Activity vs Fragment

Most Important things of android UI

Activity

활동 소개

`Activity` 클래스는 A
니다. `main()` 메서드를
는 특정 콜백 메서드를 호

활동의 개념

모바일 앱 환경은 사용자와 앱의 상호작용이 항상 동일한 위치에서 시작되는 것이 아니라는 점에서 데스크톱 앱 환경과 다릅니다. 대신 사용자 여정은 흔히 비결정론적으로 시작됩니다. 예를 들어 홈 화면에서 이메일 앱을 열면 이메일 목록이 표시될 수 있습니다. 이에 반대로 소셜 미디어 앱을 사용하고 있는 상태에서 이메일 앱을 실행하면 이메일을 작성하기 위한 이메일 앱 화면으로 바로 이동할 수 있습니다.

Android UI의 핵심

`Activity` 클래스는 이 패러다임을 촉진하도록 설계되었습니다. 한 앱이 다른 앱을 호출할 때 호출 앱은 다른 앱을 전체적으로 호출하는 것이 아니라 다른 앱의 활동을 호출합니다. 이런 방식으로 활동은 앱과 사용자의 상호작용을 위한 진입점 역할을 합니다. 활동은 `Activity` 클래스의 서브클래스로 구현됩니다.

Activity는 App의 UI를 그리는 창을 제공

다른 언어의 main()과 같은 너낌

활동은 앱이 UI를 그리는 창을 제공합니다. 이 창은 일반적으로 화면을 채우지만 화면보다 작고 다른 창 위에 떠 있을 수 있습니다. 일반적으로 한 활동은 앱에서 하나의 화면을 구현합니다. 예를 들어 앱의 활동 중 하나는 *환경설정* 화면을 구현하고 또 다른 활동은 *사진 선택* 화면을 구현할 수 있습니다.

But 생명 주기의 아무곳이나 호출해

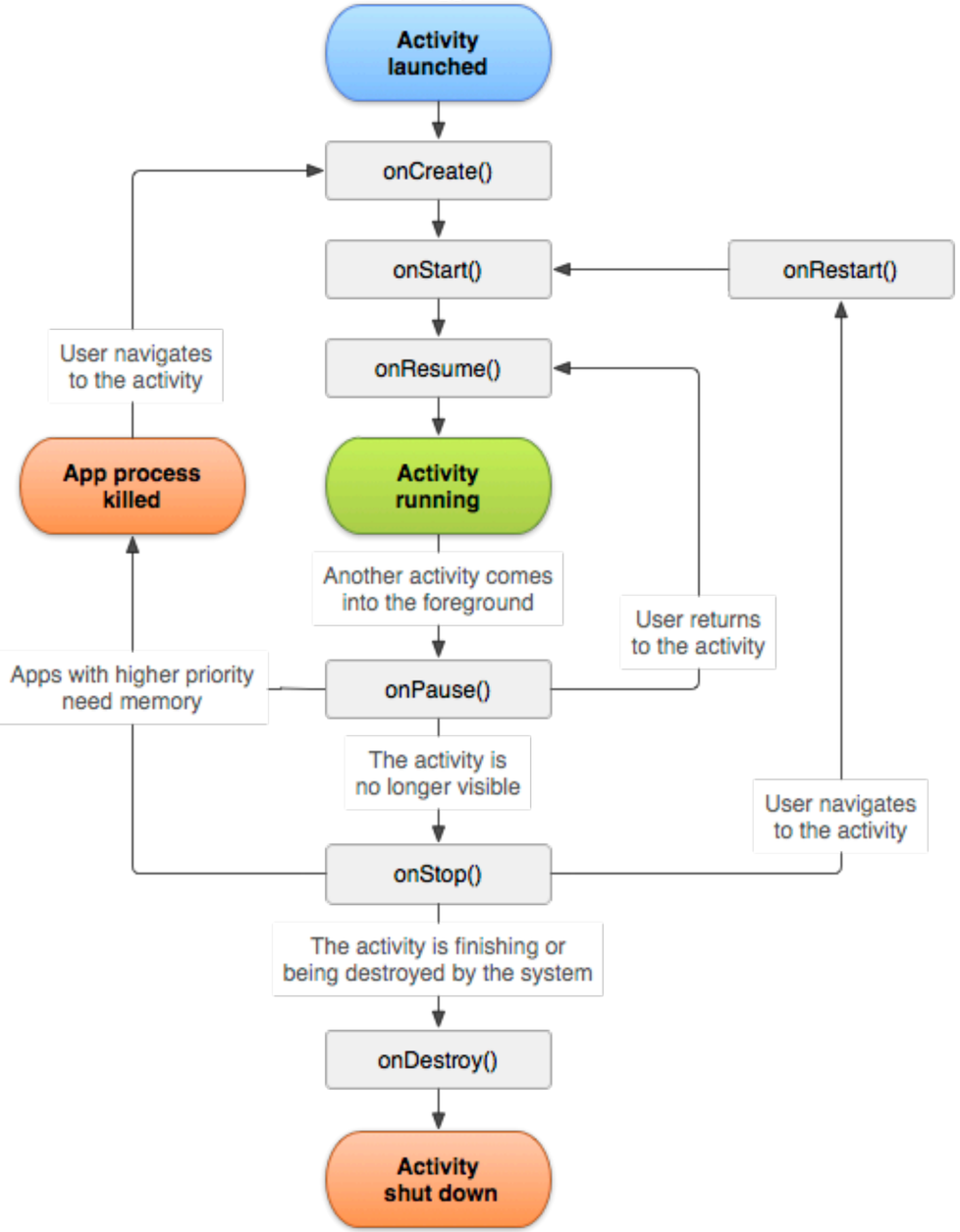
Activity 실행 가능

대부분의 앱에는 여러 화면이 포함되어 있습니다. 즉, 대부분의 앱은 여러 활동으로 구성됩니다. 일반적으로 앱에서 하나의 활동이 *기본 활동*으로 지정되며 이 기본 활동은 사용자가 앱을 실행할 때 표시되는 첫 번째 화면입니다. 각 활동은 다양한 활동을 실행하기 위해 또 다른 활동을 시작할 수 있습니다. 예를 들어 간단한 이메일 앱의 기본 활동은 이메일 받은편지함을 표시하는 화면을 제공할 수 있습니다. 여기에서 기본 활동은 이메일 작성 및 개별 이메일 열기와 같은 작업을 위한 화면을 제공하는 다른 활동들을 실행할 수 있습니다.

활동이 앱의 일관된 사용자 환경을 형성하기 위해 함께 작동하지만 각 활동은 다른 활동에 단지 느슨하게 결합됩니다. 일반적으로 앱의 활동 간에는 최소한의 종속성만 있습니다. 실제로 활동은 흔히 다른 앱에 속하는 활동을 시작합니다. 예를 들어 브라우저 앱은 소셜 미디어 앱의 공유 활동을 실행할 수 있습니다.

앱의 활동을 사용하려면 앱의 manifest에 활동 관련 정보를 등록하고 활동 수명 주기를 적절히 관리해야 합니다. 이 문서의 나머지 부분에서는 이러한 주제를 소개합니다.

Activity 생명주기



onCreate() : 가장 먼저 실행되는 생명주기, main() 너낌
여기서 setContentView() 같은 연결구를 써주면 된다.

onResume() : 사용자와 상호작용 하는 상태.

onPause() : 전화가 오거나 배터리 부족 등 방해하는 상황에 호출된다.
그 후 바로 onPause()를 호출함

onStop() : onPause에서 사용자에게 더이상 표시되지 않으면 호출됨.
새로운 엑티비티가 화면을 다 차지한 경우 등

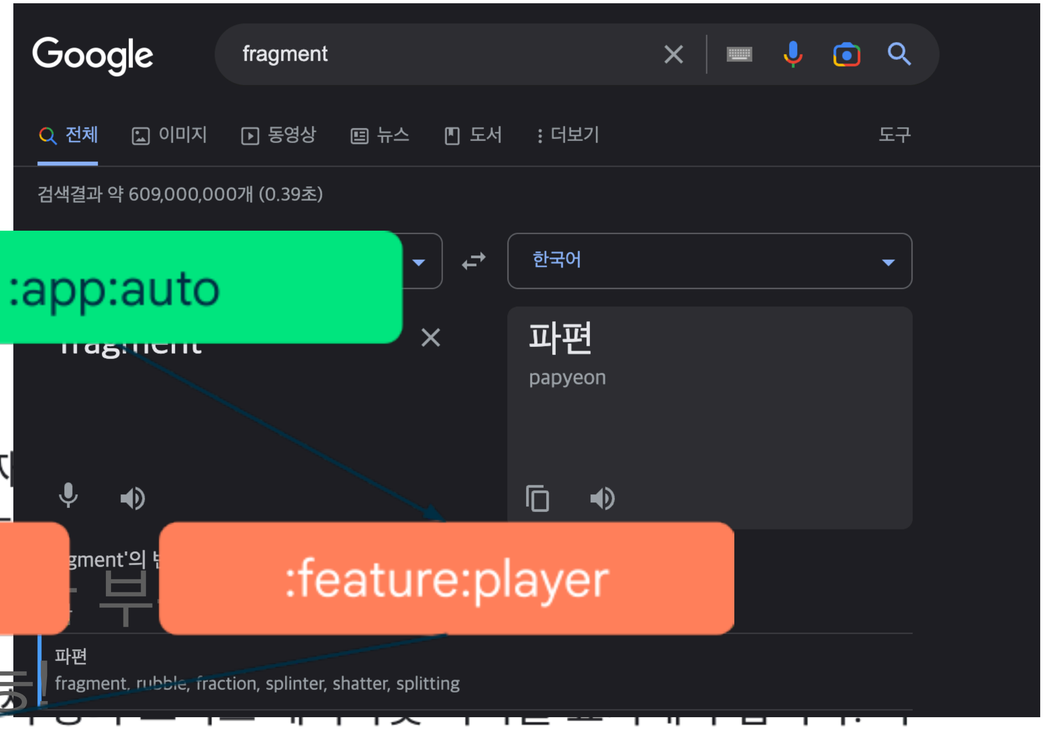
onDestroy() : Activity가 소멸되기 직전, finish()를 사용한 경우 호출됨
이런 경우 말고 화면 회전, 멀티 윈도우 모드를 사용한 경우에도 호출되고
다시 onCreate가 호출됨.

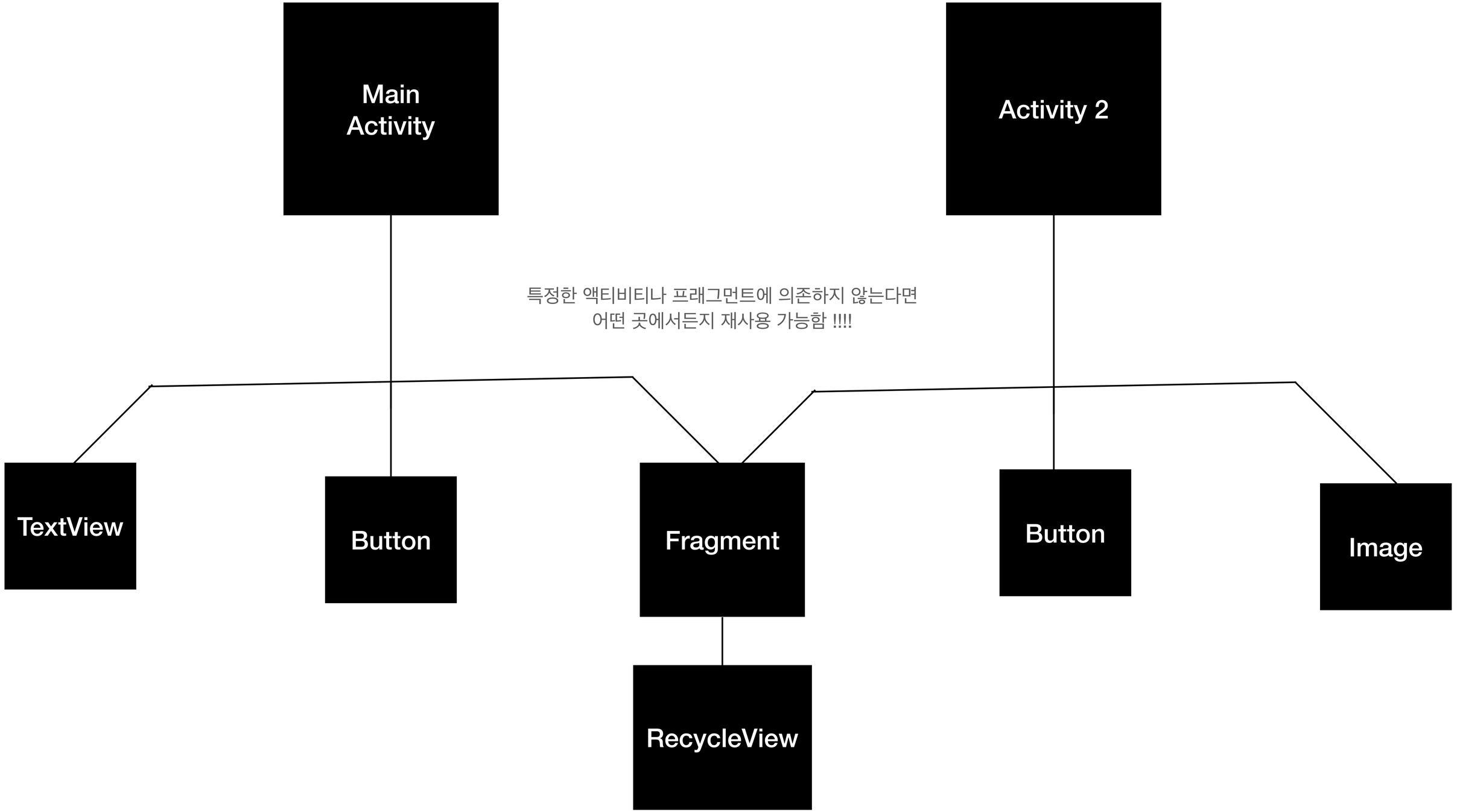
Fragment

모듈성

프래그먼트는 UI를 개별 청크로 분할할 수 있도록 하여 활동의 UI에 모듈성과 재
스(예: 탐색 흐
는 데 더 적

다양한 화면 크기에 반응하는 앱을 생각해 보세요. 큰 화면에서는 앱이 왼쪽 탭
은 화면에서는 앱이 하단 탐색 메뉴와 선형 레이아웃 목록을 표시해야 합니다. 활동에서 이러한 모든 변형을 관리하는 작업은 어려울
수 있습니다. 콘텐츠에서 탐색
동은 올바른 탐색 UI를 표시하
고 프래그먼트는 적절한 레이




Main Activity
Activity 2
특정한 액티비티나 프래그먼트에 의존하지 않는다면
어떤 곳에서든지 재사용 가능함 !!!!
TextView
Button
Fragment
Button
Image
RecycleView

Fragment 장점

액티비티에 비해 가벼움

생성, 파괴가 액티비티에 비해 자유롭다

코드의 모듈화가 가능하다. 확장성, 재사용성 증가 및 빌드시간 단축!

Fragment 단점

단독으로 존재 불가능!
반드시 액티비티 내에서 생성 되야뎀

Activity 자체도 언제 어디서 초기화, 파괴 될지
모르는데 Fragment 또한 독자적인 생명주기를 가지므로
제대로 생명주기를 이해하지 않으면 코드 짜기가 힘듬
-> Lifecycle of Components 를 사용하면 어느정도 해결

Fragment Manger를 직접 접근 하는 경우
예상하지 못한 동작을 할 수 있음