

ViewBinding 기초

뷰 바인딩의 필요성

- 뷰의 요소를 불러오기 위해서 findViewById를 사용한다
- kotlin-android-extensions를 사용하면 findViewById를 생략하고 간편하게 쓸 수 있다
- 문제는 서로다른 xml에서 id를 동일하게 사용할 수 있기 때문에 kotlin-android-extensions를 통하면 코드가 헷갈릴 수 있다
- 그래서 구글에서는 안드로이드 스튜디오 4.1부터 kotlin-android-extensions의 지원을 중단하고 뷰 바인딩을 사용하도록 안내하고 있다.

뷰 바인딩이란

- 뷰 바인딩을 활성화하면 각 xml 파일에 대해 ViewBinding 클래스를 상속 받는 개별 뷰 바인딩 클래스가 자동으로 생성된다
- onCreate() 안에서 뷰 바인딩 클래스의 인스턴스를 생성
- 인스턴스가 뷰의 Id를 프로퍼티로 제공하게 된다

뷰 바인딩의 장점

- Null-safe: 뷰 바인딩은 서로 다른 layout의 같은 ID를 가진 뷰를 정확히 구분할 수 있다. 만약 그럴 수 없을 경우 @Nullable로 만들어 사용할 수 없게 한다.
- Type-safe: findViewById를 사용할 경우 뷰에 잘못된 타입을 지정할 우려가 있는데 뷰 바인딩에서는 그런 문제가 발생하지 않는다.