**Chapter2-2 정리**

1. R class

: R class는 app이 빌드될 때 생성되어진다. 또한 res directory에 있는 contents 뿐만 아니라 app의 모든 assets을 갖는다.

1. “@+id”

: 컴파일러에게 ID constant를 R class에게 추가하라고 전해주는 prefix이다.

1. View 와 viewGroup

: layout에서 작성된 view와 viewGroup은 계층구조를 이룬다.

1. Context

: 현재 app의 상태 정보와 통신을 위해 사용되는 object다.

1. CompileSdkVersion

: app이 컴파일 할 수 있는 API을 의미한다. 이를 통해 최신버전의 API에서 추가된 features를 app 내에서 사용할 수 있다. Gradle에 기본적으로 추가된 라이브러리들이 최신버전으로 업데이트 되면서 API버전도 같이 올라가게 되므로 항상 최신버전으로 유지하는 것이 좋다.

1. TargetSdkVersion

: 현재 테스트한 app API 버전을 의미한다. targetSdkVersion은 일반적으로 compileSdkVersion인 경우가 많다.

1. MinsdkVersion

: 앱이 지원하는 최소한의 API 버전을 의미한다. 해당 API 이하의 환경에서 앱 빌드는 정상 작동되지 않는다.

1. Jatpack Libraries

: jatpack 라이브러리는 모든 AOS 환경에서 동일한 작업을 보여주기 위해 호환 가능한 기능을 담은 라이브러리이다. 해당 라이브러리는 androidx package를 통해서 참조되어 제공된다.

1. Vector image

: 벡터 이미지는 API 21 버전 이상부터 지원된다. 하지만 API이하의 버전인 디바이스에서도 동일한 이미지를 볼 수 있기 때문에 큰 변화를 느낄 수 없다. 이는 자체적으로 API 21버전보다 낮은 환경임을 파악하여 벡터 이미지를 PNG이미지로 변환하여 적용하기 때문에 API 21버전 보다 낮은 환경에서도 정상적인 화면을 볼 수 있는 것이다.

하지만 API 21버전에서 다운로드 되는 app의 사이즈보다 더 큰 용량을 차지하게 되는 문제가 발생하게 된다.

이러한 문제를 개선하기 위해서 jatpack 라이브러리가 도움을 줄 수 있다.

vectorDrawables.useSupportLibrary = true

gradle(Module:app)에서 defaultConfig에 해당 내용을 추가하고 벡터 이미지가 사용될 imageView에서 src attribute 대신에 srcCompat attribute를 사용하면 API 21보다 낮은 환경의 API는 API7.2 수준으로 동작하도록 하여 용량이 커지는 문제를 개선한다.

1. xmlns: app namespace

: app 네임스페이스는 xml에서 해당 tag는 안드로이드 프레임워크 core가 아닌, custom code나 libraries들로부터 attributes를 불러 올 수 있는 네임스페이스다.

1. tools namespace

: content의 placeholder 역할을 하는데 사용되는 네임스페이스다. tools namespace에서 사용한 attribute는 app이 컴파일될 때, 제거된다.