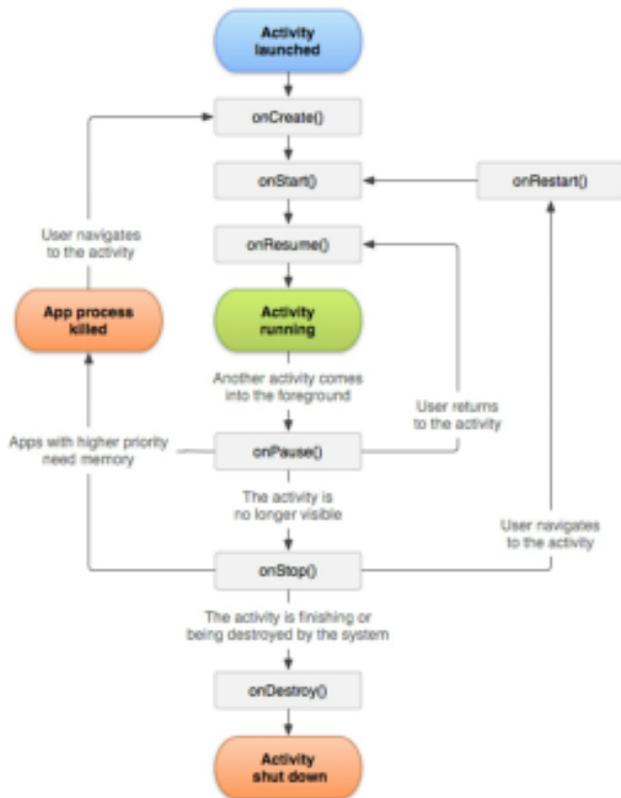


## 08/03 안드로이드 교육 - 3 (패스트캠퍼스)

**Activity** - 어플리케이션 내 하나의 화면이라고 생각

- 어플리케이션(App)의 구성요소
- 어플리케이션은 여러개의 Activity로 구성되어 있음
- Activity는 또 다른 Activity를 시작하게 할 수 있음 - 화면연결
- 새로운 Activity가 시작되면, 이전 Activity는 중단되지만 'Stack (스택)'에 보존되어 있음

\*Activity의 LifeCycle



- 앱이 실행 (Activity launched)되면, onCreate() - onStart() - onResume() 단계를 거쳐야 사용자의 Input을 받을 수 있는 Activity running 상태에 진입
- Activity running : 앱을 사용할 때 화면이 보이는 상태
- onPause() - onStop() - onDestroy 단계를 거쳐 앱이 종료

LifeCycle의 3가지 구분

- (1) 전체 수명 : onCreate()와 onDestroy() 사이 (onCreate에서 설정을 수행한 다음, 나머지 리소스를 모두 onDestroy에서 해제해주어야 함)
- (2) 가시적 수명 : onStart()와 onStop() 사이 (사용자가 Activity와 상호작용 할 수 있는 시기, 사용자가 Activity에 Input을 넣을 수 있는 시기)
- (3) 전경 수명 : onResume()과 onPause() 사이 (현재 Activity가 다른 Activity 앞에 표시되고, 사용자 입력도 집중됨)

Activity 상태저장의 4가지 단계 (다음 Activity로 넘어갔다가 이전 Activity로 돌아와도 데이터가 남아있을 수 있는 이유)

- (1) onSaveInstanceState()를 구현하여 Activity 상태에 관한 중요정보를 보존할 수 있음
- (2) Bundle로 저장하고 보존하고 있음

(3) onCreate 또는 onRestoreInstanceState()에 전달이 됨

(4) 두 함수는 받은 Bundle을 열어서 이전의 상태를 복구함

#### onCreate

- Activity가 처음 생성되었을 때 호출

- 정적 설정을 모두 수행하여야 함 (정적 설정: Activity가 커졌을 때 한번 설정하면 이후에 변할 일이 없는 설정들을 의미)

- 뒤에 항상 onStart()가 호출됨

- setContentView()를 이곳에서 호출해야 Activity의 사용자 인터페이스 레이아웃을 정의할 수 있음

(setContentView() 메소드에게 작성한 XML 파일을 전달하여 사용자 인터페이스 레이아웃을 정의함)

#### onStart

- Activity가 사용자에게 화면을 보여줄 준비가 되었을 때 호출

#### onResume

- Activity가 시작되고 사용자와 상호작용하기 직전에 호출

#### Activity Running

- Activity가 사용자에게 화면을 보여줄 준비가 되었을 때 호출

- 사용자와 Activity와 상호작용이 가능한 시기

#### onPause

- 화면의 일부가 가려진 상태

- 시스템이 다른 Activity를 재개하기 직전에 호출

- 데이터를 유지하기 위해 저장되지 않은 정보를 저장하는 용도로 사용

- 해당 메소드의 리턴 값이 반환이 될 때까지 다음 Activity가 실행되지 않기 때문에 빠르게 종료시켜야 함

- 화면이 다시 보여지게 되면 onResume()이 호출되고, 사라지게 되면 onStop()이 호출됨

#### onStop

- Activity가 더이상 사용자에게 보여지지 않는 상태

- 사용자에게 다시 보여지게 되면 onStart()가 호출되고, 사라지게 되면 onDestroy()가 호출됨

#### onDestroy

- Activity가 소멸되기 직전에 호출

- Activity의 마지막 Life cycle

- 의도적으로 종료가 되었을 수도 있고 (다른 Activity로 전환), 시스템 리소스를 절약하기 위해 종료되었을 수도 있음

개발을 할 때, 원하는대로 작업이 잘 돌아가는지 중간 작업을 확인하기 위해 Log를 찍을 수 있음

JAVA: System.out.println("text")

Android: Log.d("tag", "msg")

안드로이드 스튜디오 하단 Logcat 탭을 클릭하여 확인 - 검색 탭에 tag 값을 입력하여 원하는 부분만 확인 가능