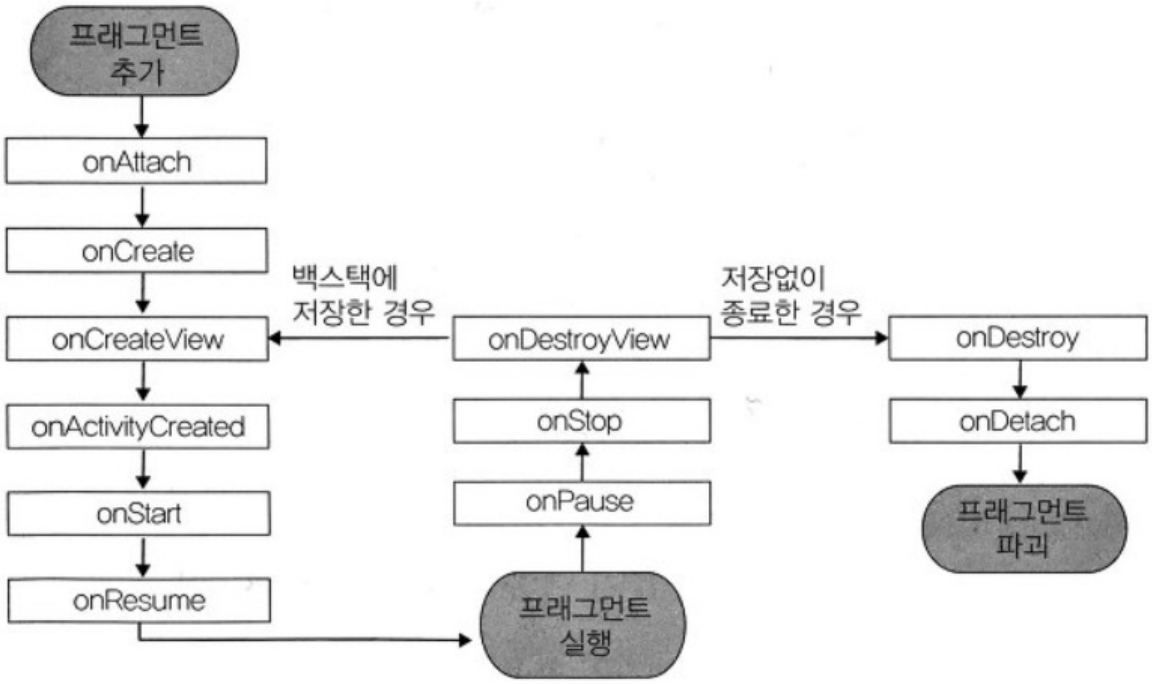
1. Fragment 란?

- Fragment 특징
  액티비티를 분할하여 화면의 한 부분을 정의한다.
  액티비티와 같이 레이아웃, 동작 처리, 생명주기를 가지는 독립적인 모듈이다.
  다른 액티비티에서도 사용 할 수 있어 재사용성이 뛰어나다.
  액티비티 내에서 실행 중에 추가, 제거가 가능하다.

- Fragment 생명주기



액티비티에 프래그먼트가 부착되고 초기화 되는 단계인 '프래그먼트 추가', 이후 정해진 동작을 수행하는 단계를 '프래그먼트 실행', 역할을 다하고 종료되는 시점은 '프래그먼트 파괴'. 이렇게 크게 세 단계로 구분 될 수 있습니다.

void onAttach(Activity activity)
프래그먼트가 처음으로 액티비티에 부착 될 때 호출됩니다. 앞서 말씀드렸듯이 프래그먼트는 필수적으로 하나의 액티비티에 종속되어야 합니다. 이 때문에 시작단계에서 가장 먼저 어떤 액티비티에 붙을것인지를 결정하게 되는것입니다. 매개변수로 종속될 액티비티를 전달받는것으로 보아 여러 액티비티에서 재사용 될 수 있음을 알 수 있습니다.

void onCreate(Bundle savedInstanceState)
이 콜백함수를 통해 프래그먼트에 꼭 필요한 요소들을 먼저 초기화 하고 시작 할 수 있습니다.
그리고 매개변수로 전달받는 savedInstanceState의 경우 프래그먼트가 재생성되기 이전의 상태를 저장하고 있는 변수로써,
이 값을 참조해 이전 내용을 복구하게 됩니다. 이는 Fragment BackStack과 연관이 되어있습니다.

View onCreateView (LayoutInflater inflater, View Group container, Bundle savedInstance)
프래그먼트에 쓰일 view 들을 정의하고 초기화 하게 됩니다. 이 때, 프래그먼트는 자신의 레이아웃을 루트 뷰로 설정하고 이를 inflate 하게 됩니다.
container 를 통해 프래그먼트가 액티비티의 어느 위치에 자리 잡아야 될지를 전달 받습니다.
지막으로 프래그먼트상에 생성된 뷰들을 종속 된 액티비티의 뷰(container) 에 리턴해 줌으로써 화면에 표시 되게 됩니다.

void onActivityCreated(Bundle savedInstanceState)
onCreate 함수의 역할과 같이 프래그먼트의 구성요소들을 초기화 하는 시점입니다.
차이점이 있다면 액티비티가 완전히 생성된 이후의 시점이기 때문에 액티비티의 컨트롤들에 접근하거나 프래그먼트의 구성요소들을 초기화 할 때 안전성을 보장받게 됩니다.

void onPause()
프래그먼트가 정지 되는 시점을 정의합니다. 프래그먼트의 정지가 반드시 소멸을 의미하는것은 아니지만 다시 해당 프래그먼트로 돌아온다는 보장도 없기 때문에 이 시점에서 남겨두어야 하거나 영구적으로 보존해야 할 자료들을 저장하게 됩니다.

- Oreo Notification 과 이전 버전 Notification 차이?

1. 권한
   개발자가 사용자에게 직접 권한을 요청해야하는데, 요청할 때 필요한 권한 그룹들의 목록을 모두 다 적어야한다.

예를 들어, 기존의 경우 WRITE_EXTERNAL_STORAGE 권한의 경우, 해당 권한을 허용하면 자동적으로 READ_EXTERNAL_STORAGE가 부여됐으나, 이제는 권한 요청시 배열안에 READ도 같이 넣어야한다.
같은 권한 그룹에 있다 하더라도, 필요한 권한은 모두 다 명시해서 적는 것이 중요하다.

2. 푸시
   푸시를 받았을 때, 사용자에게 보여지는 Notification에서 채널(Channel)이란 개념이 있어
   Oreo이상의 기기에서는 Channel을 하나 생성하고 create로 해주고 notificationBuilder의 생성자에 넣어주면 된다.

3. 백그라운드
   앱이 백그라운드 상태에 돌입하면 위치 정보 등에 접근 제어가 걸렸다.