

# 2017 LINC+ 사업단 사회맞춤형 JAVA 실무인력양성과정 안드로이드 앱 개발 기초

컴퓨터공학부  
강승우

## 6. 액티비티와 인텐트

2018.02.09

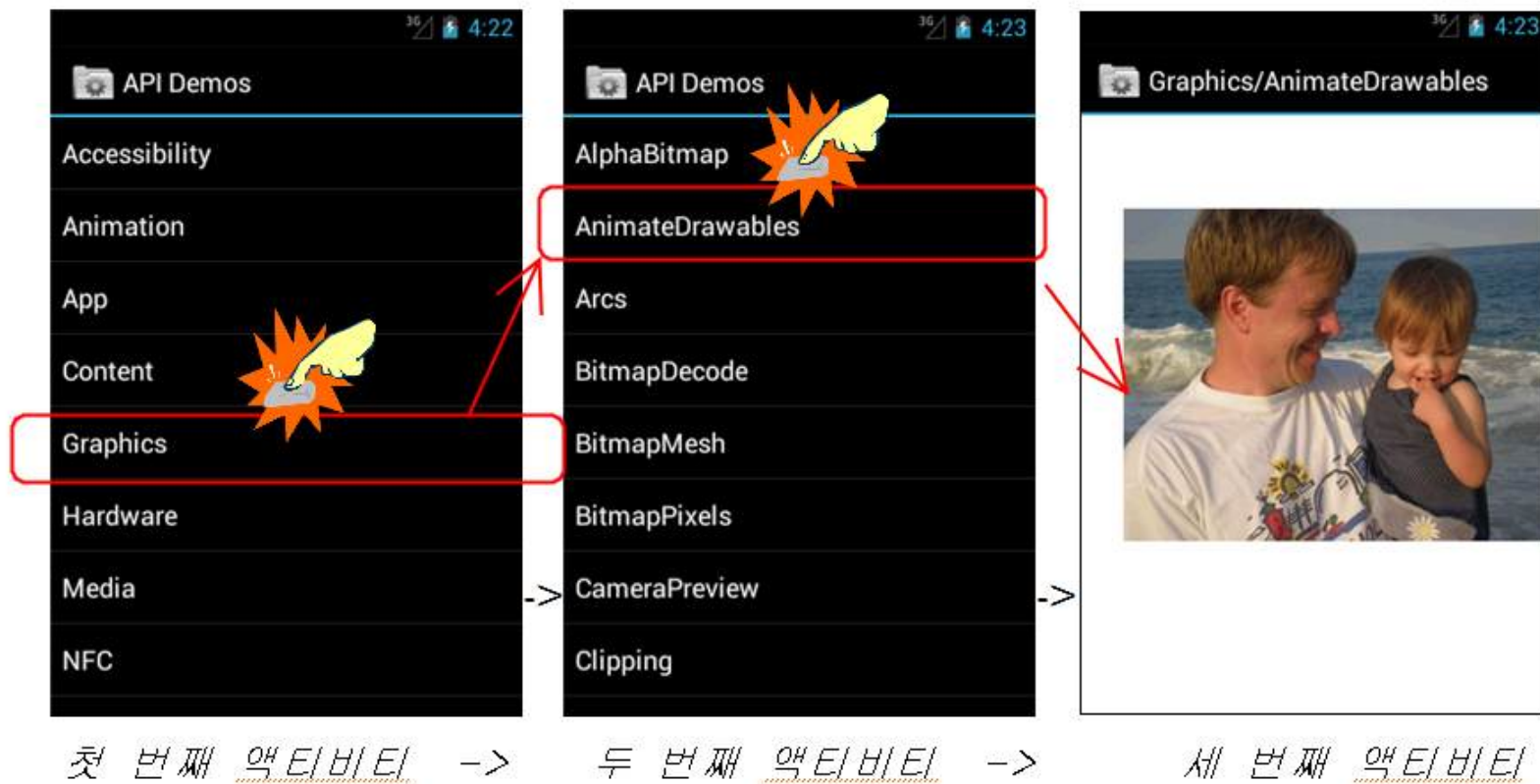
# 안드로이드 애플리케이션 요소

- Activity
  - Service
  - Broadcast Receiver
  - Content Provider
- 
- 보통 한 개 이상의 액티비티(Activity)로 구성

# 액티비티 (Activity)

- 일반적으로 UI를 갖는 하나의 스크린을 나타냄
- 안드로이드 애플리케이션은 반드시 하나의 “launcher” (or “main”) activity를 가지고 있어야 함
  - Launcher activity: 애플리케이션의 시작 포인트가 되는 액티비티
- 각 액티비티는 AndroidManifest.xml 파일에 등록되어 있어야 함
- 하나 이상의 View를 가질 수 있음
- Activity lifecycle callback 메소드를 통해 실행이 관리됨

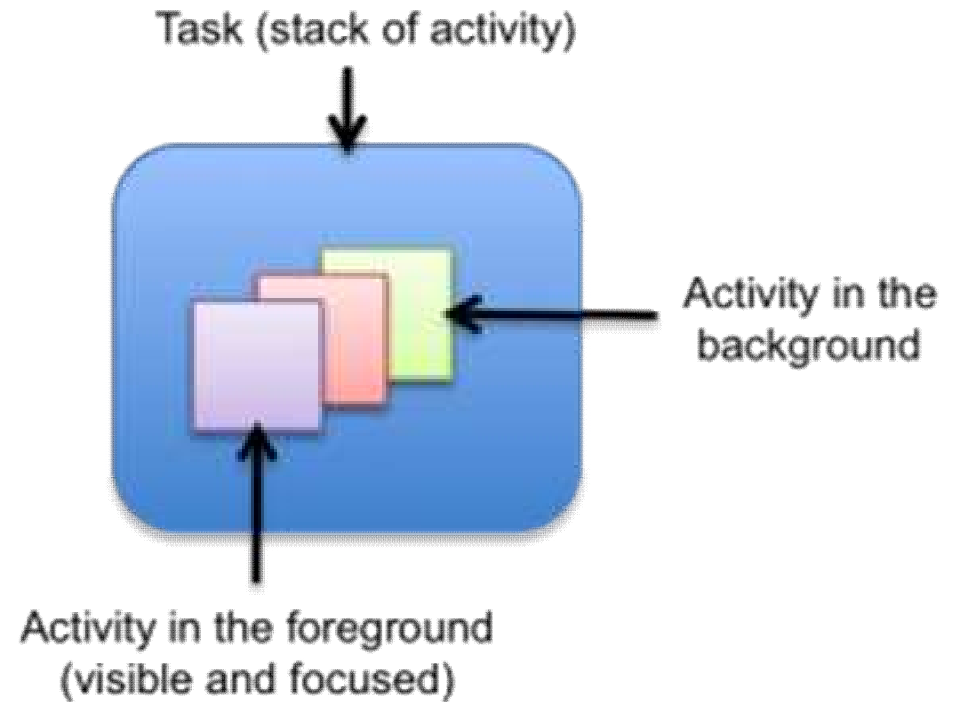
# 여러 개의 액티비티로 구성된 앱의 예



# Task

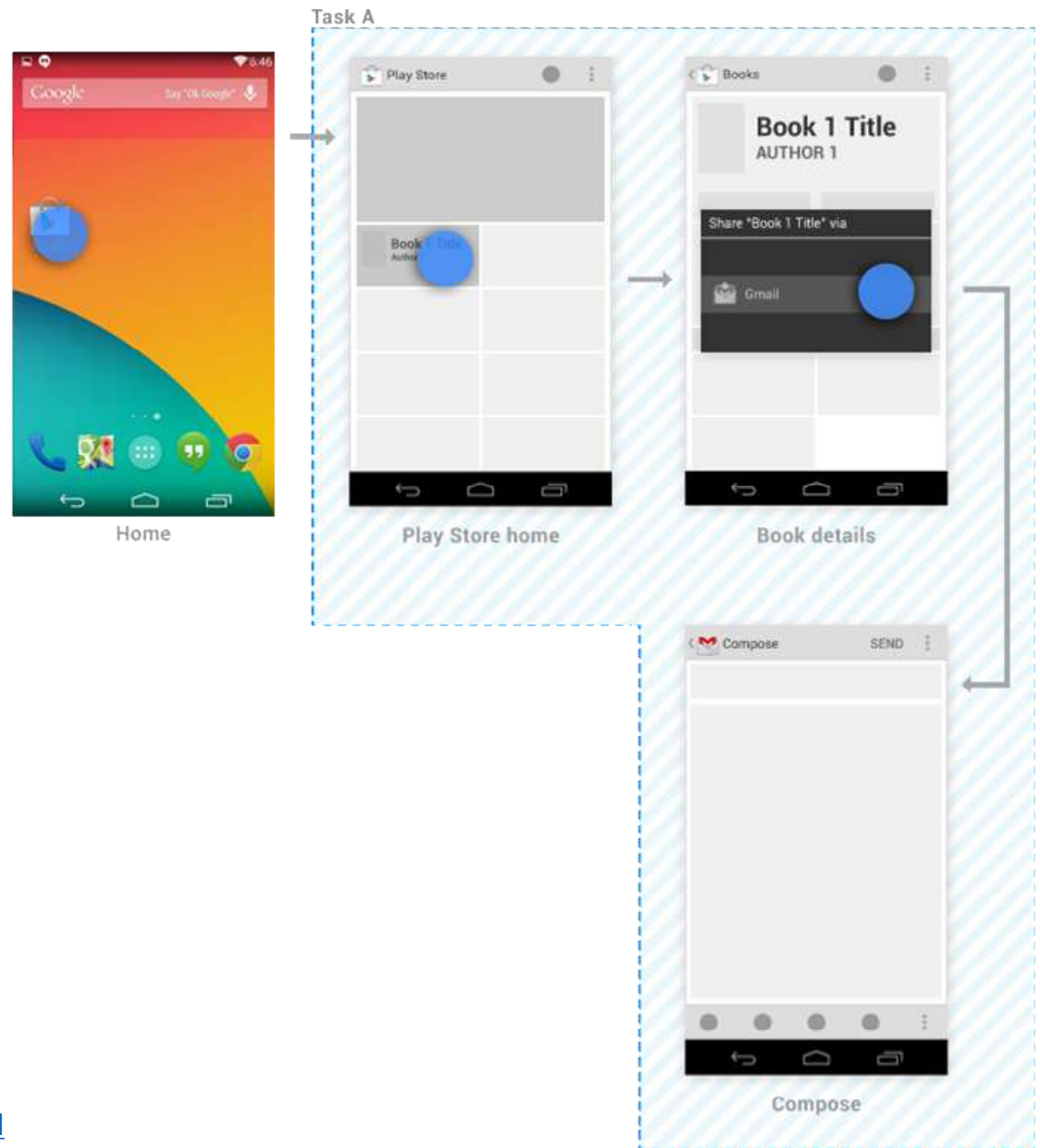
- 일련의 실행 흐름에 속한 액티비티들의 그룹
  - 사용자가 수행하고자 하는 작업을 위해 따르는 액티비티의 시퀀스
  - 서로 다른 애플리케이션의 액티비티도 하나의 태스크에 속할 수 있음

- Activity의 Stack으로 관리 됨
  - “Back Stack”

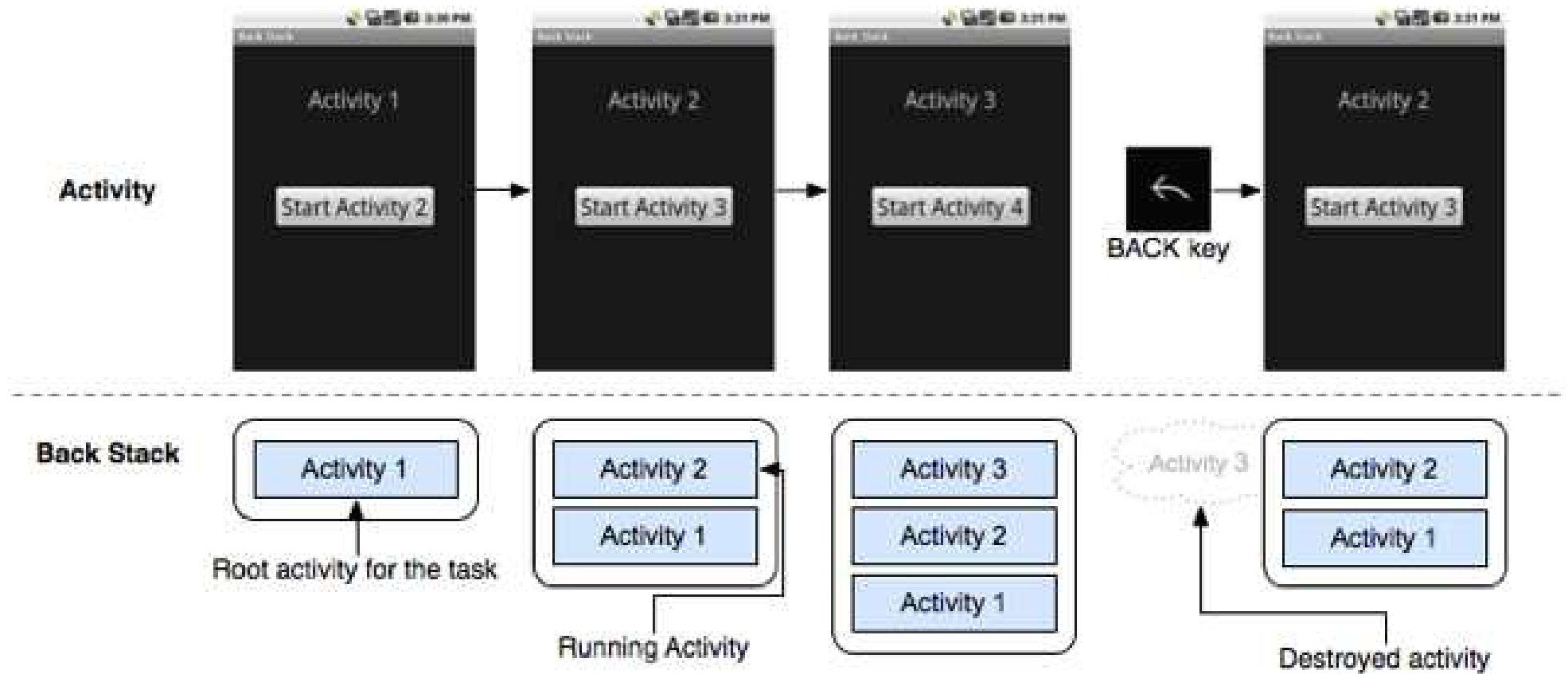


# Task 예제

- 서로 다른 애플리케이션과 액티비티의 실행 흐름



# 액티비티 스택



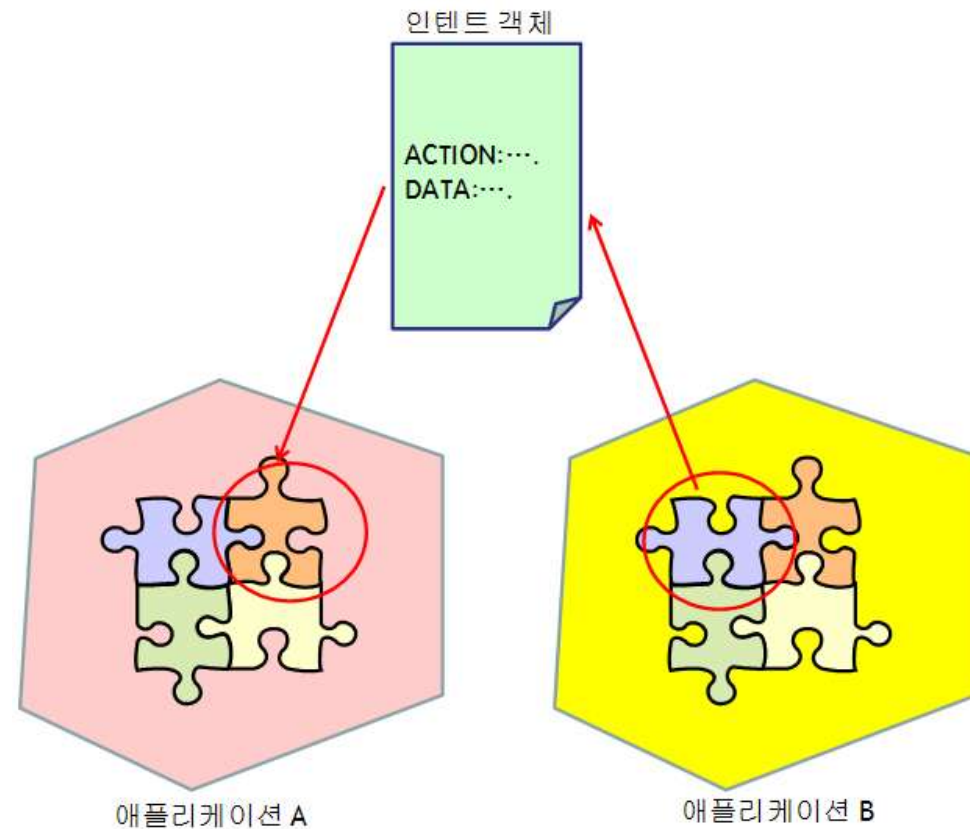


# 액티비티 전환

- 한 액티비티(화면)에서 다른 액티비티(화면)으로 전환하려면 어떻게 해야 할까?

# 액티비티 전환

- 한 액티비티에서 다른 액티비티를 시작하려면 실행하려는 액티비티의 시작에 필요한 정보를 보내주어야 함
  - 이때 인텐트 (Intent)를 사용함



# 인텐트 (Intent)

- 다른 application component가 어떤 action을 수행하도록 요청하기 위해 사용하는 메시징 객체
- 대표적인 use cases
  - Activity를 시작
  - Service를 시작
  - Broadcast를 전달
- 종류
  - 명시적 인텐트 (Explicit intent): 실행할 component 이름을 명확하게 지정
  - 암시적 인텐트 (Implicit intent): 수행해야 하는 일반적인 액션만 선언 (지도 상의 위치를 표시하고자 할 때 지도를 보여줄 수 있는 component면 어떤 것이라도 좋다)

<https://developer.android.com/guide/components/intents-filters.html>

<https://developer.android.com/reference/android/content/Intent.html>

# 명시적 인텐트(Explicit Intent) 이용 예

- 코드 예제

```
Intent intent = new Intent(this, NextActivity.class);  
startActivity(intent);  
// Executed in an Activity, so 'this' is the Context
```

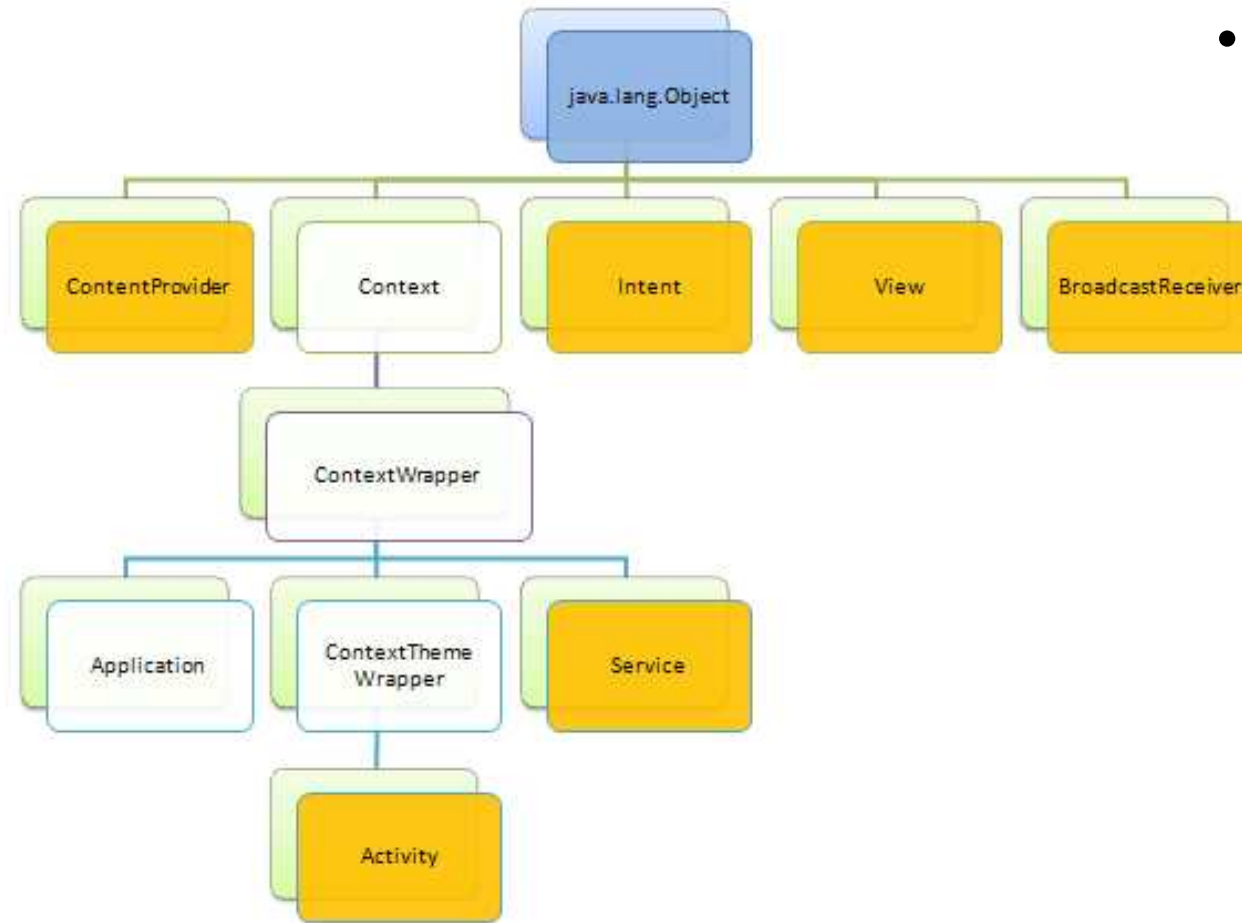
- Intent(Context, Class)

- 특정 component(class 객체)에 대한 인텐트 생성

- startActivity(Intent)

- Intent 객체에 명시된 액티비티를 시작
- 액티비티가 종료될 때 아무런 반환값을 받지 않게 됨

# Context



- Context 클래스

- 애플리케이션 환경에 대한 정보에 접근할 수 있도록 해주는 통로 역할
  1. 시스템이 관리하는 애플리케이션 관련 정보(애플리케이션의 특정 리소스나 클래스)에 접근
    - `getPackageName()`, `getResource()`
  2. 애플리케이션 레벨의 작업을 수행하기 위한 API 호출
    - 안드로이드 시스템 서비스 API
    - `startActivity()`, `bindService()`

# Context

- Context 구분

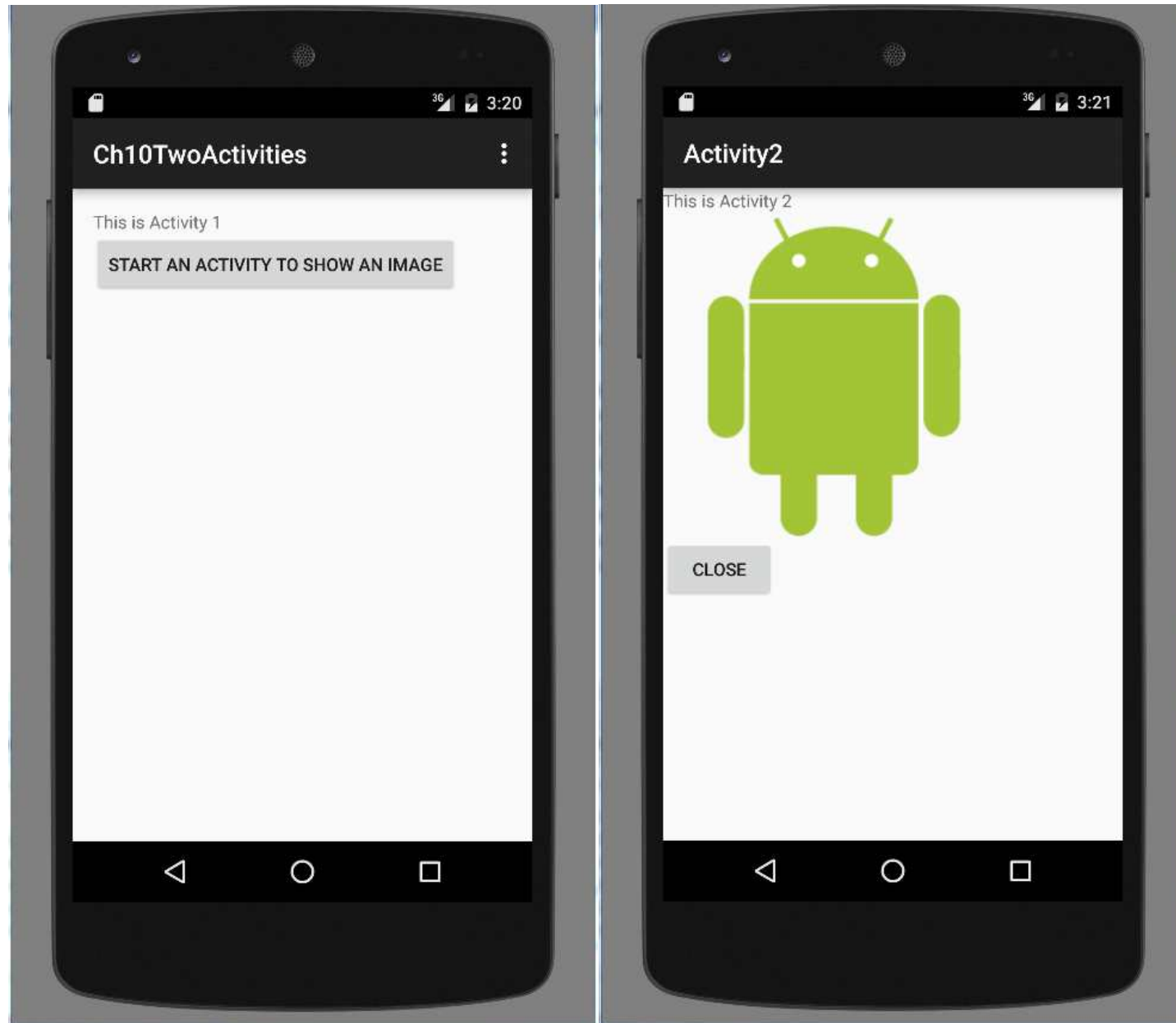
- Application Context

- Application life-cycle에 접목되는 개념
    - 하나의 애플리케이션이 실행되어 종료될 때까지 유지
    - Application context 객체 참조: `getApplicationContext()`, `getApplication()`
    - 현재 활성화된 activity만이 아닌 어플리케이션 전체에 대한 context가 필요한 경우에 사용

- Activity Context

- Activity life-cycle에 접목되는 개념이다.
    - 액티비티가 소멸되는(`onDestroy()`) 경우 사라질 수 있는 객체
    - Activity context 객체 참조: `ActivityName.this`, `getBaseContext()`
    - 다이얼로그는 activity context를 통해서만 만들 수 있음

- 두 개의 Activity가 있는 애플리케이션 만들기
- 예제 프로젝트 이름:  
04\_Activity\_Intent/Ch10TwoActivities



# 예제 소스 구조

- activity\_main.xml
  - TextView / Button
- MainActivity.java
  - Button 객체 onClick 메소드
    - Intent 객체 생성
    - startActivity() 호출

- activity\_second.xml
  - TextView / ImageView / Button
- SecondActivity.java
  - Button 객체 onClick 메소드
    - finish() 호출

- AndroidManifest.xml
  - <activity> element 추가: SecondActivity 설정



# 액티비티 결과 받기

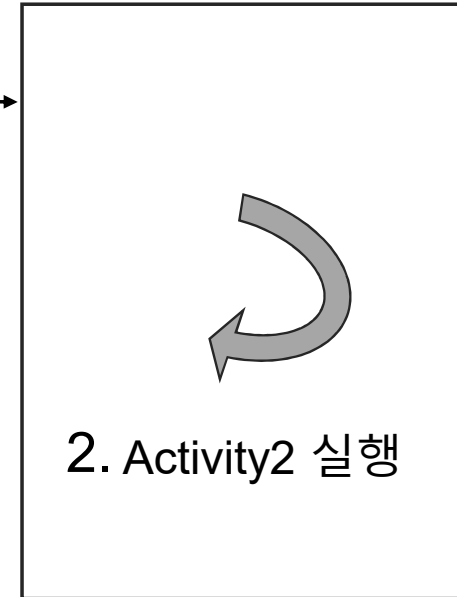
Activity1 (main activity)



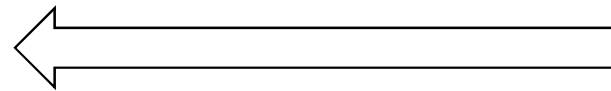
start

1. Activity1에 의해  
Activity2가 시작

Activity2



2. Activity2 실행



3. Activity2의 실행 종료 시  
Activity1에 결과를 전달

How?

# 액티비티 결과 받기를 위한 메소드

다른 activity 실행 시 호출해야 하는 메소드 (startActivity() 메소드 대신 사용)

- `startActivityForResult(Intent intent, int requestCode)`
  - `intent`
    - 실행하고자 하는 액티비티 정보를 담고 있는 인텐트 객체
  - `requestCode`
    - 시작된 액티비티가 종료될 때 `onActivityResult()` 메소드에서 반환됨

실행 시킨 activity로부터 결과를 받기 위해 구현해야 하는 메소드

- `onActivityResult(int requestCode, int resultCode, Intent data)`
  - `requestCode`
    - `startActivityForResult()` 호출할 때 넘겨준 request code
  - `resultCode`
    - 실행된 액티비티에서 `setResult()` 메소드 호출을 통해 반환되는 result code
  - `data`
    - 부가 데이터를 담는 Intent 객체

# 액티비티 결과 받기 예

- SubActivity를 시작한 MainActivity의 경우

```
...  
public class MainActivity extends Activity {
```

```
...  
// 서브 액티비티를 시작한다.
```

```
Intent in = new Intent(MainActivity.this, SubActivity.class);  
startActivityForResult(in, COMMAND);
```

```
@Override
```

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
```

```
... // 여기서 값을 전달받는다.
```

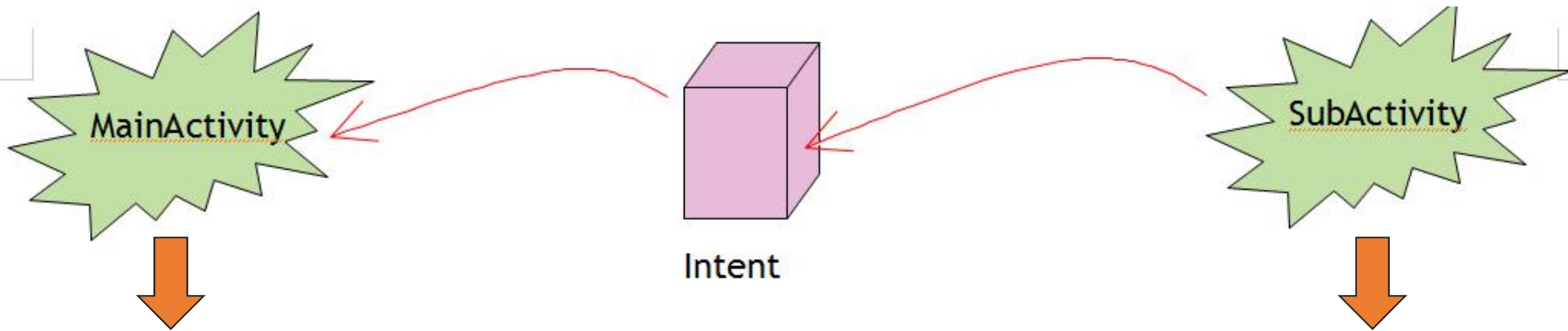
```
}
```

```
}
```

SubActivity

결과값을 전달한다.

# 인텐트를 통해 데이터 전달하기



```
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)
```

# 데이터 전달 방법

- 결과를 전달하고자 하는 SubActivity의 경우

1. 전달할 결과를 담을 Intent 객체 생성

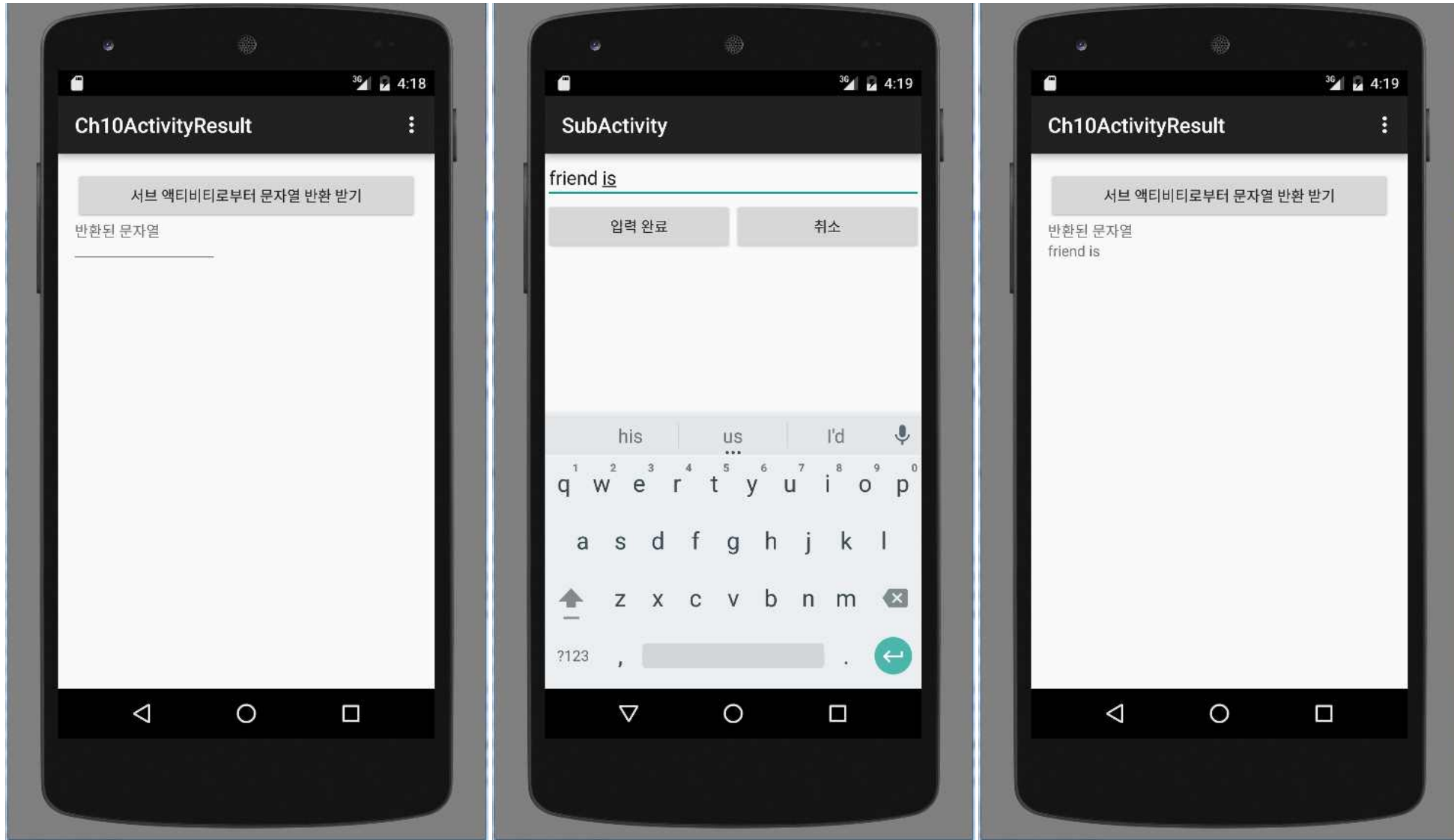
2. Intent 객체에 데이터 저장

3. Intent 객체를 전달할 결과로 설정

- 코드 예

```
Intent intent = new Intent();  
  
intent.putExtra("INPUT_TEXT", edit.getText().toString());  
  
setResult(RESULT_OK, intent);
```

- 액티비티 결과 받기 예제 프로젝트 이름: 04\_Activity\_Intent/Ch10ActivityResult



# 예제 소스 구조

- activity\_main.xml
  - Button / TextView / TextView
- MainActivity.java
  - Button onClick() 메소드
    - Intent 객체 생성
    - startActivityForResult() 호출
    - onActivityResult() 구현
      - 결과로 반환된 Intent의 문자열을 TextView의 텍스트로 설정

- activity\_sub.xml
  - EditText / Button / Button
- SubActivity.java
  - Button\_ok onClick() 메소드
    - Intent 객체 생성 / EditText에 입력된 문자열을 intent에 넣음 / setResult() 호출 / finish() 호출
  - Button\_cancel onClick() 메소드
    - setResult() 호출 / finish() 호출

- AndroidManifest.xml
  - <activity> element 추가: SubActivity 설정

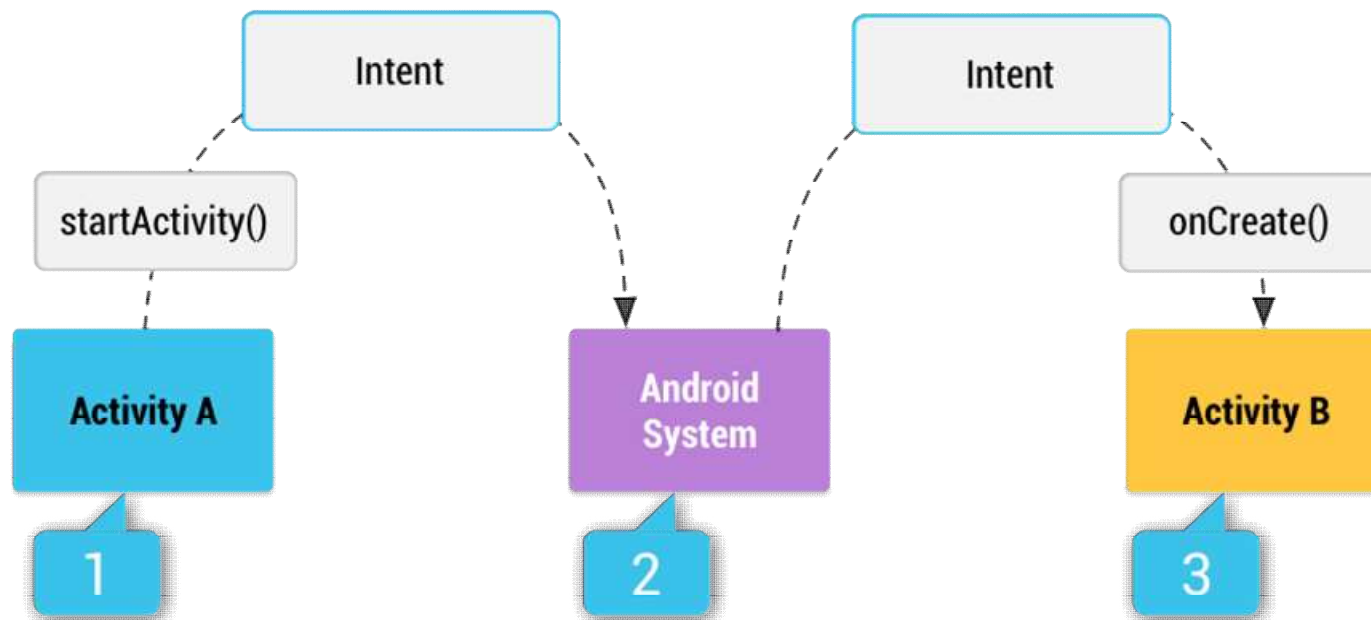
# 암시적 인텐트 (Implicit Intent)

- 내가 작성한 애플리케이션의 경우는 어떤 component(activity, service, ..)가 어떤 기능을 수행하는지 내가 알고 있다
  - 명시적 인텐트를 사용할 수 있음
  - 즉, 실행하고자 하는 component의 이름을 지정할 수 있음
- 그러나 내가 만들지 않은 애플리케이션의 경우는?  
→ 암시적 인텐트 사용
- 암시적 인텐트
  - 일반적으로 다른 애플리케이션의 component를 구동하는데 사용
  - 이메일 전송, 전화 걸기, 연락처 보기, 웹 브라우저 보기, 지도 보기 등
  - 안드로이드에 내장된 기능 대부분을 인텐트를 통해 사용 가능



# 암시적 인텐트의 사용

- 암시적 인텐트를 사용하면
  - 사용자가 요구하는 기능에 가장 적합한 component를 안드로이드 시스템이 검색하여 실행한다



1: Activity A에서 원하는 액션을 기술한 인텐트를 생성하여 `startActivity` 호출

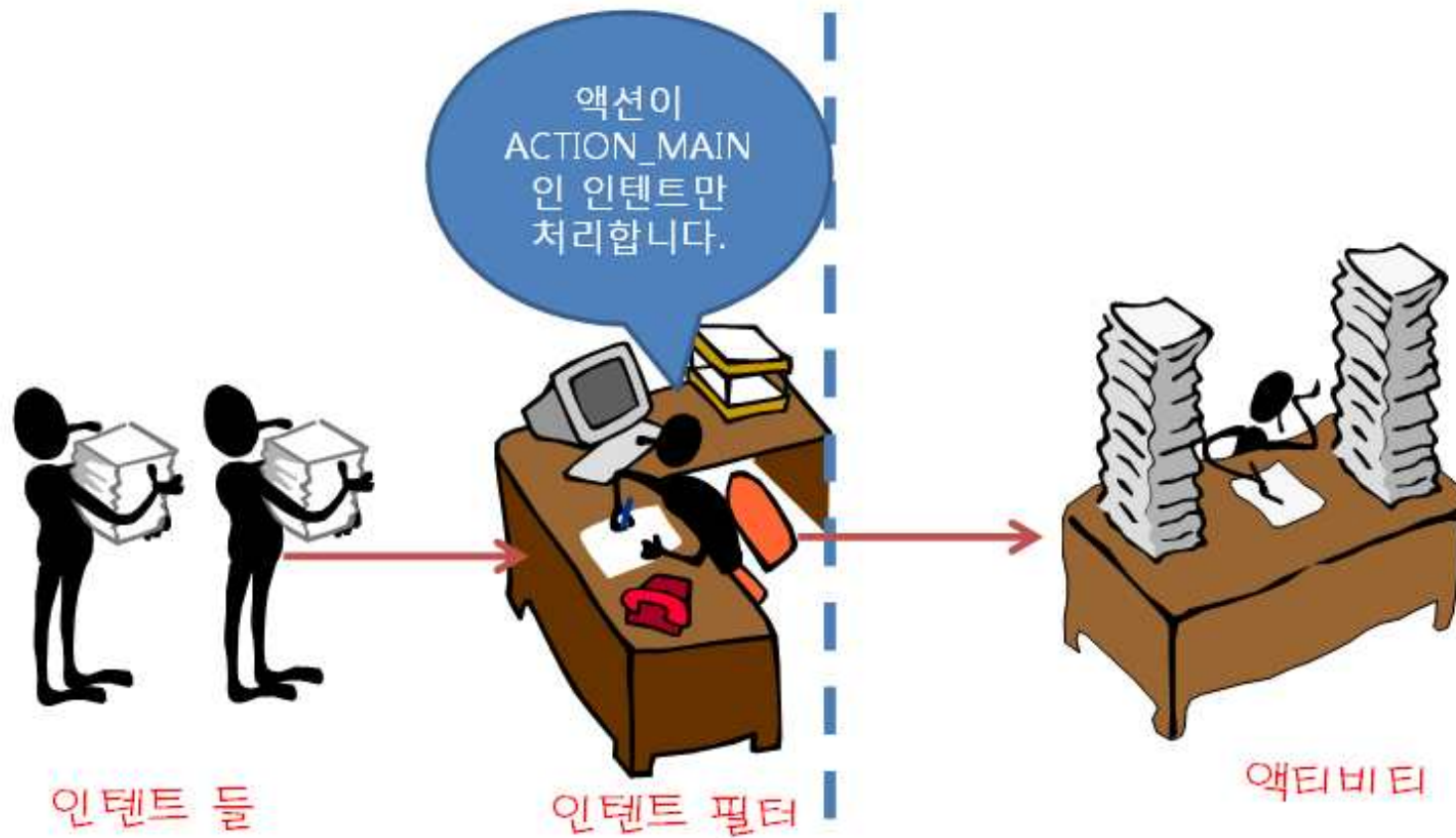
2: 안드로이드 시스템이 설치된 애플리케이션 패키지를 검색하여 해당 인텐트를 처리할 수 있는 액티비티를 찾음

3: 해당 액티비티가 Activity B라면 `onCreate` 메소드를 호출하고 인텐트를 넘김

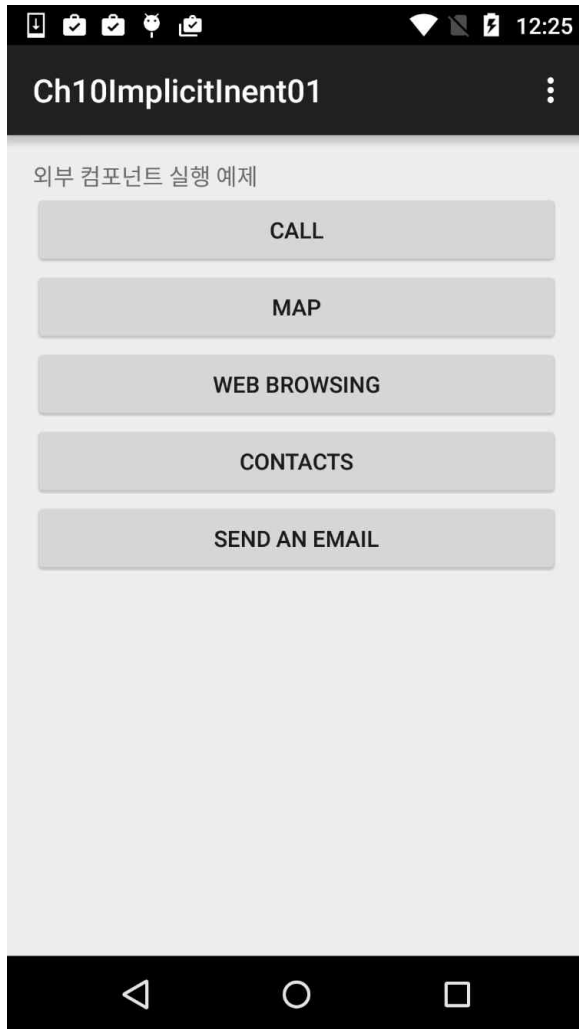
# 인텐트 필터 (Intent filter)

- 안드로이드 시스템이 적합한 component를 어떻게 찾을까?
- 각 component는 자신이 처리할 수 있는 인텐트의 종류를 안드로이드 시스템에 알려주어야 한다
  - 이를 위해 인텐트 필터를 기술한다
    - 해당 component의 기능 기술
    - component가 수신할 수 있는 인텐트의 집합 기술

# 인텐트 필터



# 암시적 인텐트 사용 예제



- 다른 애플리케이션의 기능 사용
  - 전화
  - 지도
  - 웹 브라우저
  - 연락처
  - 이메일
- 예제 프로젝트 이름
  - 04\_Activity\_Intent/Ch10ImplicitIntent01

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/an
droid"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView android:text="외부 컴포넌트 실행 예제"
android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
    <Button
        android:layout_width="match_parent"
android:layout_height="wrap_content"
        android:text="Call"
        android:id="@+id/call"
        android:onClick="onClick"/>
    <Button
        android:layout_width="match_parent"
android:layout_height="wrap_content"
        android:text="Map"
        android:id="@+id/map"
        android:onClick="onClick"/>
```

```
<Button
    android:layout_width="match_parent"
android:layout_height="wrap_content"
    android:text="Web browsing"
    android:id="@+id/web"
    android:onClick="onClick"/>
<Button
    android:layout_width="match_parent"
android:layout_height="wrap_content"
    android:text="Contacts"
    android:id="@+id/contact"
    android:onClick="onClick"/>
<Button
    android:layout_width="match_parent"
android:layout_height="wrap_content"
    android:text="Send an email"
    android:id="@+id/email"
    android:onClick="onClick"/>

</LinearLayout>
```

버튼 누를 때  
호출

```
public void onClick(View v){
```

```
    Intent intent = null;
```

```
    switch (v.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;
```

```
        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;
```

```
        case R.id.email:
```

```
            intent = new Intent(Intent.ACTION_SENDTO, Uri.parse("mailto:swkang@koreatech.ac.kr"));
```

```
            break;
```

```
    }
```

```
    if(intent != null) {
```

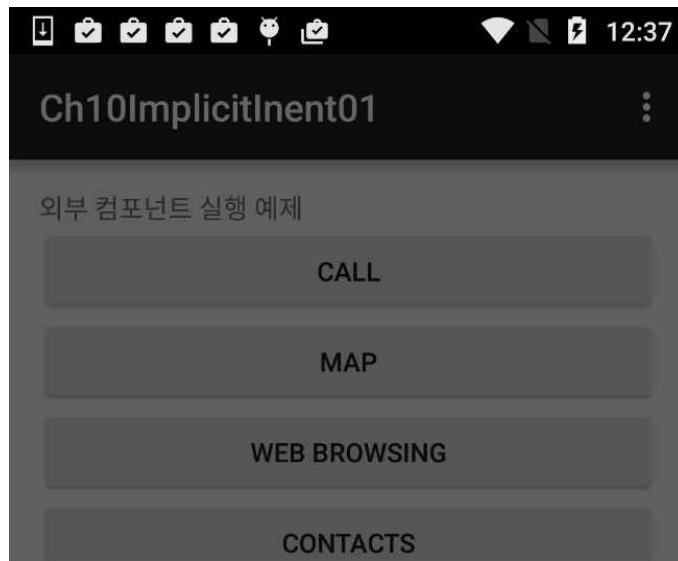
```
        if(intent.resolveActivity(getPackageManager()) != null) { startActivity(intent); }
```

```
    }
```

null이 아닌 경우 인텐트를 처리할 앱이 최소한 하나는 있다는 의미

선택된 버튼의  
기능에 따라  
적합한 Intent  
객체 생성

생성된 Intent  
객체를 이용  
startActivity  
호출



지도(으)로 열기

한 번만

항상

다른 앱 사용

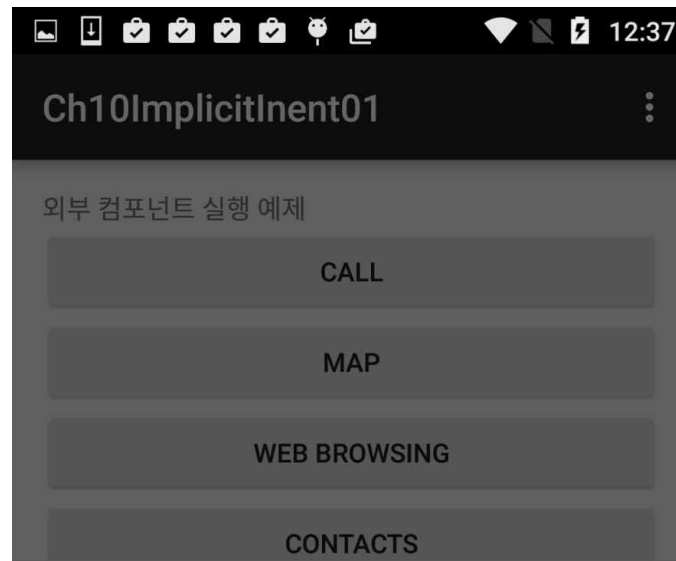


네이버 지도



다음지도

Map 버튼을  
누른 상황



Dolphin(으)로 열기

한 번만

항상

다른 앱 사용



Opera



Chrome

Web browsing  
버튼을 누른  
상황

- 인텐트에 기술된 해당 액션을 수행할 수 있는 애플리케이션이 여러 개가 있는 경우 사용자가 선택할 수 있도록 팝업이 표시됨

# 암시적 인텐트 사용

- 인텐트에 필요한 핵심 정보
  - 액션: 실행을 요청할 작업
  - 데이터: 액션 실행 시 필요한 데이터
  - 그 외: 데이터 타입, 카테고리, 엑스트라, 패키지
- Intent(String action, Uri uri)
  - action
    - ACTION\_VIEW, ACTION\_DIAL, ACTION\_MAIN, ... 등 여러 액션이 Intent 클래스에 String 상수로 정의되어 있음
  - uri
    - 액션 수행 시 필요한 데이터를 Uri 클래스의 객체로 제공
- <https://developer.android.com/reference/android/content/Intent.html>
- <https://developer.android.com/guide/components/intents-filters.html>

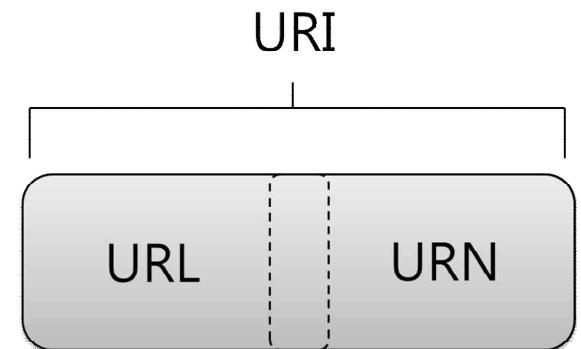


# 인텐트 액션 종류

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. (앱의 시작점이 되는 메인 액티비티)
ACTION_CALL	액티비티	전화 통화를 시작한다.
ACTION_SYNC	액티비티	모바일 장치의 데이터를 서버 상의 데이터와 일치시킨다.
ACTION_DIAL	액티비티	전화 번호를 누르는 화면을 표시한다.

# URI

- URI (Uniform Resource Identifier)
  - 어떤 자원에 접근하기 위한 유일한 식별자 (주소, 키)
  - 자원: 웹 페이지, 텍스트, 비디오 또는 사운드 클립, 이미지 등
  - 일반적으로 많이 사용되는 URI
    - URL (Uniform Resource Locator): 브라우저 주소창에 입력하는 웹 페이지 주소
    - URL은 보통 프로토콜, 도메인, 자원의 위치정보로 구성
  - URL, URN을 포괄하는 개념
    - URN (Uniform Resource Name): 자원의 위치와 무관하고 영속적이며 유일한 자원의 이름
  - [scheme://host:port/path](#)
- 예제
  - [http://www.w3.org/Icons/WWW/w3c\\_main.gif](http://www.w3.org/Icons/WWW/w3c_main.gif)
  - <ftp://file.com:9000>
  - <mailto:name@email.com>



# 안드로이드 URI 객체 생성

- Uri.parse(String uriString)
  - 주어진 URI 인코딩된 문자열에 대해서 Uri 객체를 생성
  - <https://developer.android.com/reference/android/net/Uri.html>
- URI scheme 예제
  - http:
  - mailto:
  - file:
    - 파일 위치
  - tel:
    - 전화 걸기 (전화번호를 기술)
  - content:
    - 연락처와 같은 데이터 위치 (Content Provider가 관리하는 데이터를 기술)
  - geo:
    - 지도 상에 위치 표시
    - geo:latitude, longitude?z=zoom
    - 설정된 위도, 경도를 지도의 중앙에 위치, z로 줌 레벨 설정 (옵션). 0(전세계) -21(개별 건물)

# 액션과 Uri로 표현된 데이터의 예

- 
- ▷ 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.

Google Maps와 관련된 인텐트는 아래 웹 페이지 참고

<https://developers.google.com/maps/documentation/android-api/intents>

# 인텐트 데이터 설정

- setData, getData 메소드를 사용하여 따로 할 수도 있음
  - 인텐트 객체에 데이터를 설정하고 접근할 수 있음

```
intent = new Intent(Intent.ACTION_VIEW, Uri.parse("geo:37.30, 127.2?z=10"));
```



```
intent = new Intent(Intent.ACTION_VIEW);  
intent.setData(Uri.parse("geo:37.30, 127.2?z=10"));
```