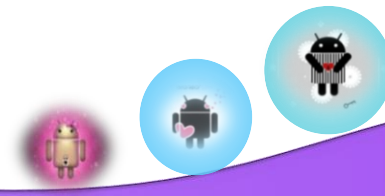
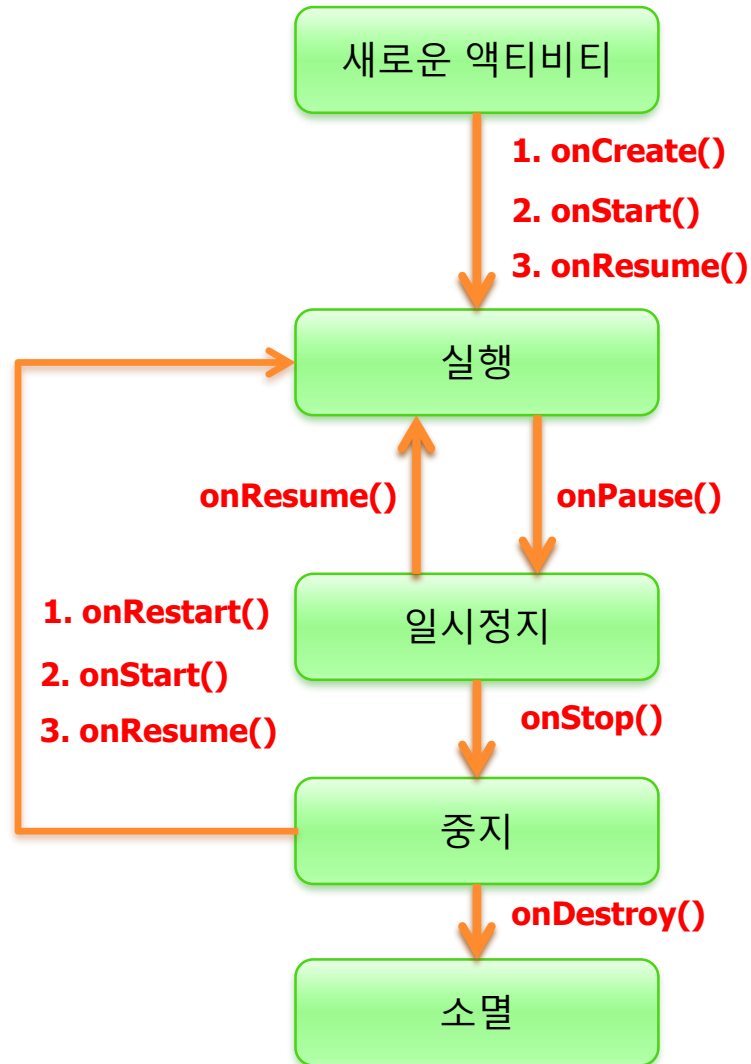




# 생명주기 (LIFECYCLE)





onCreate()	<ul style="list-style-type: none"> <li>• 액티비티가 처음에 만들어졌을 때 호출</li> <li>• 화면에 보이는 뷰들의 일반적인 상태를 설정하는 부분</li> <li>• 이전 상태가 저장되어 있는 경우에는 번들 객체를 참조하여 이전 상태 복원 가능</li> <li>• 이 메소드 다음에는 항상 onStart() 메소드가 호출됨</li> </ul>
onStart()	<ul style="list-style-type: none"> <li>• 액티비티가 화면에 보이기 바로 전에 호출</li> <li>• 액티비티가 화면 상에 보이면 이 메소드 다음에 onResume() 메소드가 호출</li> <li>• 액티비티가 화면에서 가려지게 되면 이 메소드 다음에 onStop() 메소드가 호출</li> </ul>
onResume()	<ul style="list-style-type: none"> <li>• 액티비티가 사용자와 상호작용하기 바로 전에 호출</li> </ul>
onRestart()	<ul style="list-style-type: none"> <li>• 액티비티가 중지된 이후에 호출되는 메소드로 다시 시작되기 바로 전에 호출</li> <li>• 이 메소드 다음에는 항상 onStart() 메소드가 호출</li> </ul>
onPause()	<ul style="list-style-type: none"> <li>• 또 다른 액티비티를 시작하려고 할 때 호출</li> <li>• 저장되지 않은 데이터를 저장소에 저장하거나 애니메이션 중인 작업을 중지하는 등의 기능을 수행하는 메소드</li> <li>• 이 메소드가 리턴하기 전에는 다음 액티비티가 시작될 수 없으므로 이 작업은 매우 빨리 수행된 후 리턴되어야 함.</li> <li>• 액티비티가 이 상태에 들어가면 시스템은 액티비티를 강제 종료할 수 있음</li> </ul>
onStop()	<ul style="list-style-type: none"> <li>• 액티비티가 사용자에게 더 이상 보이지 않을 때 호출됨</li> <li>• 액티비티가 소멸되거나 또 다른 액티비티가 화면을 가릴 때 호출됨</li> <li>• 액티비티가 이 상태에 들어가면 시스템은 액티비티를 강제 종료할 수 있음</li> </ul>
onDestroy()	<ul style="list-style-type: none"> <li>• 액티비티가 소멸되어 없어지기 전에 호출</li> <li>• 이 메소드는 액티비티가 받는 마지막 호출이 됨</li> <li>• 액티비티가 애플리케이션에 의해 종료되거나(finish() 메소드 호출) 시스템이 강제로 종료시키는 경우에 호출될 수 있음</li> <li>• 위의 두 가지 경우를 구분할 때 (isFinishing() 메소드 호출) 시스템</li> <li>• 액티비티가 이 상태에 들어가면 시스템은 액티비티를 강제 종료할 수 있음</li> </ul>

# 프로젝트 생성

4



Application name: **LifeCycle**

Company Domain: **admid.example.com**

Project location: **C:\Android\StdWork\lifecycle**

Target Android Devices

- **Phone and Tablet**
- **Minimun SDK API 17 : Android 4.0.3**

Add an activity to Mobile : **Empty Activity**

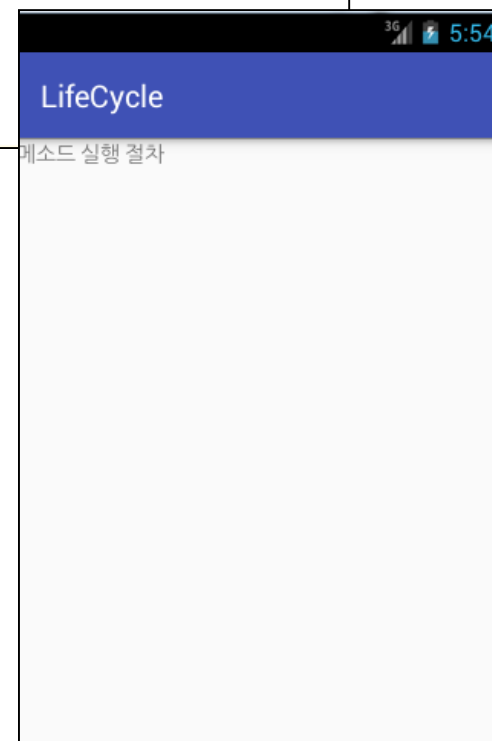
Customize the Activity

Activity Name: **MainActivity**

Layout Name: **activity\_main**



```
activity_main.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">  
  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="메소드 실행 절차" />  
  
</LinearLayout>
```



MainActivity.java x

```
1 package com.example.admin.firstmessage;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13        Toast.makeText(getApplicationContext(),
14            "onCreate", Toast.LENGTH_SHORT).show();
15    }
16
17    @Override
18    protected void onDestroy() {
19        Toast.makeText(getApplicationContext(),
20            "onDestroery()", Toast.LENGTH_SHORT).show();
21        super.onDestroy();
22    }
```

MainActivity.java x

```
23
24     @Override
25     protected void onResume() {
26         Toast.makeText(getApplicationContext(),
27             "onResume()", Toast.LENGTH_SHORT).show();
28         super.onResume();
29     }
30
31     @Override
32     protected void onStart() {
33         Toast.makeText(getApplicationContext(),
34             "onStart()", Toast.LENGTH_SHORT).show();
35         super.onStart();
36     }
37
38     @Override
39     protected void onStop() {
40         Toast.makeText(getApplicationContext(),
41             "onStop()", Toast.LENGTH_SHORT).show();
42         super.onStop();
43     }
44
45     @Override
46     protected void onPause() {
47         Toast.makeText(getApplicationContext(),
48             "onPause()", Toast.LENGTH_SHORT).show();
49         super.onPause();
50     }
51 }
```

