여러 액티비티 활용

코틀린 안드로이드 앱 개발

강사: 배정만

구성 요소 활성화

- 응용 프로그램은 매니패스트 파일에 의해 결함된 구성요소의 모음
- 각 구성요소는 메세지를 전달하여 활성화
- 액티비티를 표시하기 위해서는 메세지를 런타임에 전달하고 활성화 해야함

두번째 액티비티

```
class FirstActivity : AppCompatActivity(), View.OnClickListener {
    override fun onClick(v: View) {
        val second = SecondActivity() // 작동하지 않음
    }
}

class SecondActivity : AppCompatActivity()
```

- 위 액티비티를 활성화 하려면
    1. Intent 객체 생성
    2. 할 것 또는 어떻게 할 지 등을 지정
    3. 인텐트를 런타임에 보내고 컴포넌트 활성화

인텐트(Intent)의 역할

- 명시적인 의도로 다른 액티비티 시작하기
- 한 액티비티에서 다른 액티비티로 데이터 전달
- 두번째 액티비티에서 호출한 액티비티로 데이터 반환
- 액티비티의 라이프 사이클 함수 포함
- 약간의 Fragments

인텐트 전달
두번째 액티비티 실행

```
class FirstActivity : AppCompatActivity(), View.OnClickListener {
    override fun onClick(v: View) {
        val intent = Intent(this, SecondActivity::class.java)
        startActivity(intent)
    }
}

class SecondActivity : AppCompatActivity()
```

- 인텐트 객체 생성
  - Intent intent = new Intent(<Context>, <Target>);
  - 컨텍스트: 현재 활성화 런타임의 정보묶음, 생성하고자 하는 액티비티 컴포넌트
- 런타임에 인텐트 전달
  - startActivity(intent);

실습 - 두번째 액티비티 실행

1. 빈 액티비티 2개를 만듬
   - MainActivity, SecondActivity
2. MainActivity에 버튼을 추가하고 클릭 이벤트 등록
3. 클릭 이벤트에 SecondActivity를 생성하는 인텐트(Intent)를 만들고 런타임에 전달

실습 - 첫번째 액티비티로 돌아가기

1. SecondActivity에 close 버튼을 추가
2. 클로즈 버튼에 클릭 이벤트 등록
3. 클로즈 이벤트 발생시 액티비티 종료
   - finish()

매니페스트.XML

```
<manifest package=""
          xmlns:android="http://schemas.android.com/apk/res/android">
  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
    <activity android:name=".SecondActivity">
    </activity>
  </application>
</manifest>
```

Main 레이아웃 XML

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="96dp"
    android:onClick="launchSecondActivity"
    android:text="Launch Sub Activity"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:layout_editor_absoluteY="96dp"/>
```

MainActivity

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    fun launchSecondActivity(v: View) {
        val i = Intent(this, O02ClickEventActivity::class.java)
        startActivity(i)
    }
}
```

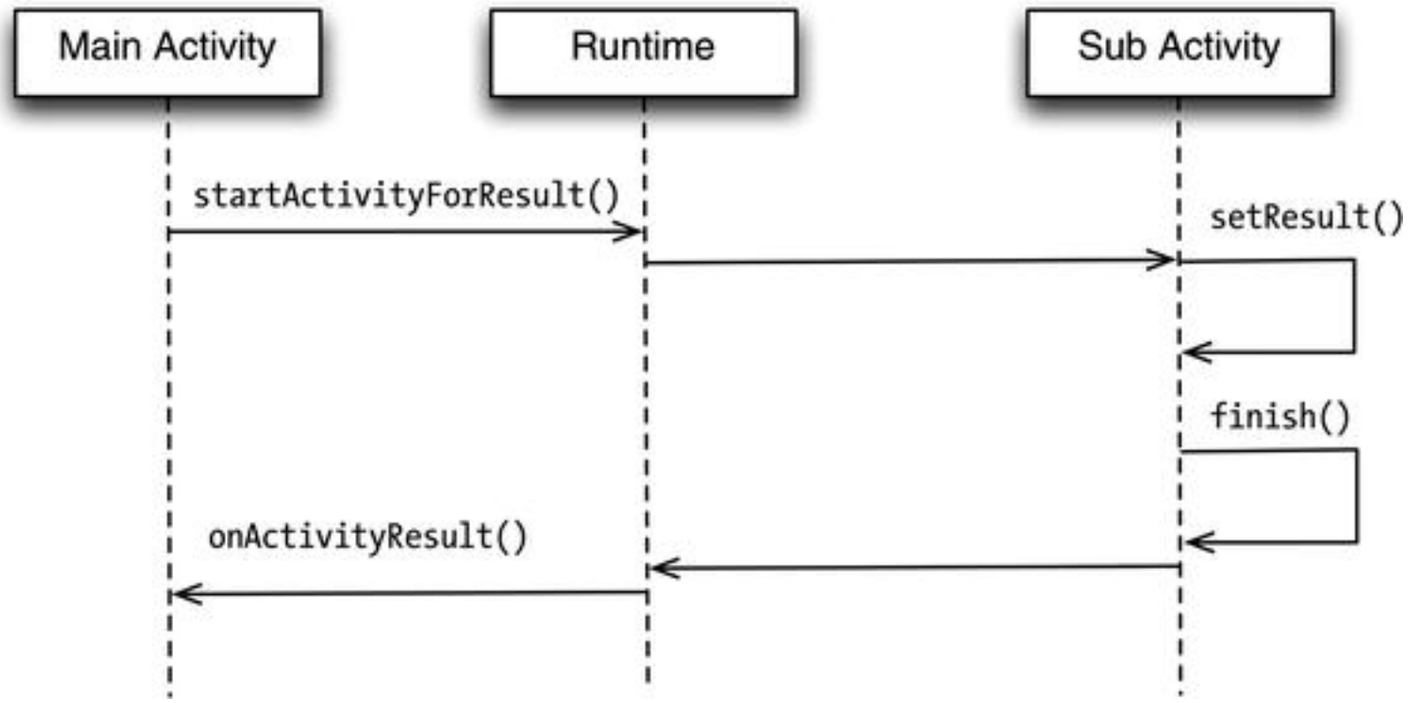
Second 레이아웃 XML

```
?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".SecondActivity">
  <Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="68dp"
    android:text="Close"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:layout_editor_absoluteY="68dp"/>
</android.support.constraint.ConstraintLayout>
```

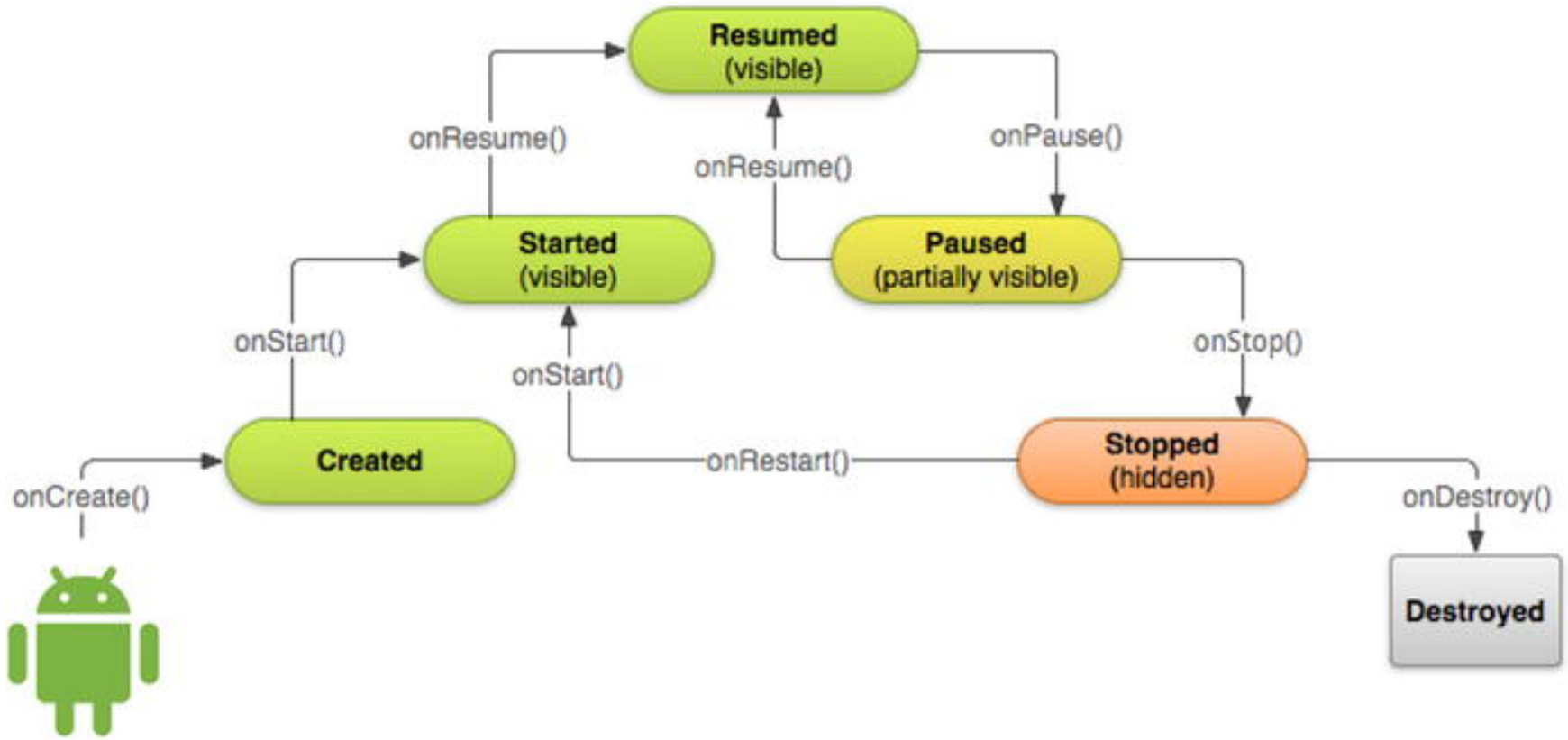
SecondActivity

```
class SecondActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_o01_hello)
        val secondButton =
findViewById<Button>(R.id.button2)
        secondButton.setOnClickListener { finish() }
    }
}
```

이벤트 시퀀스


Main Activity
Runtime
Sub Activity
startActivityForResult()
setResult()
finish()
onActivityResult()

액티비티 생명주기


Resumed
(visible)
onResume()
onResume()
onPause()
Started
(visible)
Paused
(partially visible)
onStart()
onStart()
onStop()
Created
onRestart()
Stopped
(hidden)
onCreate()
onDestroy()
Destroyed

실습 - Implicit Intents

```
Uri uri = Uri.parse("http://www.naver.com");
intent.setAction(Intent.ACTION_VIEW);
startActivity(intent);

uri = Uri.parse ( "geo : 40.7113399, -74.0263469");
intent.setAction(Intent.ACTION_VIEW);
startActivity(intent);

uri = Uri.parse ( "tel : 639285083333");
intent.setAction(Intent.ACTION_DIAL);
startActivity(intent);
```

- HTTP 요청 열기
- 맵 열기
- 다이얼 열기

intent.setAction (Intent.ACTION_VIEW):

- 인텐트의 액션을 설정하면 Android 런타임에서 디바이스 내의 어떤 애플리케이션이 요청을 가장 잘 처리 할 수 있는지 선택
- ACTION _VIEW 는 Intent 클래스에 정의 된 상수 중 하나이며 웹 URL을 처리하려는 경우 사용
- 가장 일반적인 Intent 동작에 대한 자세한 정보는 공식 Android 웹 사이트 참고
  - https://developer.android.com/guide/components/intents-common