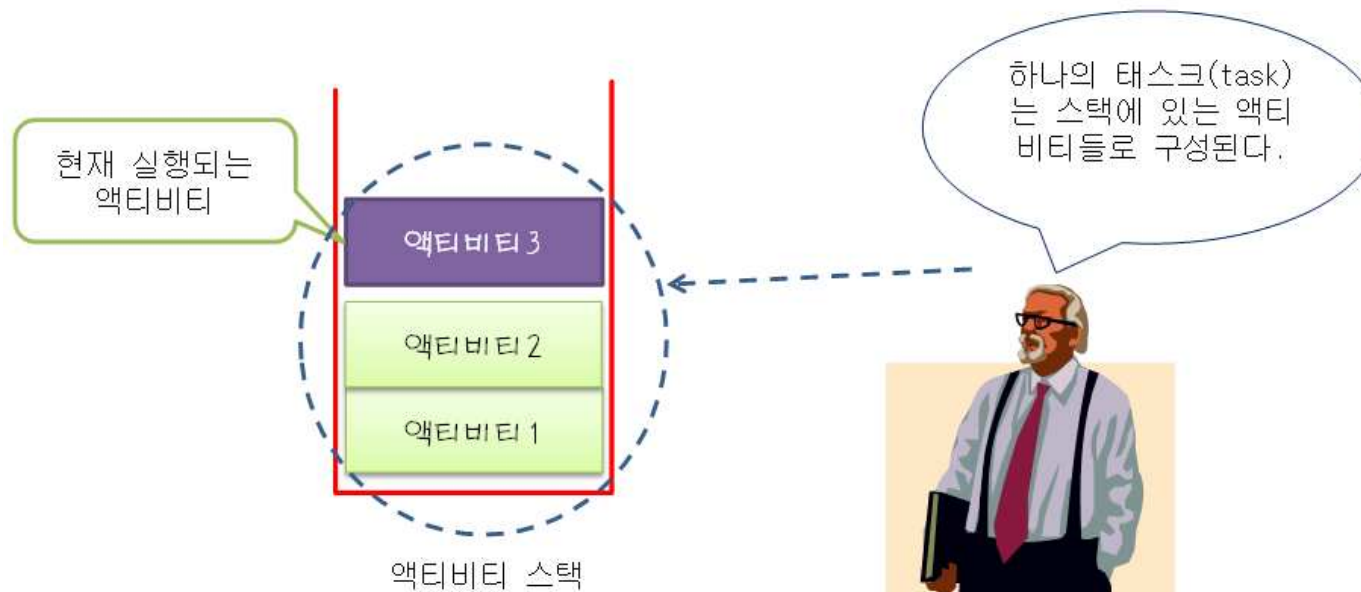
액티비티

- 애플리케이션을 구성하는 빌딩 블록



태스크

□ 스택에 있는 액티비티

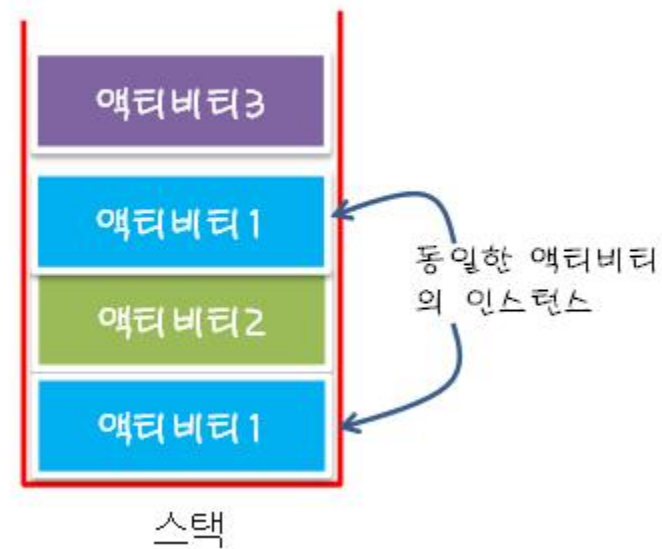


액티비티 스택

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



다음과 같은 상태도 가능하다



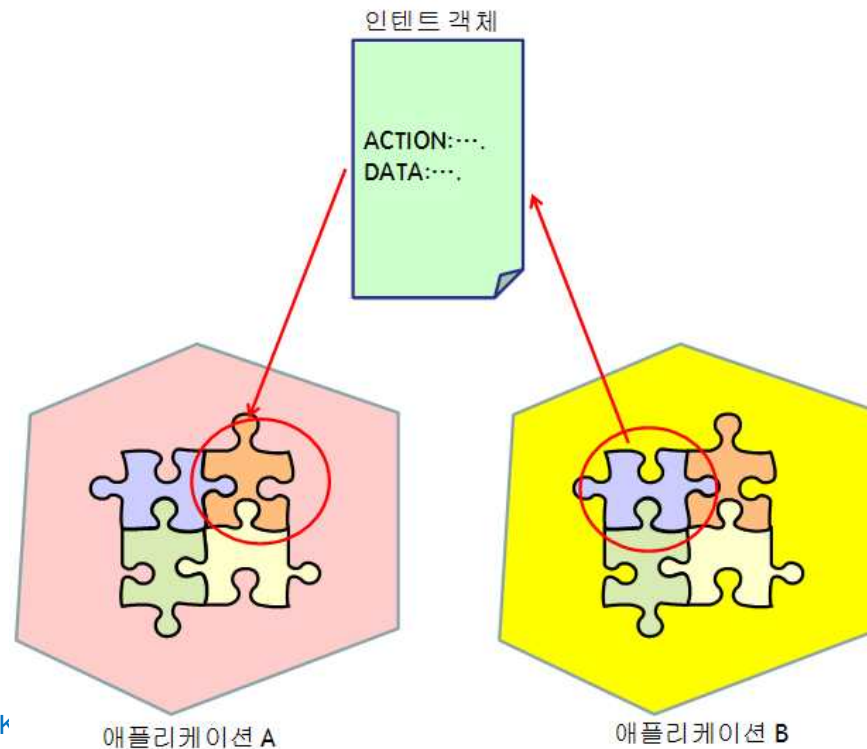
인텐트

- 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

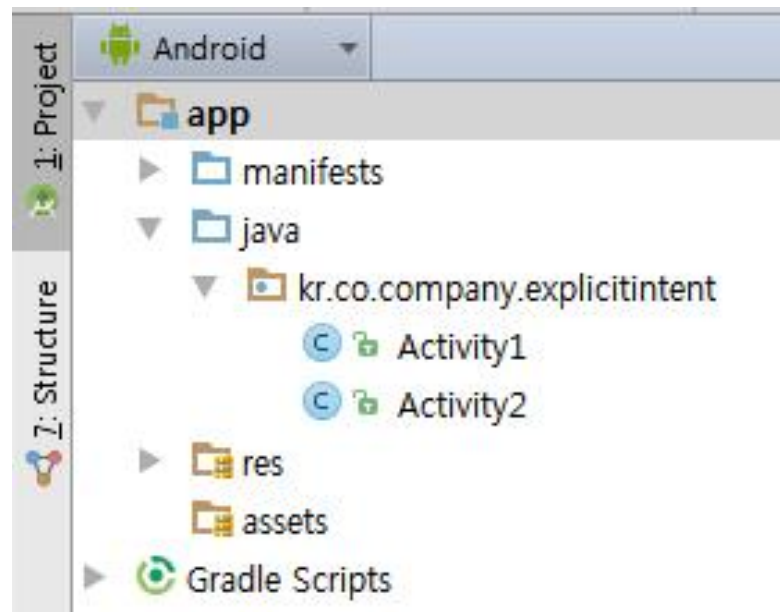
- 실행하고자 하는 액티비티의 이름을 적어 준다.



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

명시적인 인텐트 예제

- 여기서 두 개의 액티비티로 이루어진 애플리케이션을 작성하여 보자. 첫 번째 액티비티는 **Activity1**, 두 번째 액티비티는 **Activity2**라고 하자.



레이아웃 파일 layout1.xml

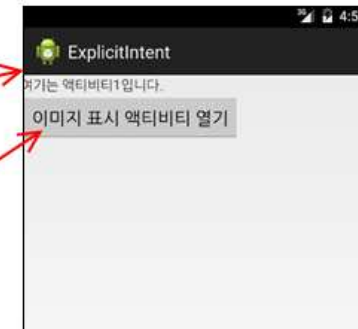
layout1.xml

```
<?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);  
            }  
        });  
    }  
}
```


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();  
            }  
        });  
    }  
}
```


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" />

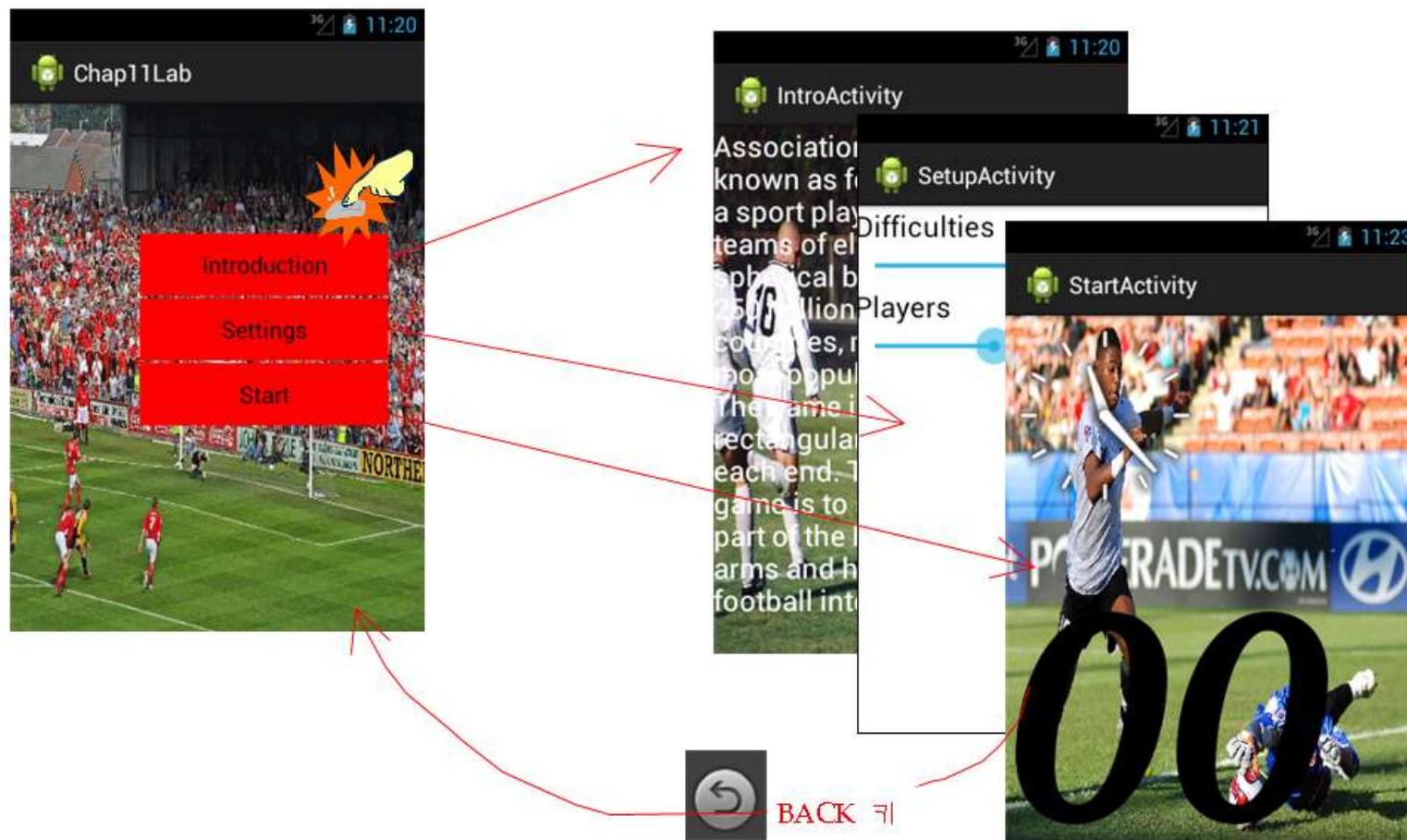
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name="Activity2" android:label="Activity2"></activity>
```

...

실행 결과



여러 페이지로 된 애플리케이션 작성



Main.xml

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/FrameLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/soccer" >
```

```
<AbsoluteLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
```

```
<Button
    android:id="@+id/button1"
    android:layout_width="192dp"
    android:layout_height="wrap_content"
    android:layout_x="100px"
    android:layout_y="100px"
    android:background="#ff0000"
    android:gravity="center"
    android:onClick="myListener"
    android:text="Introduction" />
```

...

```
</AbsoluteLayout>
</FrameLayout>
```

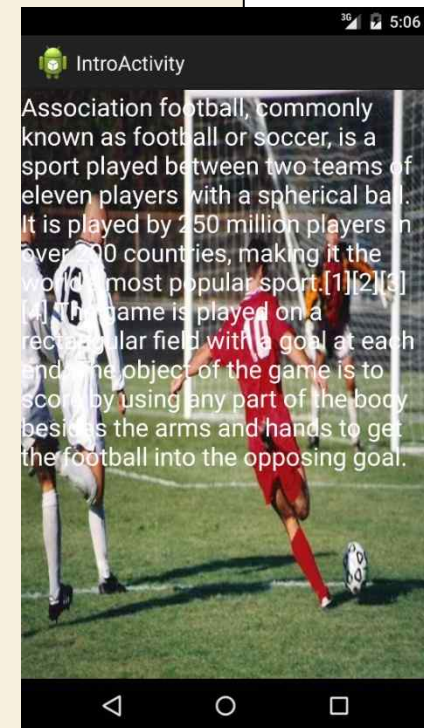


Intro.xml

```
<?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:background="@drawable/soccer2"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Association football, ...."
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="#FFFFFF" />

</LinearLayout>
```



Setup.xml

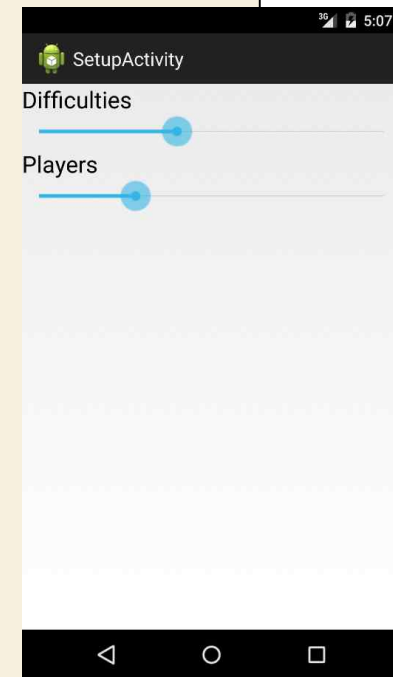
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Difficulties"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <SeekBar
        android:id="@+id/seekBar1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Players"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <SeekBar
        android:id="@+id/seekBar2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```



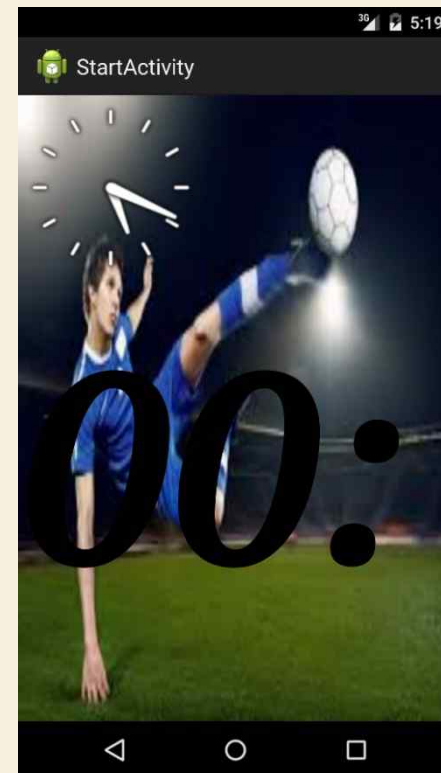
Start.xml

```
<?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:background="@drawable/soccer3"
    android:orientation="vertical" >

    <AnalogClock
        android:id="@+id/analogClock1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <Chronometer
        android:id="@+id/chronometer1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Chronometer"
        android:textSize="@dimen/Large"
        android:textStyle="bold|italic"
        android:typeface="serif" />

</LinearLayout>
```



MultiPageActivity.java

```
package kr.co.company.multipage;
// 소스만 입력하고 Ctrl-Shift-O를 눌러서 import 문장을 자동으로 생성한다.

public class MultiPageActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
    public void myListener(View target) {
        Intent intent = new Intent(getApplicationContext(),
            IntroActivity.class);
        startActivity(intent);
    }
    public void myListener1(View target) {
        Intent intent = new Intent(getApplicationContext(),
            SetupActivity.class);
        startActivity(intent);
    }
    public void myListener2(View target) {
        Intent intent = new Intent(getApplicationContext(),
            StartActivity.class);
        startActivity(intent);
    }
}
```


IntroActivity.java

```
package kr.co.company.chap11lab;

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

public class IntroActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.intro);
    }
}
```

매니페스트 파일

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
...
    <application
        ...
        <activity
            android:name="IntroActivity"
            android:label="IntroActivity" >
        </activity>

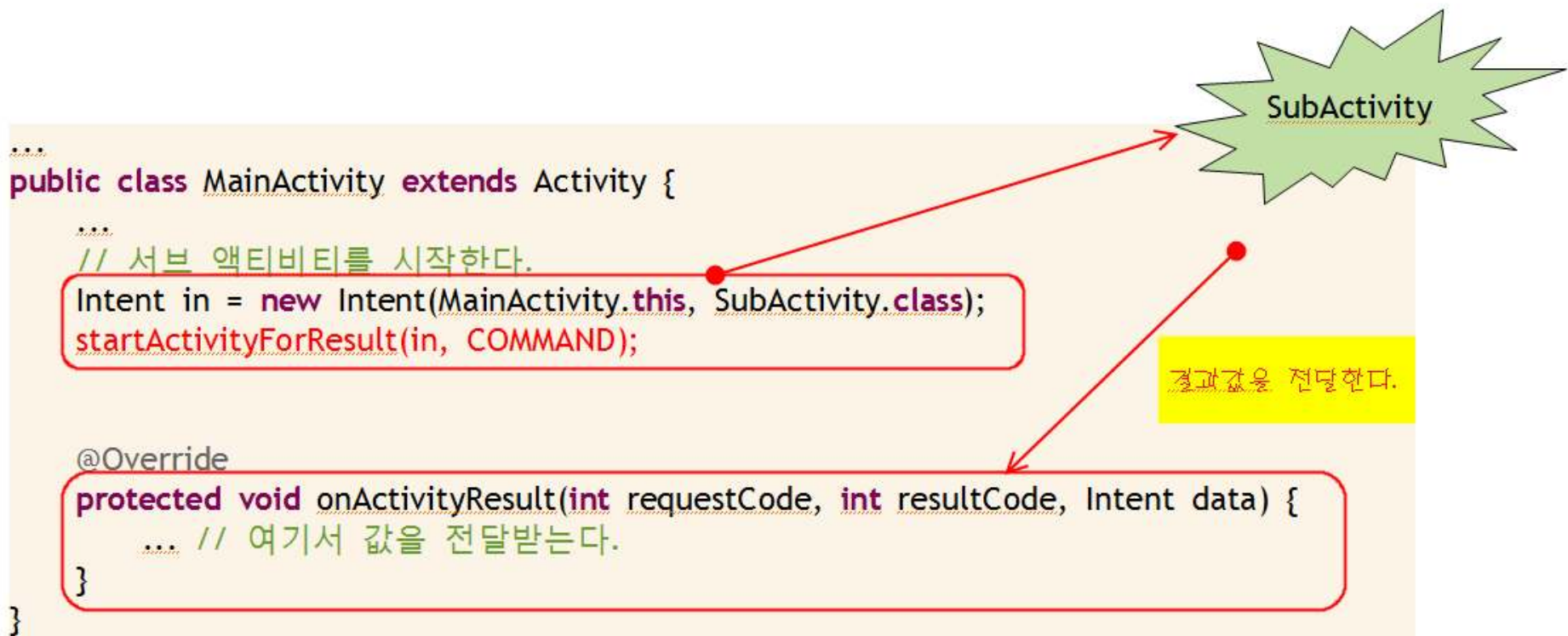
        <activity
            android:name="SetupActivity"
            android:label="SetupActivity" >
        </activity>

        <activity
            android:name="StartActivity"
            android:label="StartActivity" >
        </activity>
    </application>
</manifest>
```

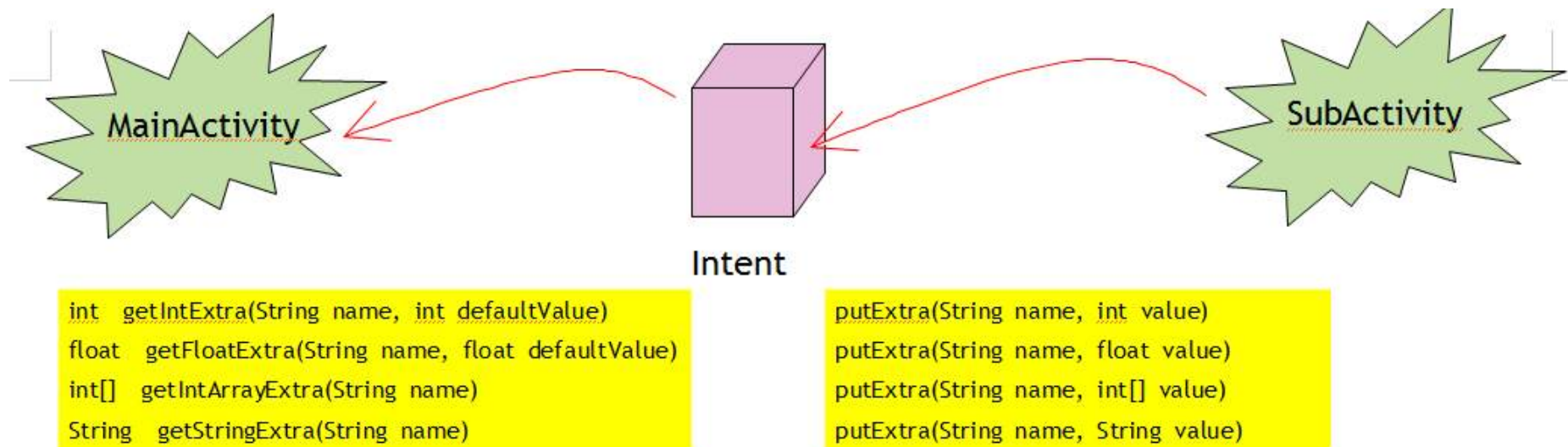
실행 결과



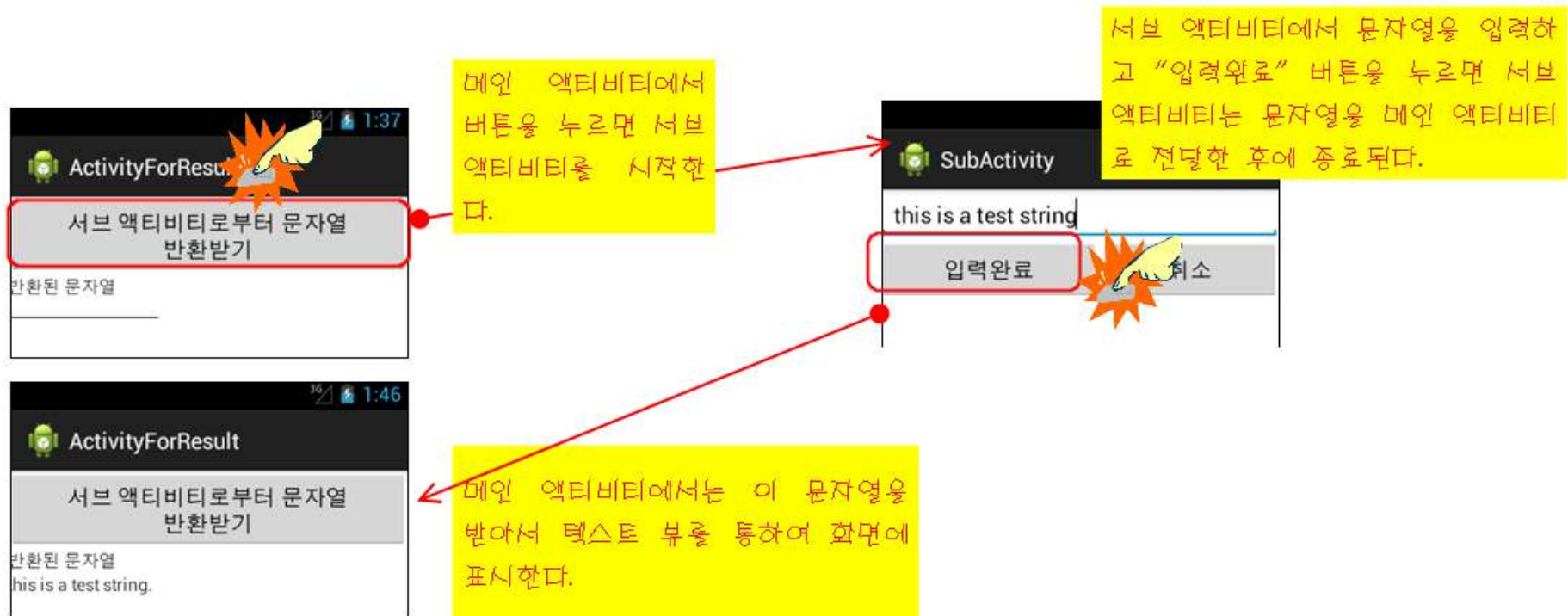
액티비티에서 결과받기



값을 저장하고, 값을 읽는 메소드



예제



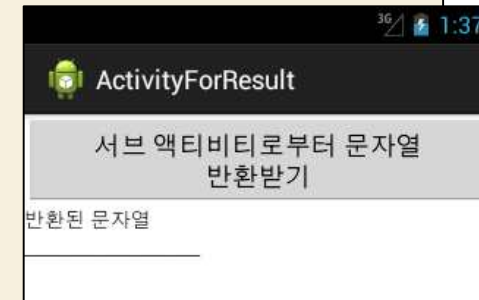
레이아웃 파일 main.xml

```
<?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>
```

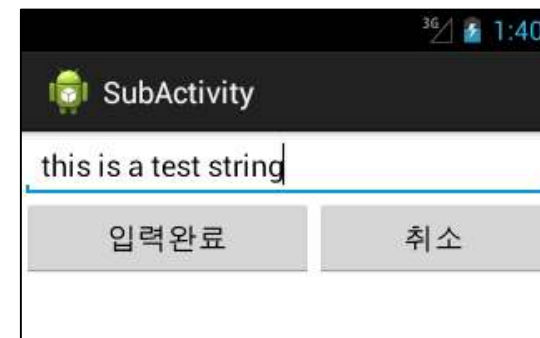


레이아웃 파일 sub.xml

```
<?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" >

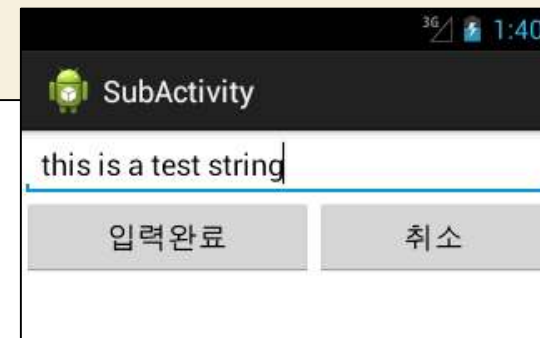
    <EditText
        android:id="@+id/edit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
        <requestFocus></requestFocus>
    </EditText>

    ...
```



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 icle) {  
        super.onCreate(icle);  
        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

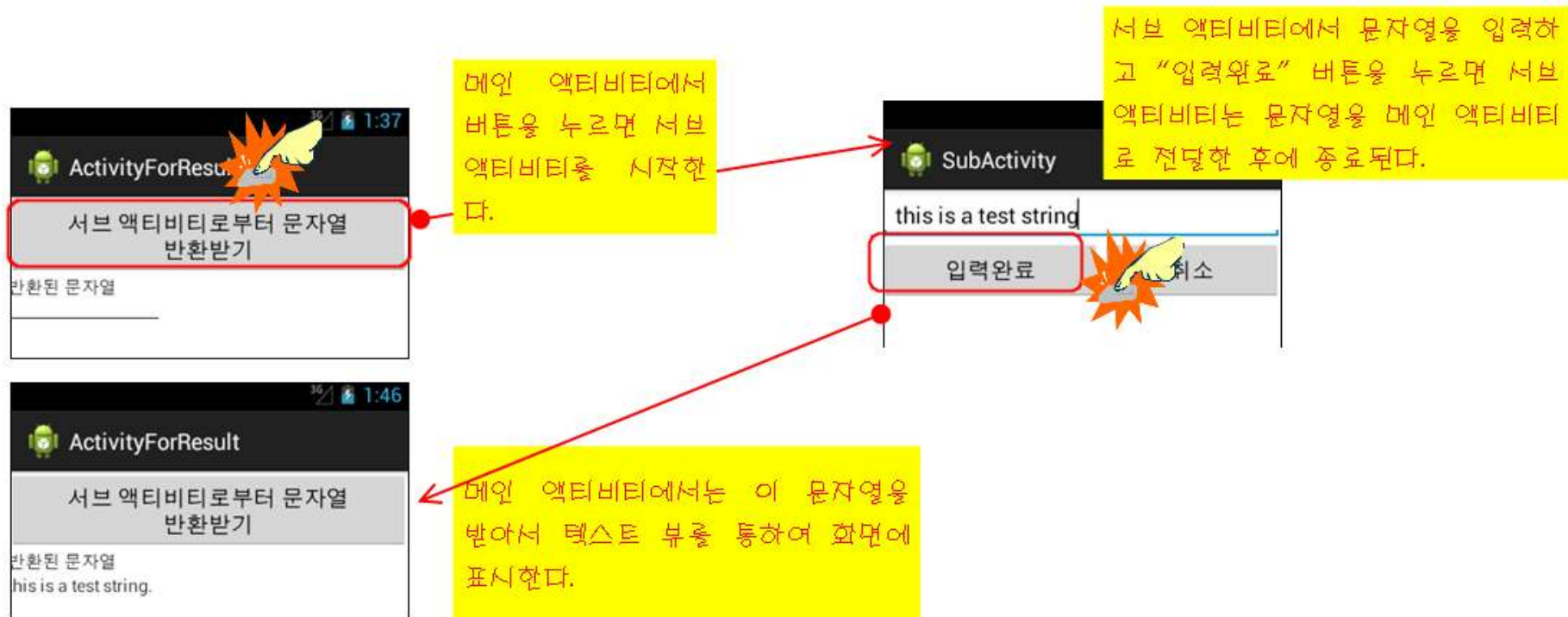
...

```
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

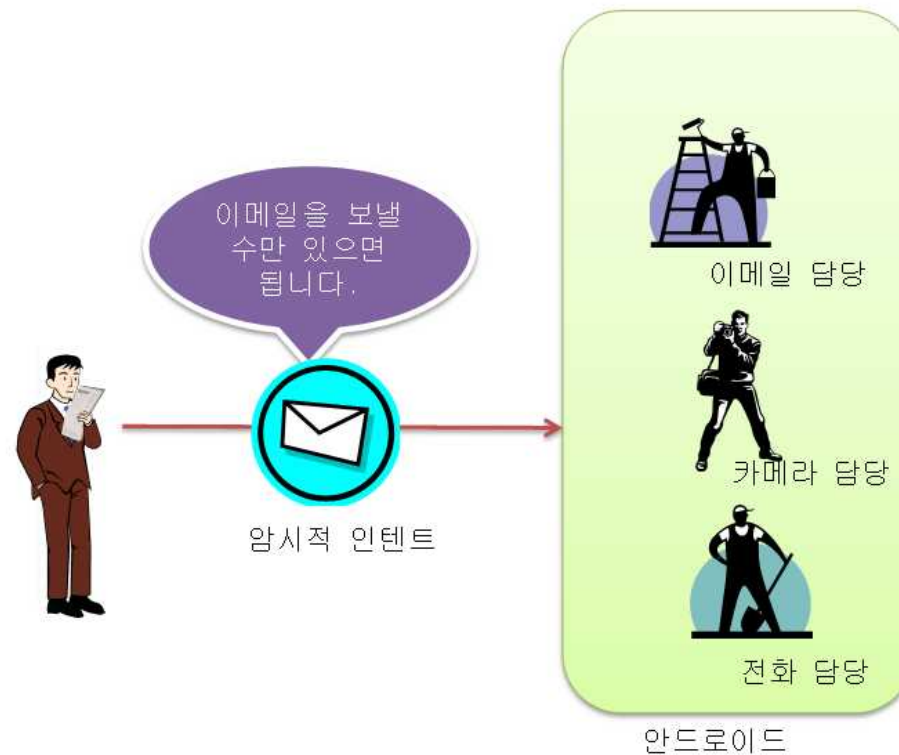
```
Button button_cancel = (Button) findViewById(R.id.button_cancel);
    button_cancel.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            setResult(RESULT_CANCELED);
            finish();
        }
    });
}
```

실행 결과



암시적인 인텐트

- 어떤 작업을 하기를 원하지만 그 작업을 담당하는 컴포넌트의 이름을 명확하게 모르는 경우에 사용



암시적인 인텐트의 형식

```
Intent intent = new Intent(Intent.ACTION_SEND);  
intent.putExtra(Intent.EXTRA_EMAIL, recipientArray);  
startActivity(intent);
```

액션의 종류

상수	타겟 컴포넌트	액션
ACTIN_VIEW	액티비티	데이터를 사용자에게 표시한다.
ACTION_EDIT	액티비티	사용자가 편집할 수 있는 데이터를 표시한다.
ACTION_MAIN	액티비티	태스크의 초기 액티비티로 설정한다.
ACTION_CALL	액티비티	전화 통화를 시작한다.
ACTION_SYNC	액티비티	모바일 장치의 데이터를 서버 상의 데이터와 일치시킨다.
ACTION_DIAL	액티비티	전화 번호를 누르는 화면을 표시한다.

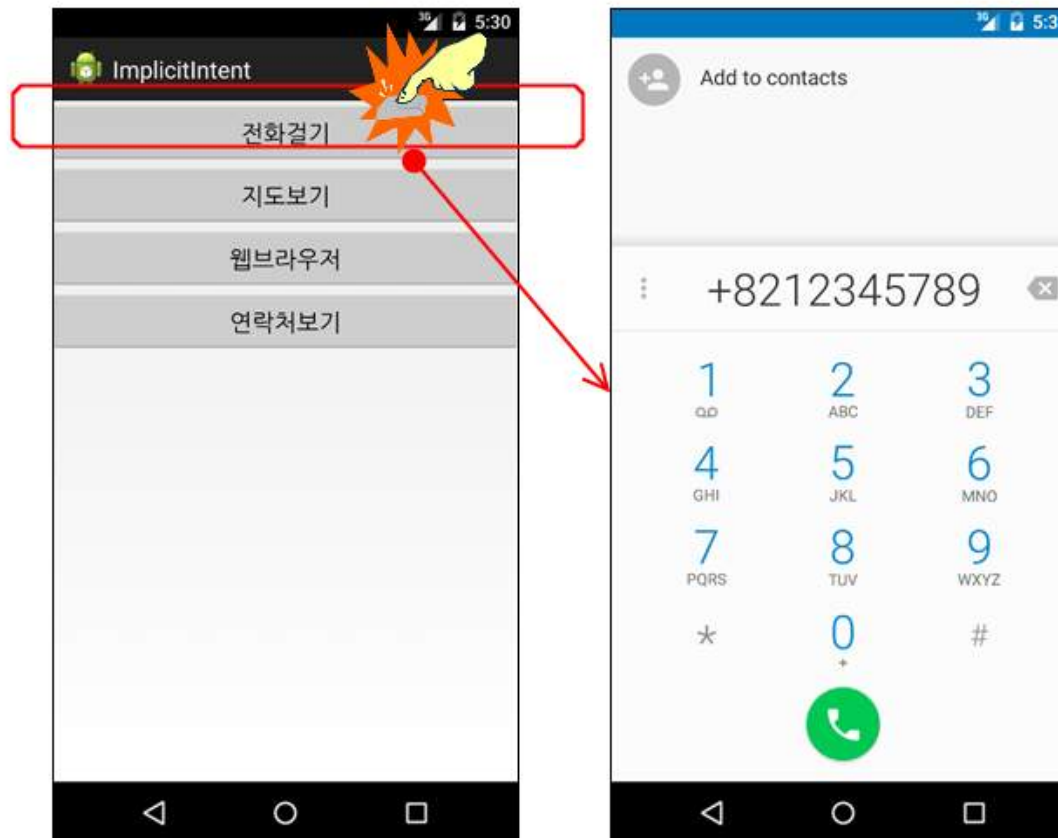
암시적인 인텐트 예

액션

- ▷ ACTION_VIEW *content://contacts/people/1* - 1번 연락처 정보를 표시한다.
- ▷ ACTION_DIAL *content://contacts/people/1* - 1번 연락처로 전화걸기 화면을 표시한다.
- ▷ ACTION_VIEW *tel:0101234567* - 0101234567번 전화번호로 전화걸기 화면을 표시한다.
- ▷ ACTION_DIAL *tel:0101234567* -- 0101234567번 전화번호로 전화걸기 화면을 표시한다.
- ▷ ACTION_EDIT *content://contacts/people/1* -- 1번 연락처 정보를 편집한다.
- ▷ ACTION_VIEW *content://contacts/people/* -- 연락처 리스트를 표시한다.

데이터

인텐트 예제



사용자 인터페이스

```
<?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:id="@+id/call"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:text="전화걸기" >
    </Button>

    <Button
        android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:text="지도보기" >
    </Button>
```

사용자 인터페이스

<Button

```
    android:id="@+id/web"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:onClick="onClick"  
    android:text="웹브라우저" >
```

</Button>

<Button

```
    android:id="@+id/contact"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:onClick="onClick"  
    android:text="연락처보기" >
```

</Button>

</LinearLayout>

암시적 인텐트 예제

```
public class ImplicitIntentActivity extends Activity {

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

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }

    public void onClick(View view) {
        Intent intent = null;
        switch (view.getId()) {
            case R.id.web:
                intent = new Intent(Intent.ACTION_VIEW,
                                    Uri.parse("http://www.google.com"));
                break;
            case R.id.call:
                intent = new Intent(Intent.ACTION_DIAL,
                                    Uri.parse("tel:(+82)12345789"));
                break;
        }
    }
}
```

암시적 인텐트 예제

```
        case R.id.map:
            intent = new Intent(Intent.ACTION_VIEW,
                                Uri.parse("geo:37.30,127.2?z=10"));
            break;
        case R.id.contact:
            intent = new Intent(Intent.ACTION_VIEW,
                                Uri.parse("content://contacts/people/"));
            break;
    }
    if (intent != null) {
        startActivity(intent);
    }
}
```


매니페스트 파일 수정

...

```
<uses-permission android:name="android.permission.CALL_PHONE" >
```

```
</uses-permission>
```

```
<uses-permission android:name="android.permission.CAMERA" >
```

```
</uses-permission>
```

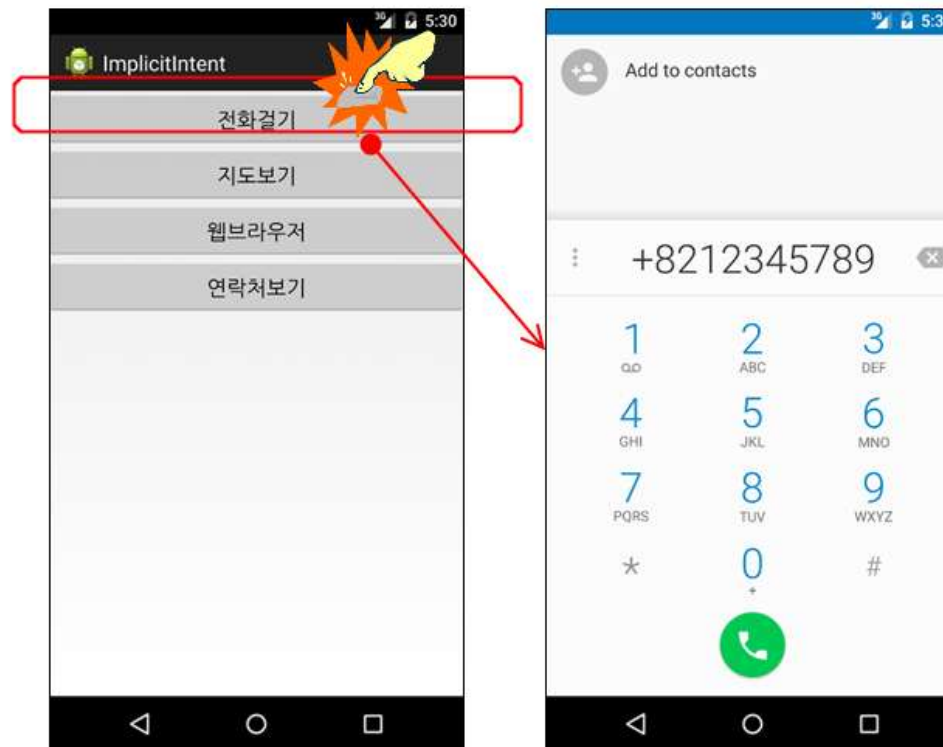
```
<uses-permission android:name="android.permission.READ_CONTACTS" >
```

```
</uses-permission>
```

```
<uses-permission android:name="android.permission.INTERNET"/>
```

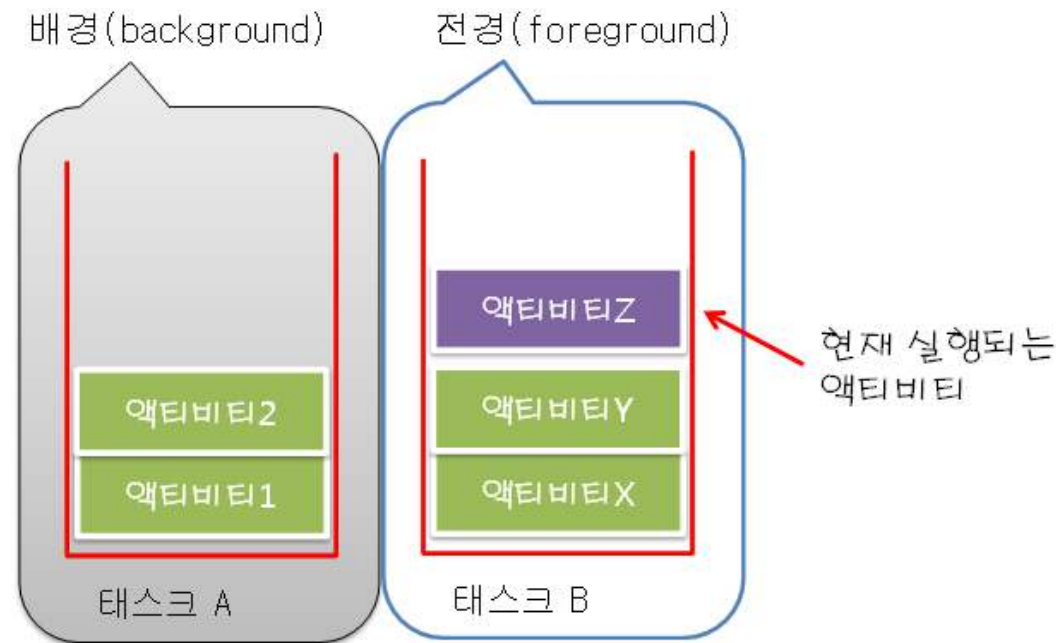
...

실행 결과



멀티태스킹

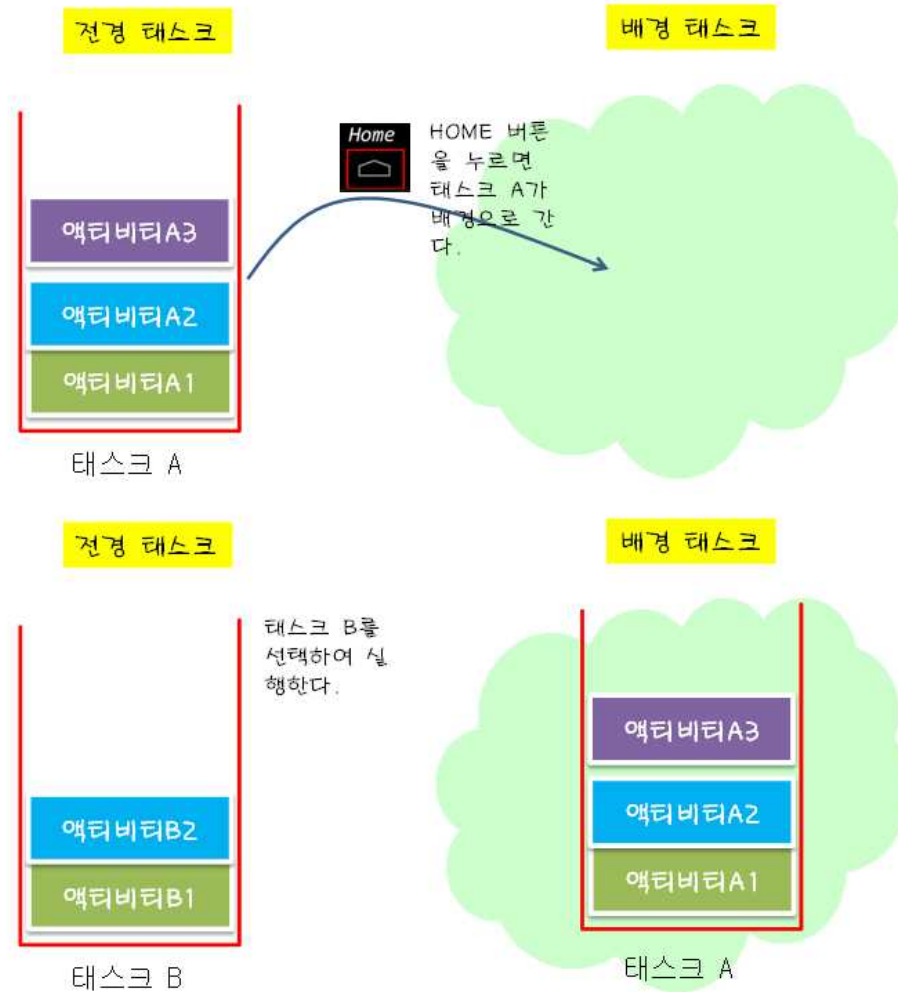
- 동시에 여러 태스크를 실행
- 현재의 태스크를 배경(background)으로 보내고 다른 태스크를 전경(foreground)에서 시작할 수 있다.



멀티 태스킹의 예

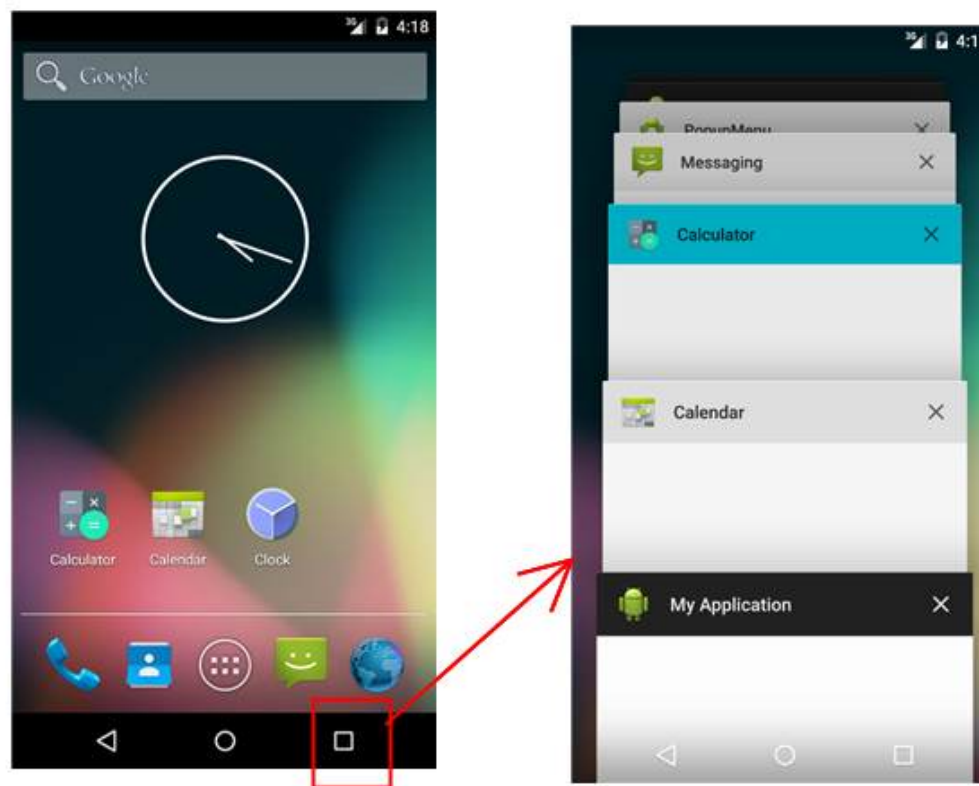


HOME 키를 누르면 멀티 태스킹이 시작된다.



오버뷰 화면

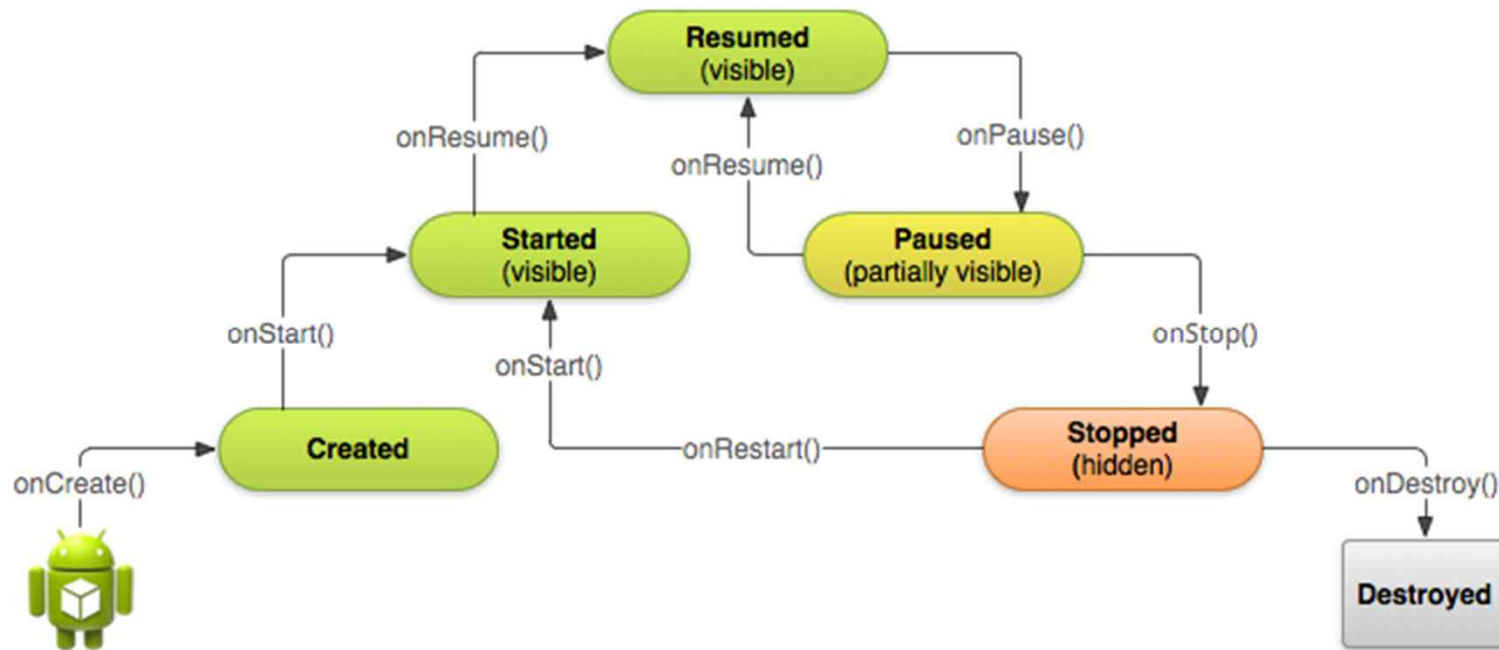
- 최근에 사용된 액티비티들과 태스크들을 보여주는 화면



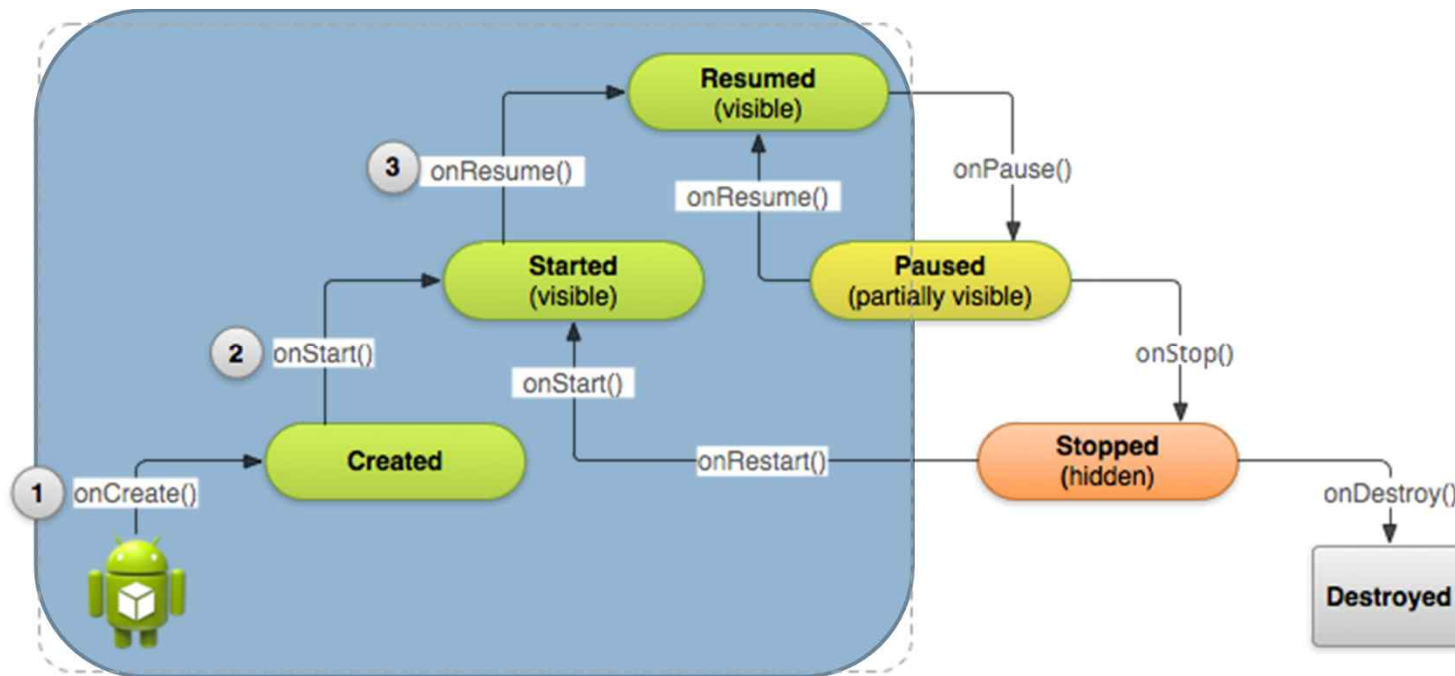
액티비티 생애주기

- 실행 상태(**resumed, running**): 액티비티가 전경에 위치하고 있으며 사용자의 포커스를 가지고 있다.
- 일시멈춤 상태(**paused**): 다른 액티비티가 전경에 있으며 포커스를 가지고 있지만 현재 액티비티의 일부가 아직도 화면에서 보이고 있는 상태이다.
- 정지 상태(**stopped**): 액티비티는 배경에 위치한다.

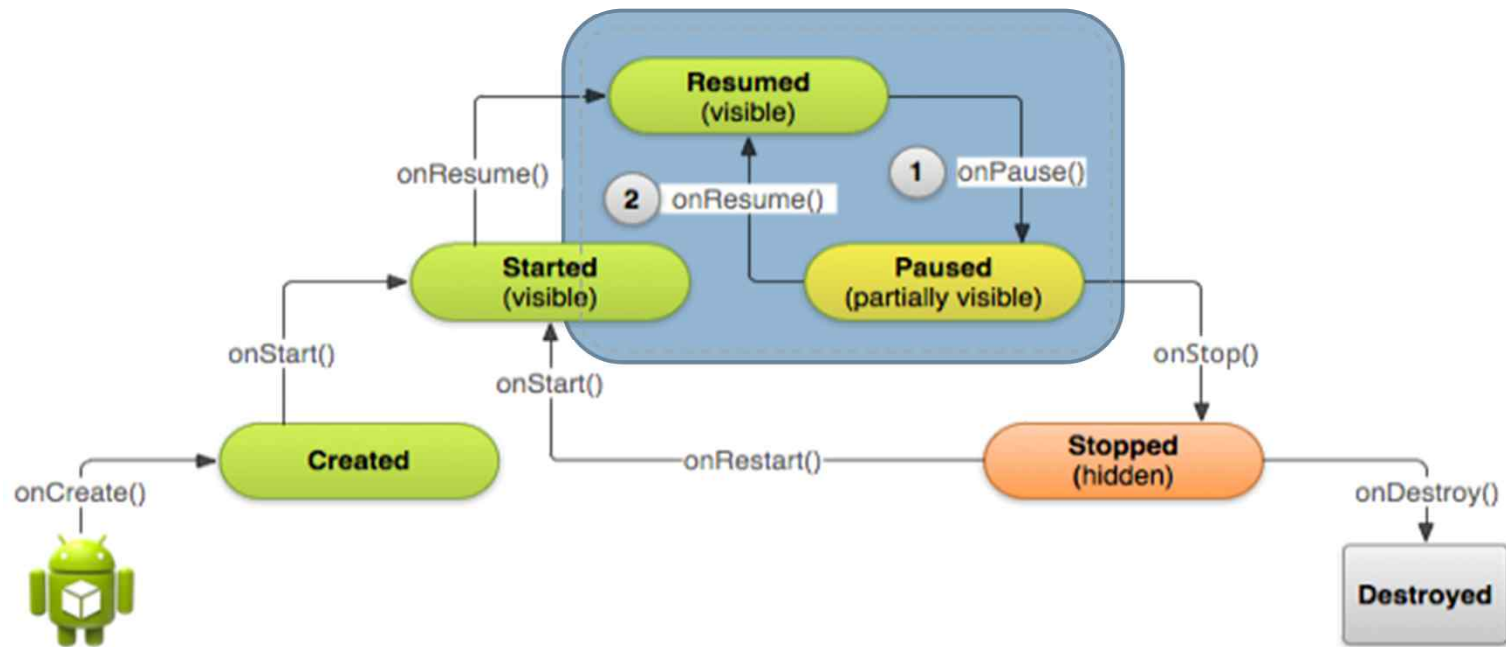
액티비티 상태



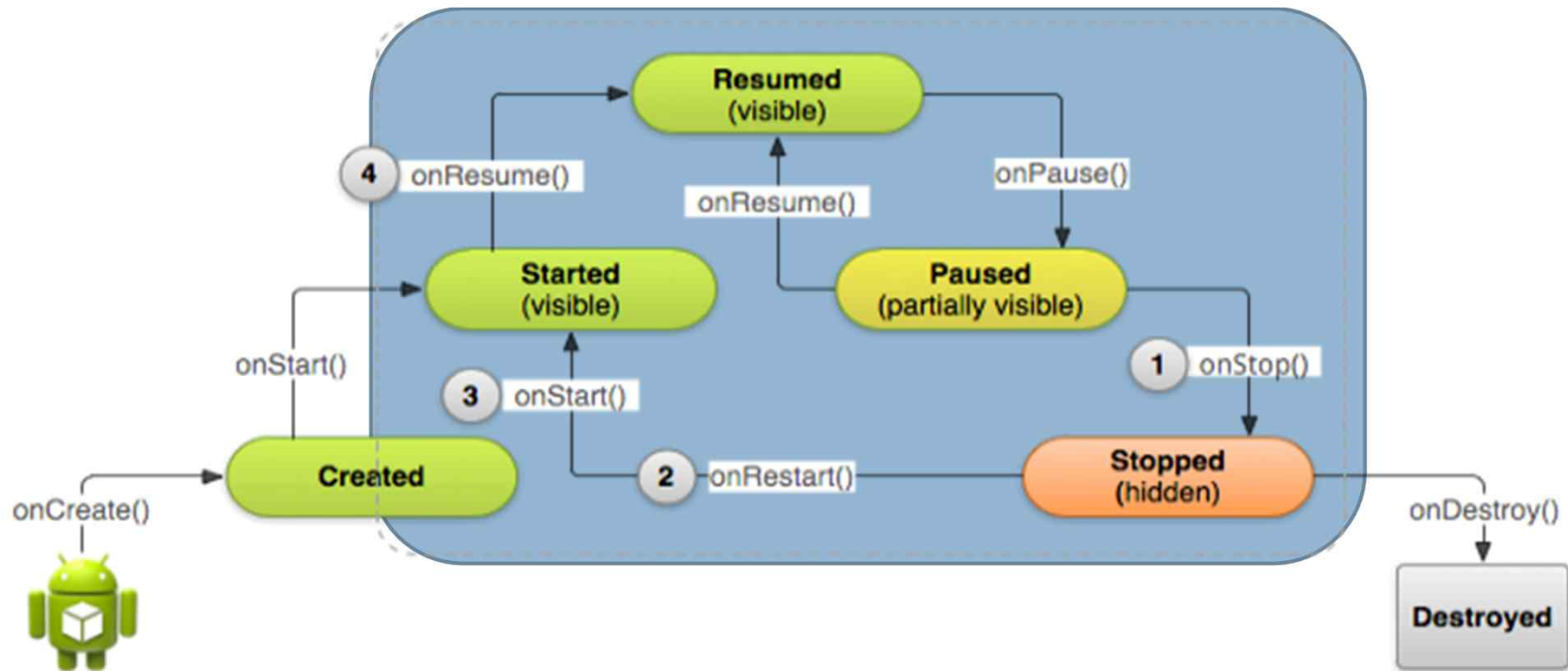
액티비티 객체 생성 단계



이식 머춤 상태



정지/되어다가 다시 실행하는 경우



중요한 콜백 메소드

□ onCreate()

- ▣ 액티비티가 생성되면서 호출
- ▣ 중요한 구성요소들을 초기화

□ onPause()

- ▣ 사용자가 액티비티를 떠나고 있을 때, 이 메소드가 호출
- ▣ 그 동안 이루어졌던 변경사항을 저장

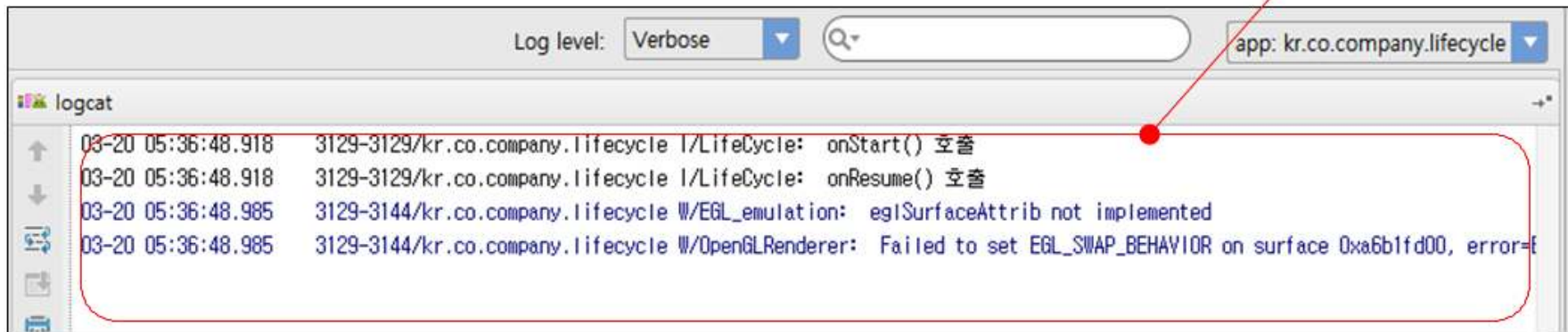
액티비티 생애주기 예제

```
...  
public class LifeCycleTest extends Activity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
    @Override  
    public void onStart() {  
        super.onStart();  
        Log.i("LifeCycle", "onStart() 호출");  
    }  
    @Override  
    public void onResume() {  
        super.onResume();  
        Log.i("LifeCycle", "onResume() 호출");  
    }  
}
```

액티비티 생애주기 예제

```
@Override
    public void onPause() {
        super.onPause();
        Log.i("LifeCycle", "onPause() 호출");
    }
    @Override
    public void onStop() {
        super.onStop();
        Log.i("LifeCycle", "onStop() 호출");
    }
    @Override
    public void onDestroy() {
        super.onDestroy();
        Log.i("LifeCycle", "onDestroy() 호출");
    }
}
```

실행 결과

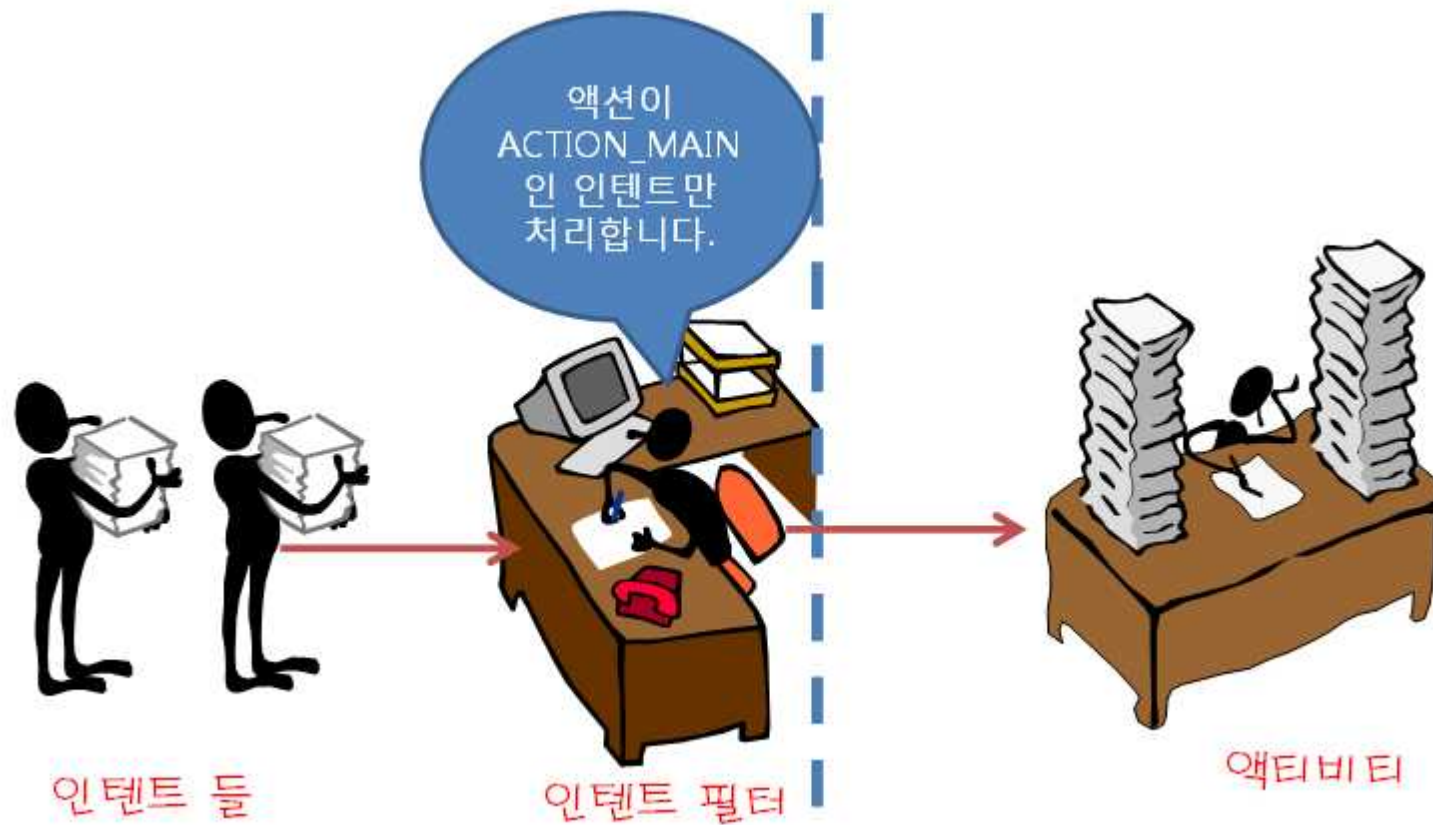


인텐트 필터

- 컴포넌트는 자신들이 처리할 수 있는 인텐트의 종류를 인텐트 필터에 기록한다.



인텐트 필터



이테=픽터

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="kr.co.company.ImplicitIntent"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".ImplicitIntent"
            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>
    <uses-sdk android:minSdkVersion="9" />
    <uses-permission android:name="android.permission.CALL_PHONE"></uses-permission>
</manifest>
```

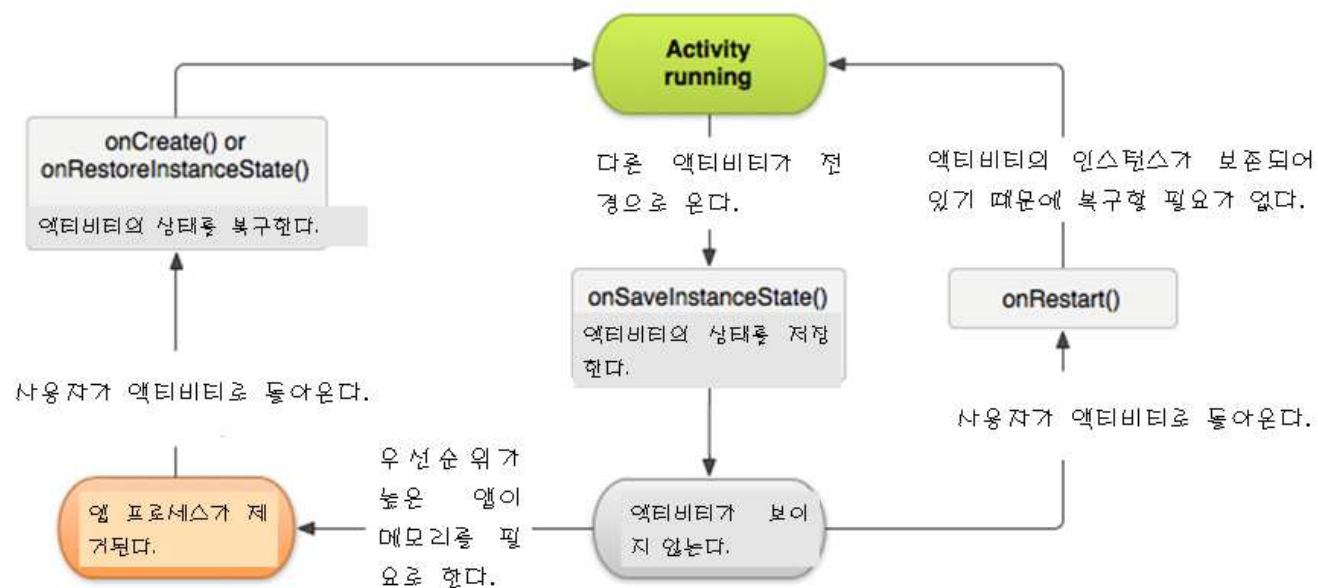
노트패드의 인텐트 필터

```
<activity android:name="NotesList" android:label="@string/title_notes_list">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>

    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <action android:name="android.intent.action.EDIT" />
        <action android:name="android.intent.action.PICK" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="vnd.android.cursor.dir/vnd.google.note" />
    </intent-filter>

    <intent-filter>
        <action android:name="android.intent.action.GET_CONTENT" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="vnd.android.cursor.item/vnd.google.note" />
    </intent-filter>
</activity>
```

액티비티 상태 저장



액티비티 상태 저장

- 선택한 상품을 저장하는 애플리케이션 작성



상태 저장 및 복구

```
public class SaveRestoreTestActivity extends Activity {
    Button button1, button2;
    TextView text;
    int count = 0;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        text = (TextView) findViewById(R.id.text);
        button1 = (Button) findViewById(R.id.button1);
        button1.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                count++;
                text.setText("현재 개수=" + count);
            }
        });
        button2 = (Button) findViewById(R.id.button2);
        button2.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                count--;
                text.setText("현재 개수=" + count);
            }
        });
    }
}
```

상태 저장 및 복구

```
public class SaveRestoreTestActivity extends Activity {  
    ...  
    public void onCreate(Bundle savedInstanceState) {  
        ...  
        if (savedInstanceState != null) {  
            count = savedInstanceState.getInt("count");  
            text.setText("현재 개수=" + count);  
        }  
    }  
    @Override  
    public void onSaveInstanceState(Bundle outState) {  
        super.onSaveInstanceState(outState);  
        outState.putInt("count", count);  
    }  
}
```

실행 결과

