

웹앱(하이브리드 앱)은 HTML, CSS, JS 등을 이용해서 앱을 만드는겁니다.  
일단 웹처럼 신나게 코딩을 하고, 그걸 다른 플랫폼에 맞도록 걸부분을 설정(?)하는겁니다.

뭔가 설명하기가 힘드네요.

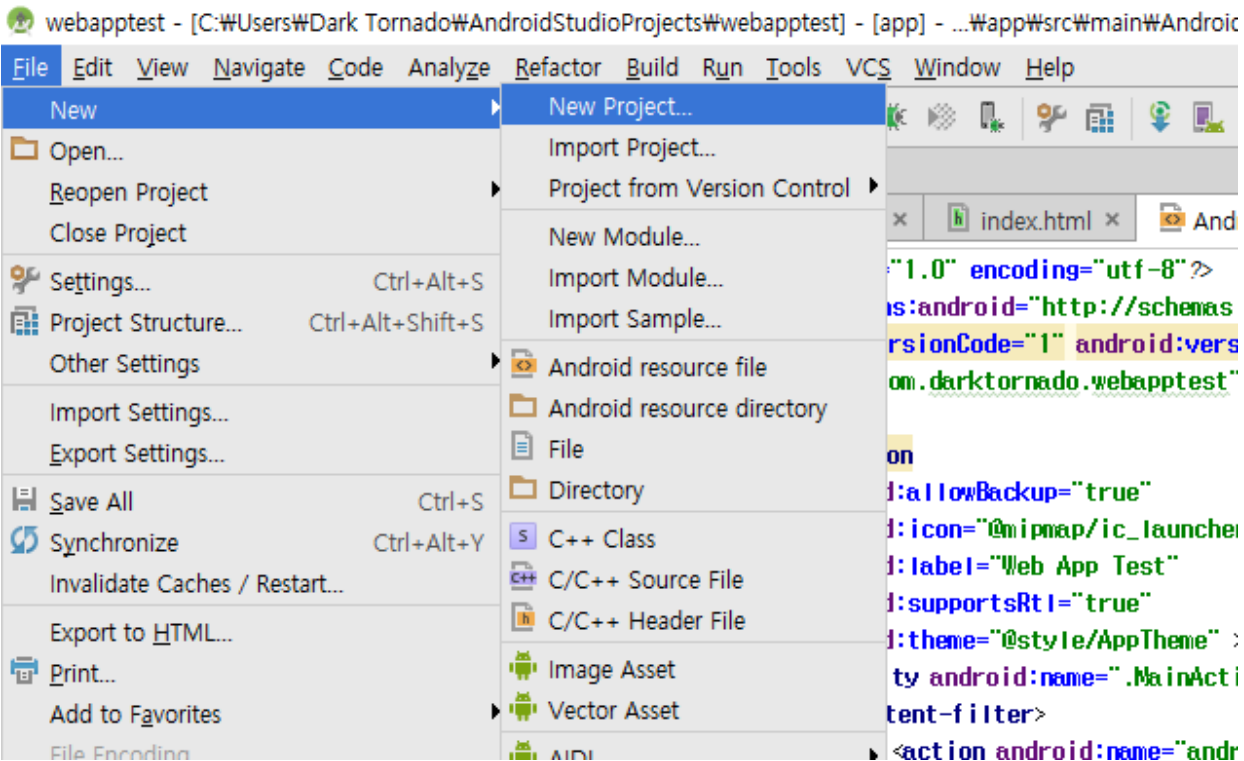
여기에서 설명하는 것은 앱 내부는 HTML 등으로 되어 있고, 걸부분만 .apk로 만드는 방식으로 앱을 만드는겁니다.  
걸부분을 어떻게 하느냐에 따라 안드로이드 용인지 ios용인지 등이 결정됩니다.

JDK, 환경변수, 안스 설치 등등은 알아서 하세요.

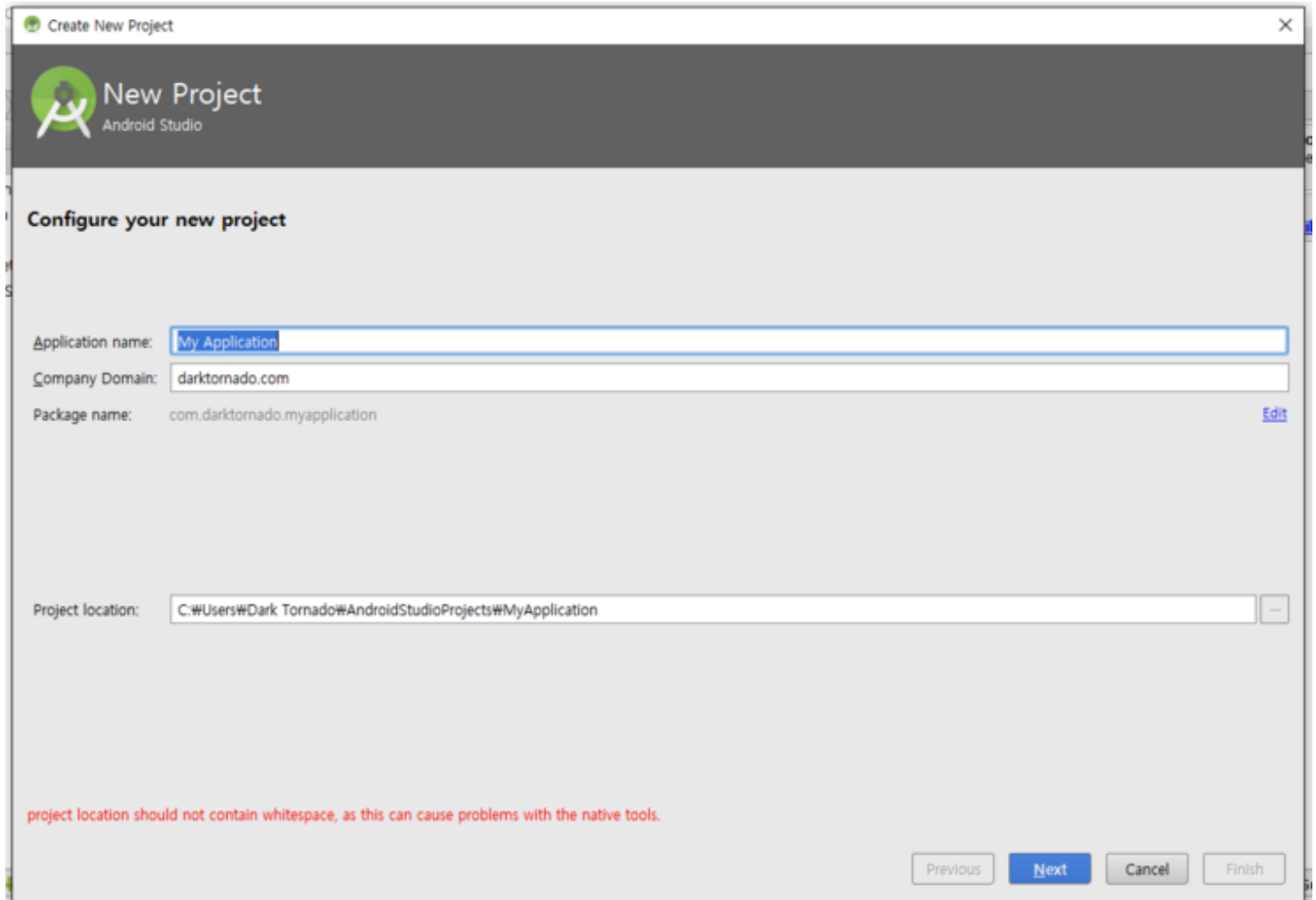
일단, 안스를 켜봅시다.



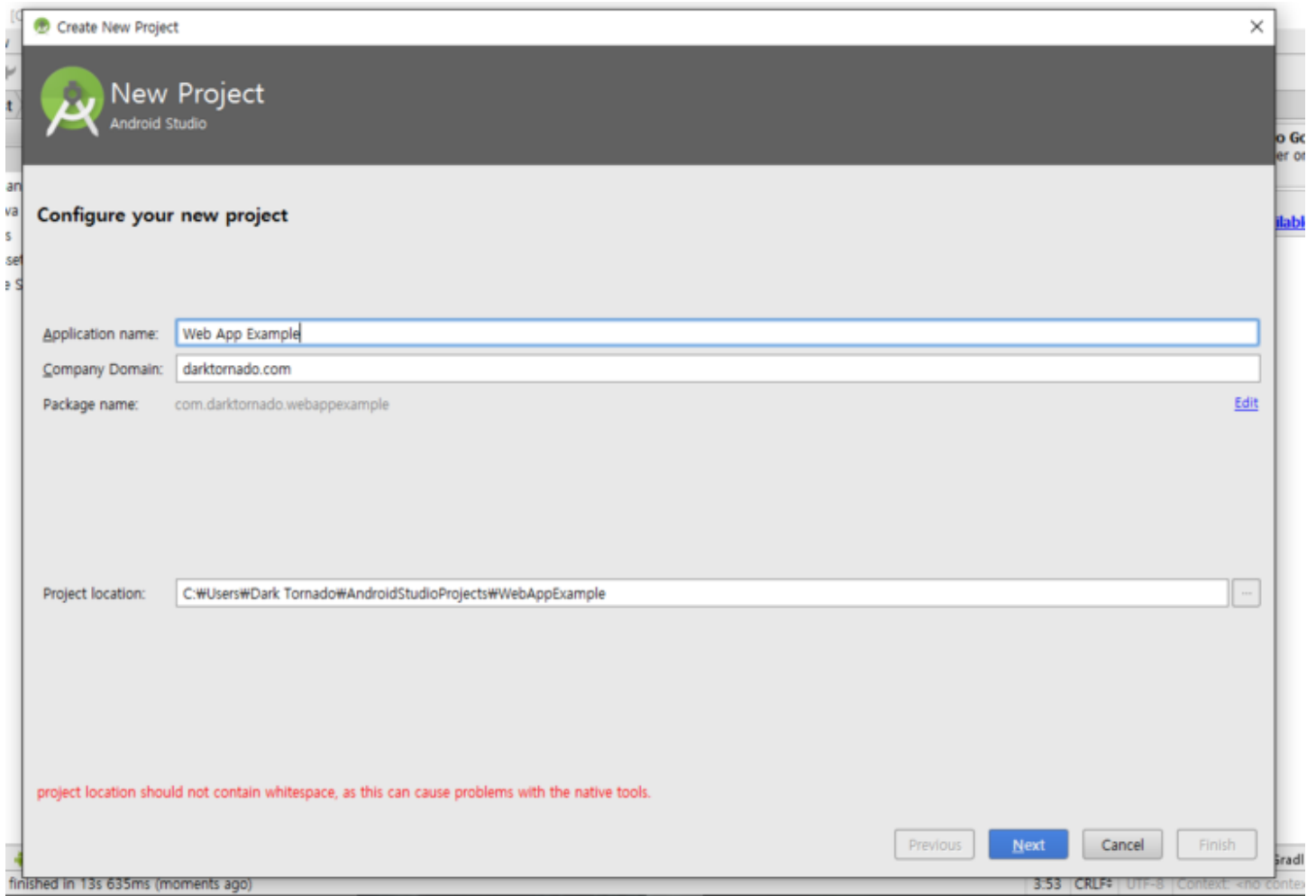
일단 켜졌으면, 새로운 프로젝트를 만듭시다,



블리할라

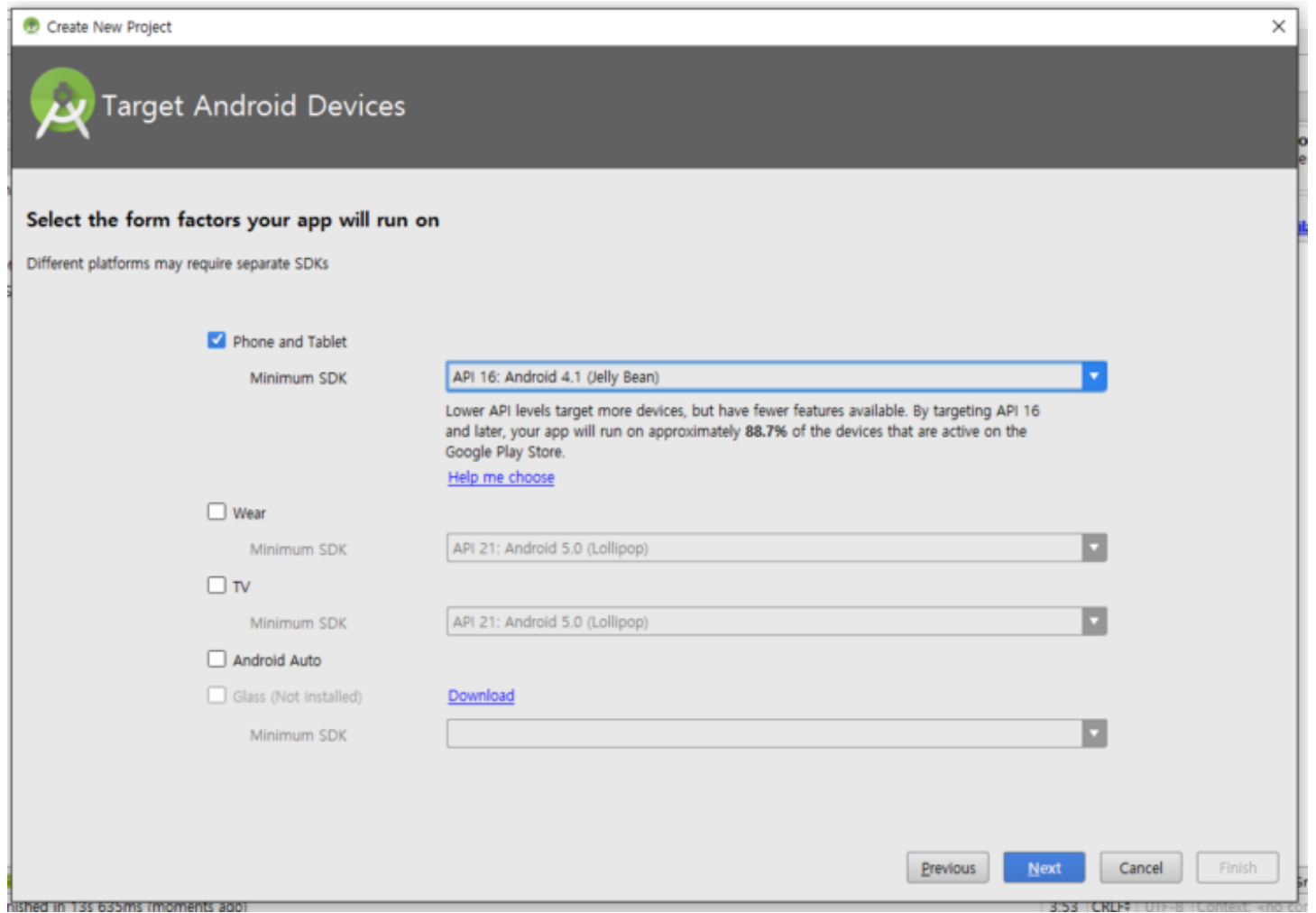


블리할라.

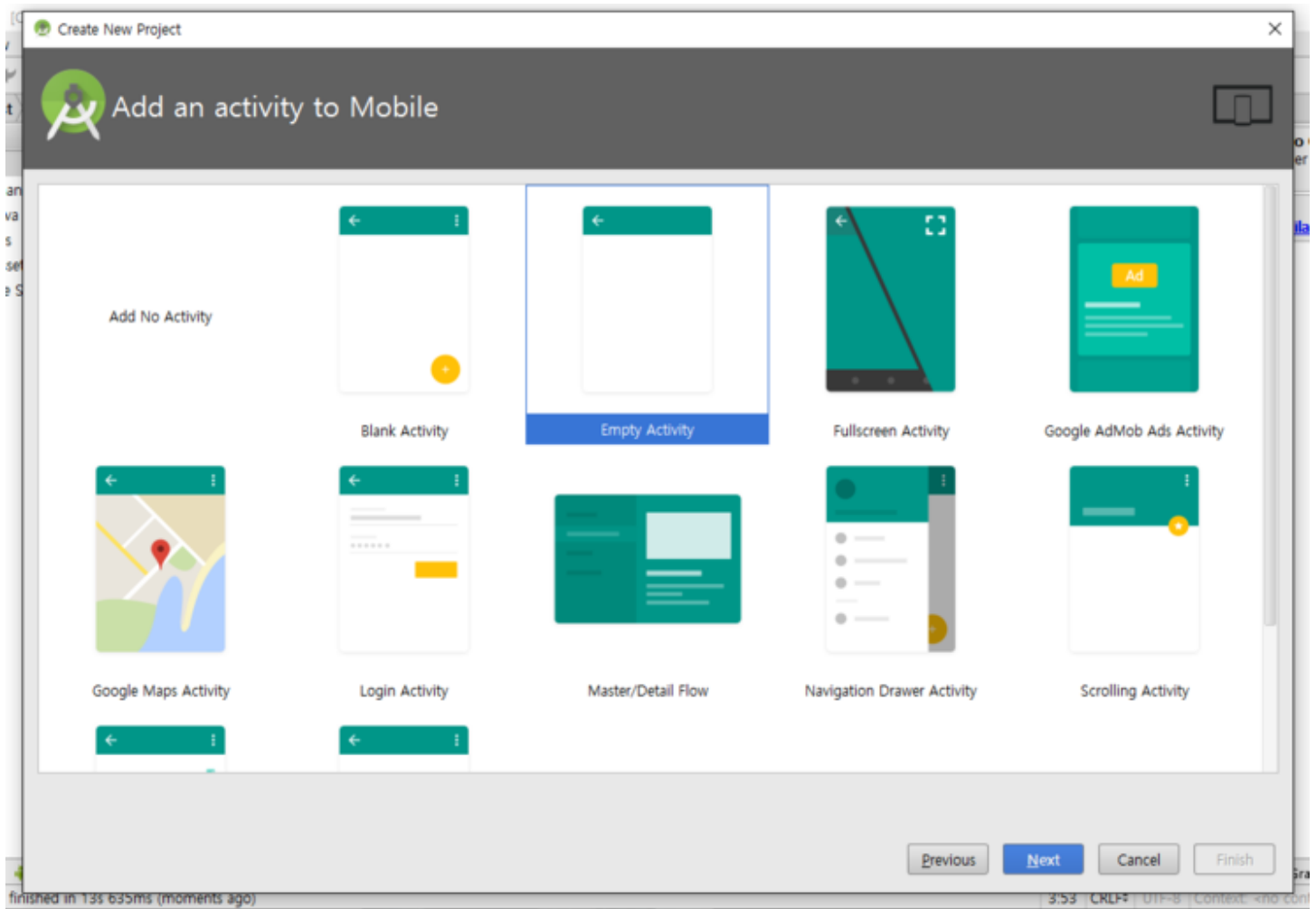


API 16(젤리빈 4.1) 이상에서 설치가 가능하도록 했습니다.

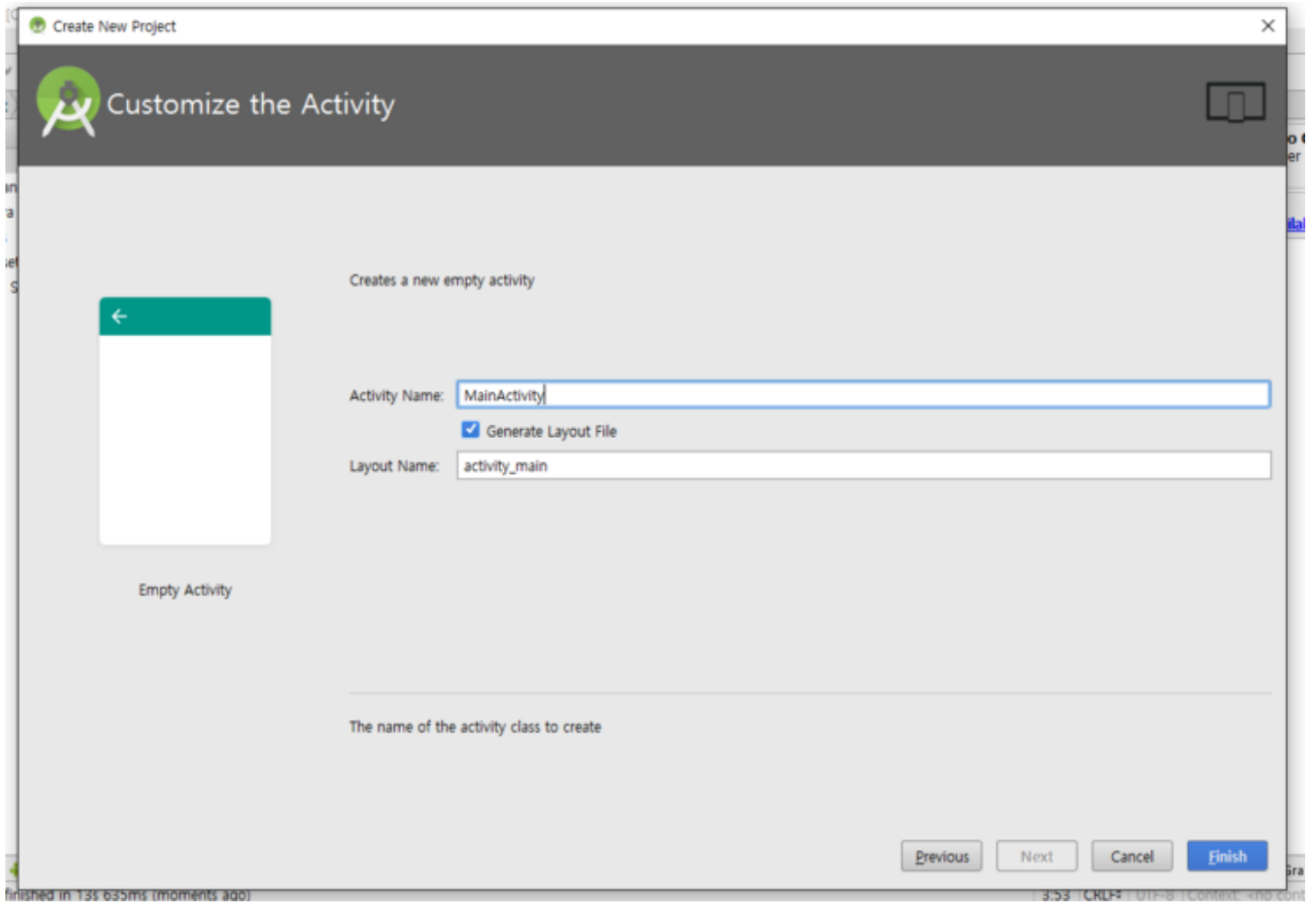
여기서 사용하는 모 메소드가 16 이상에서만 지원한다더군요.



Empty Activity를 선택해줍니다.

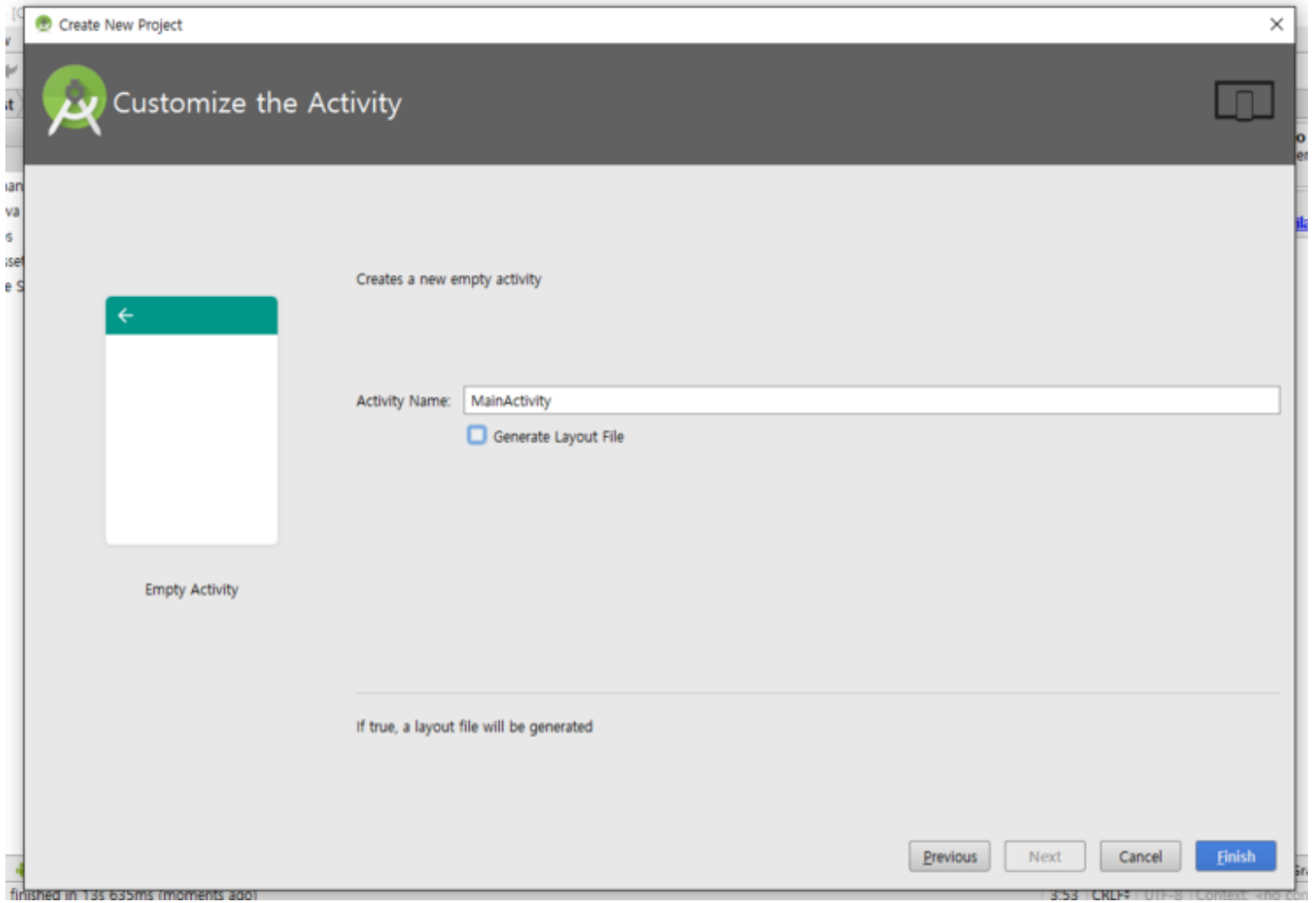


그러면 이렇게 뜨는데, 전 xml 할 줄 모르니,

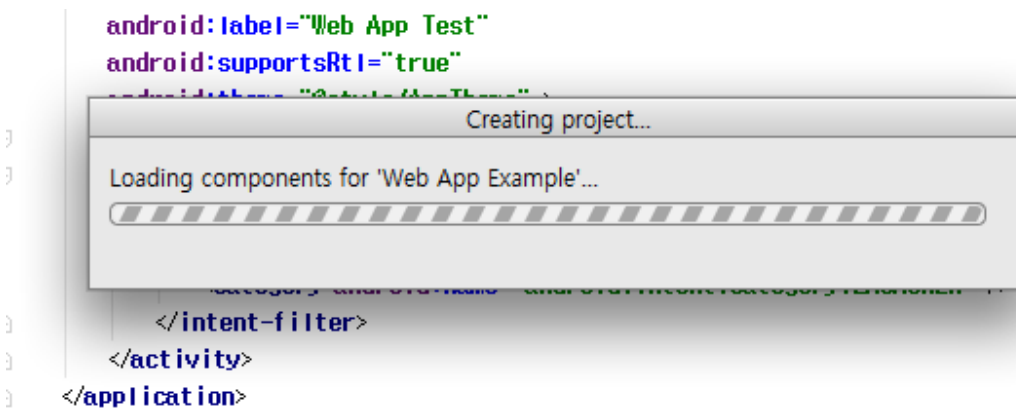


Generate Layout File(xml로 레이아웃 만드는데)을 체크 해제 해줍니다.

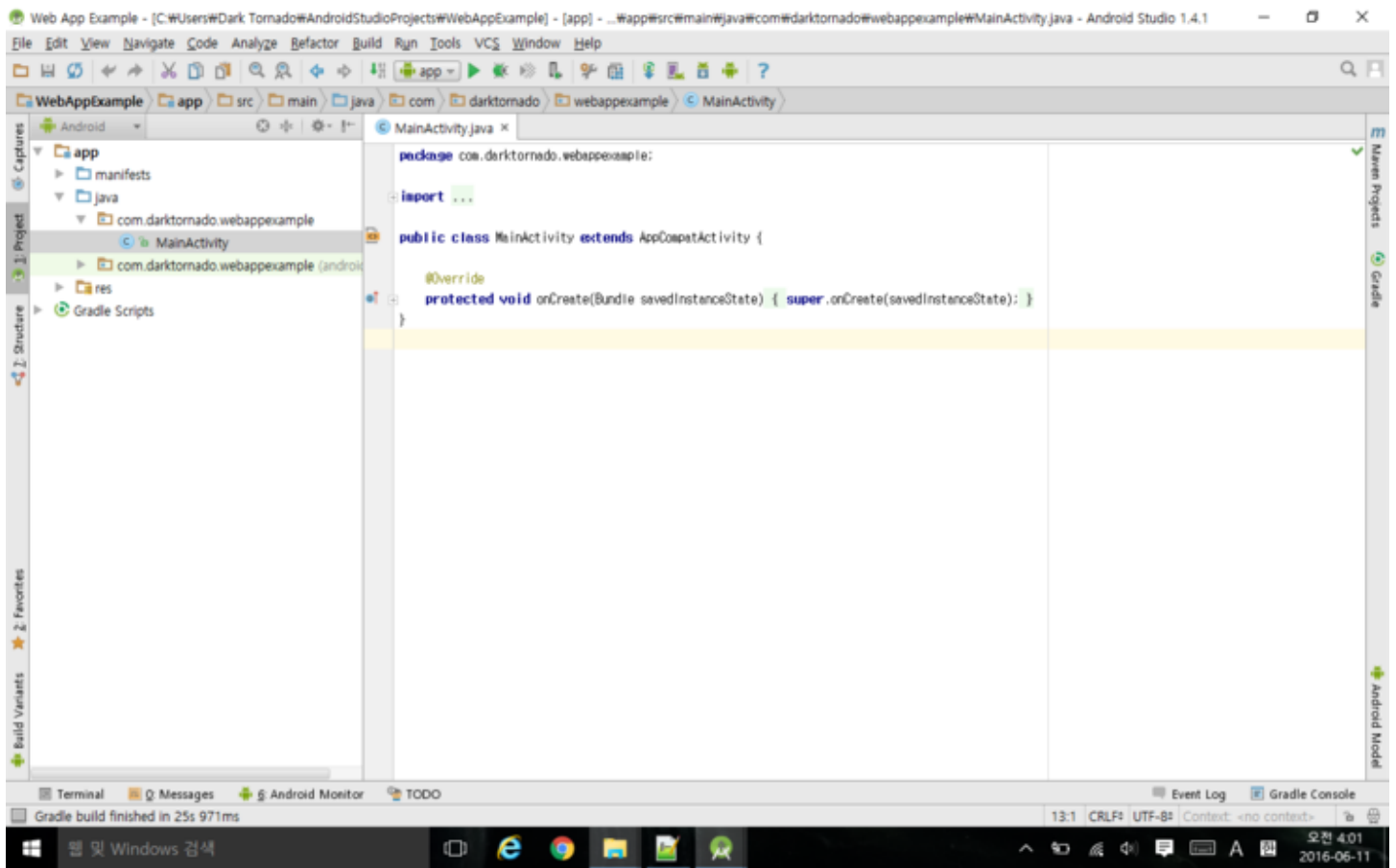
그리고, Finish 버튼을 누르면 프로젝트가 생성됩니다.



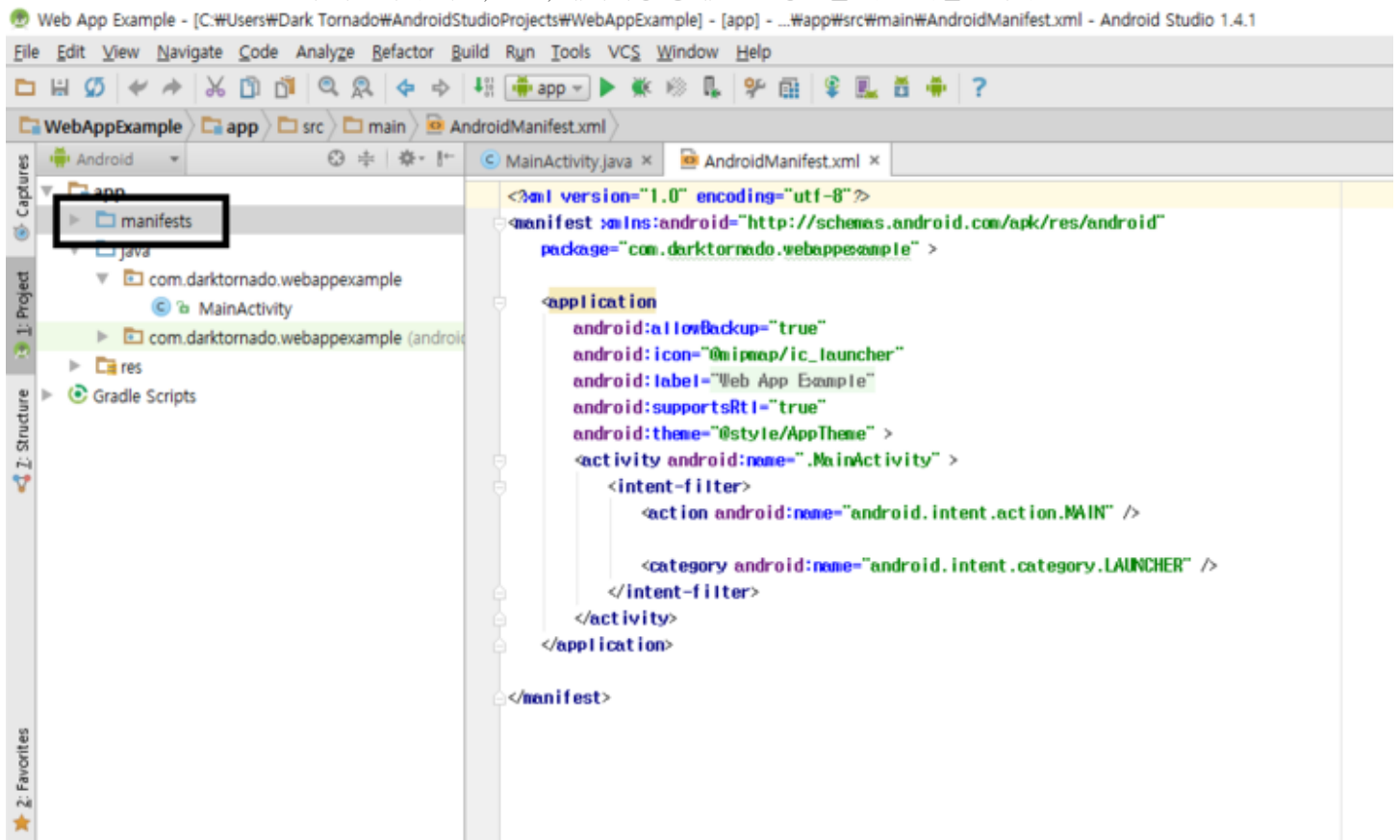
날리랄라



생성 및 로드가 완전히 끝나면 이런 창이 뜹니다.

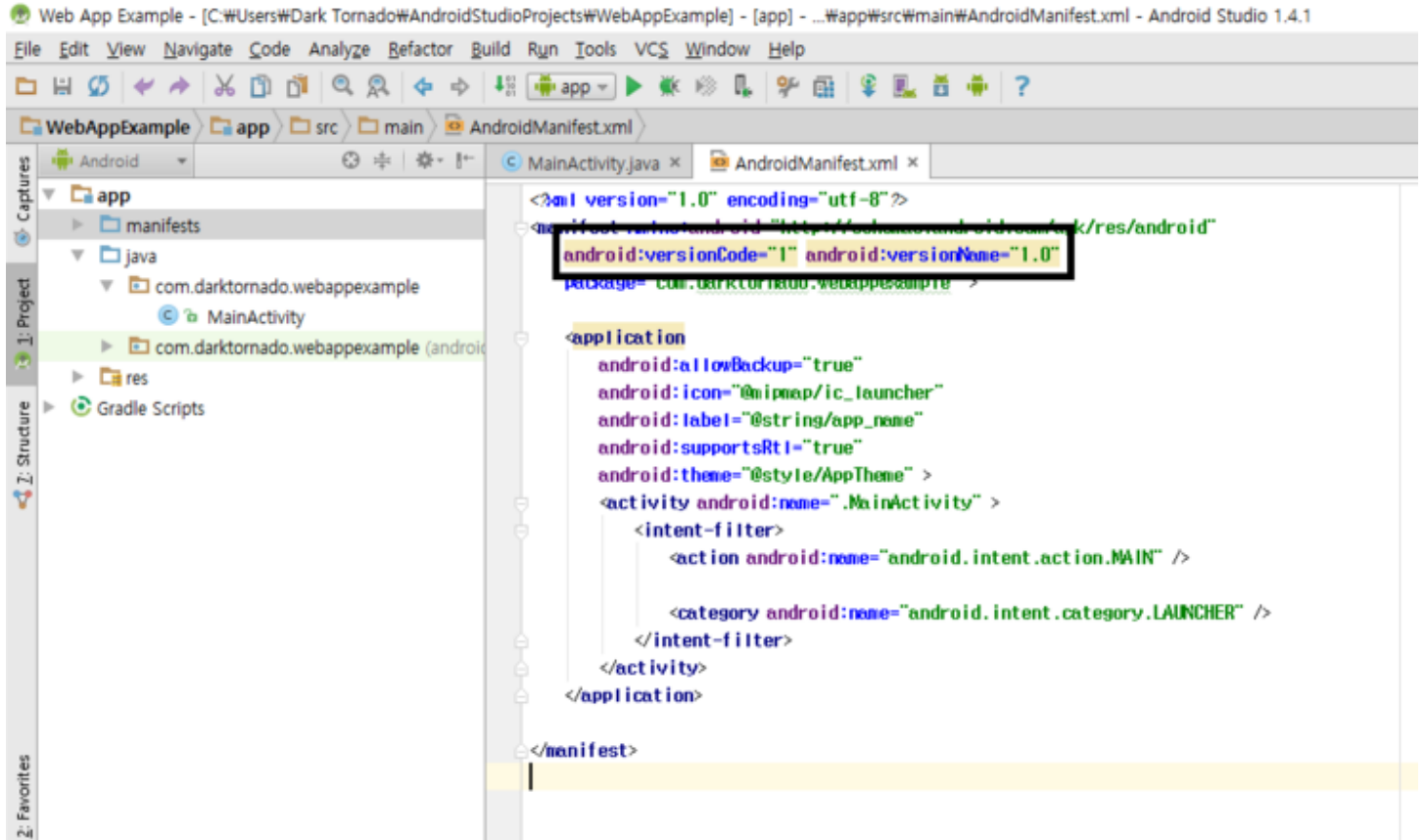


AndroidManifest.xml 파일의 싸대기를 때려봅시다. 네모친 부분을 더블 클릭하거나 하면 열립니다.  
AndroidManifest.xml은 앱의 이름이나 버전, 권한, 패키지명 등에 관한 정보를 가진 파일입니다.

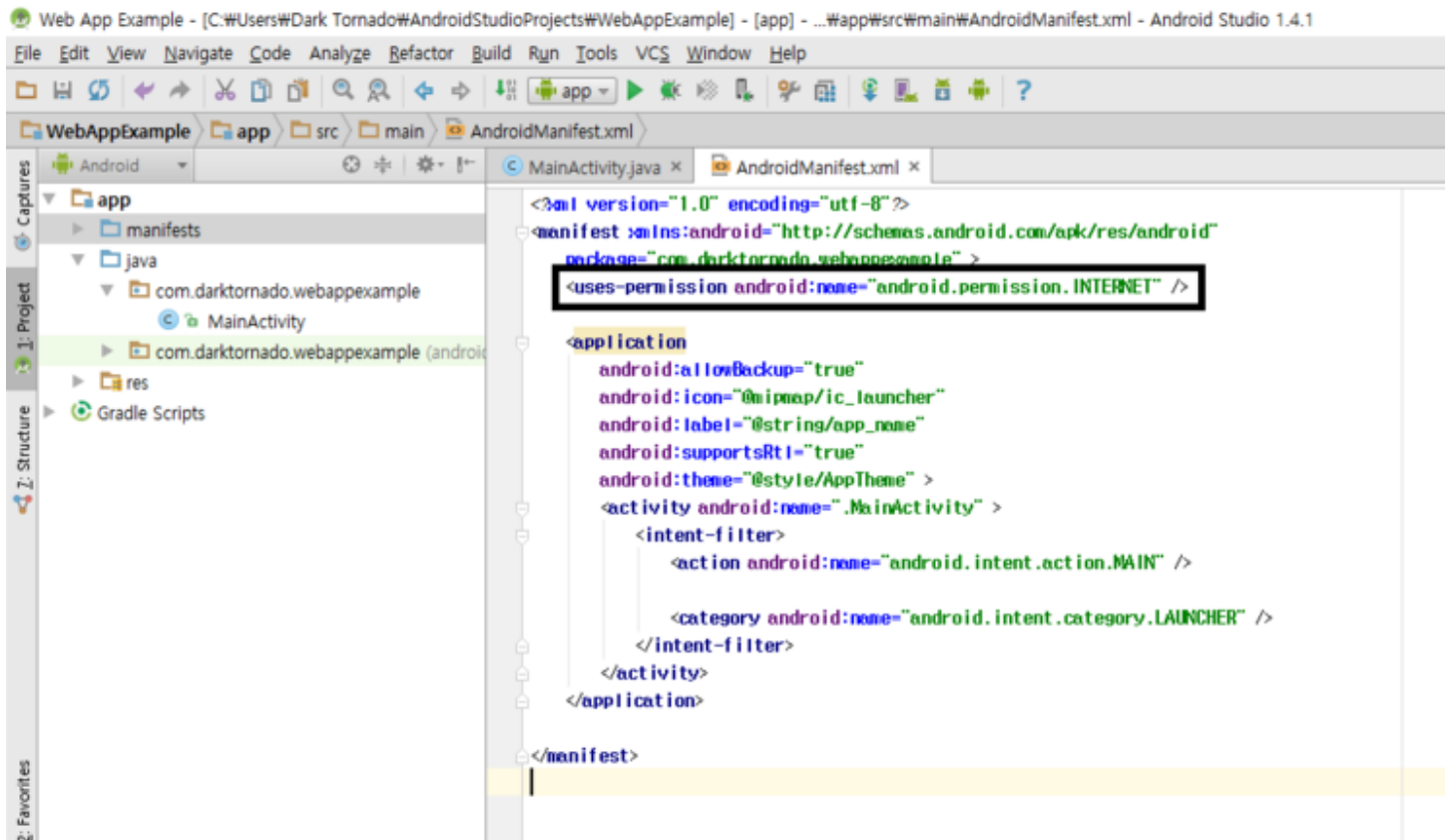




버전 코드와 버전명을 적어줍니다 귀찮으면 적지 마세요(?).

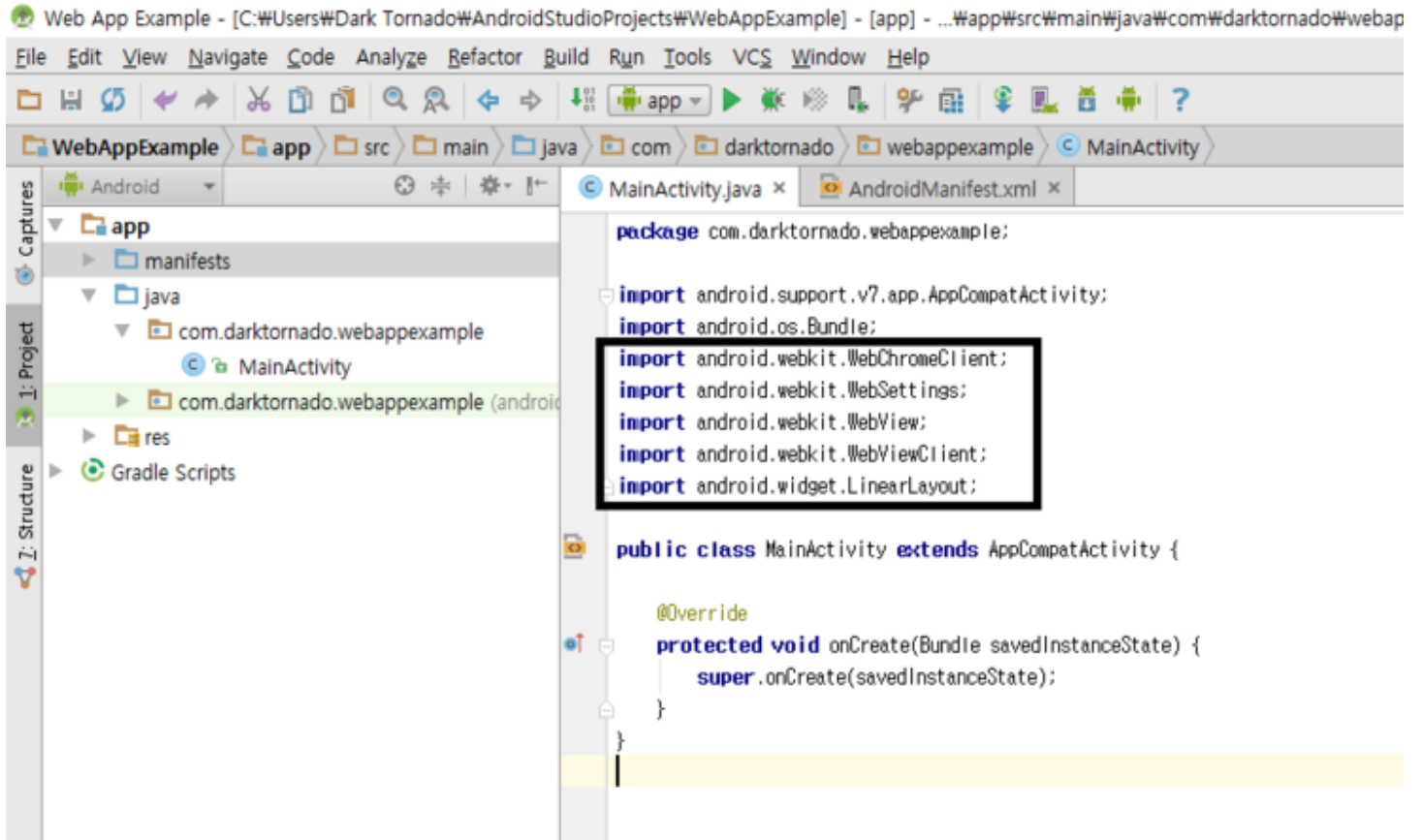


만약, 본인의 홈페이지나 블로그 등도 띄울 생각이면, 인터넷 권한을 추가하셔야 합니다.

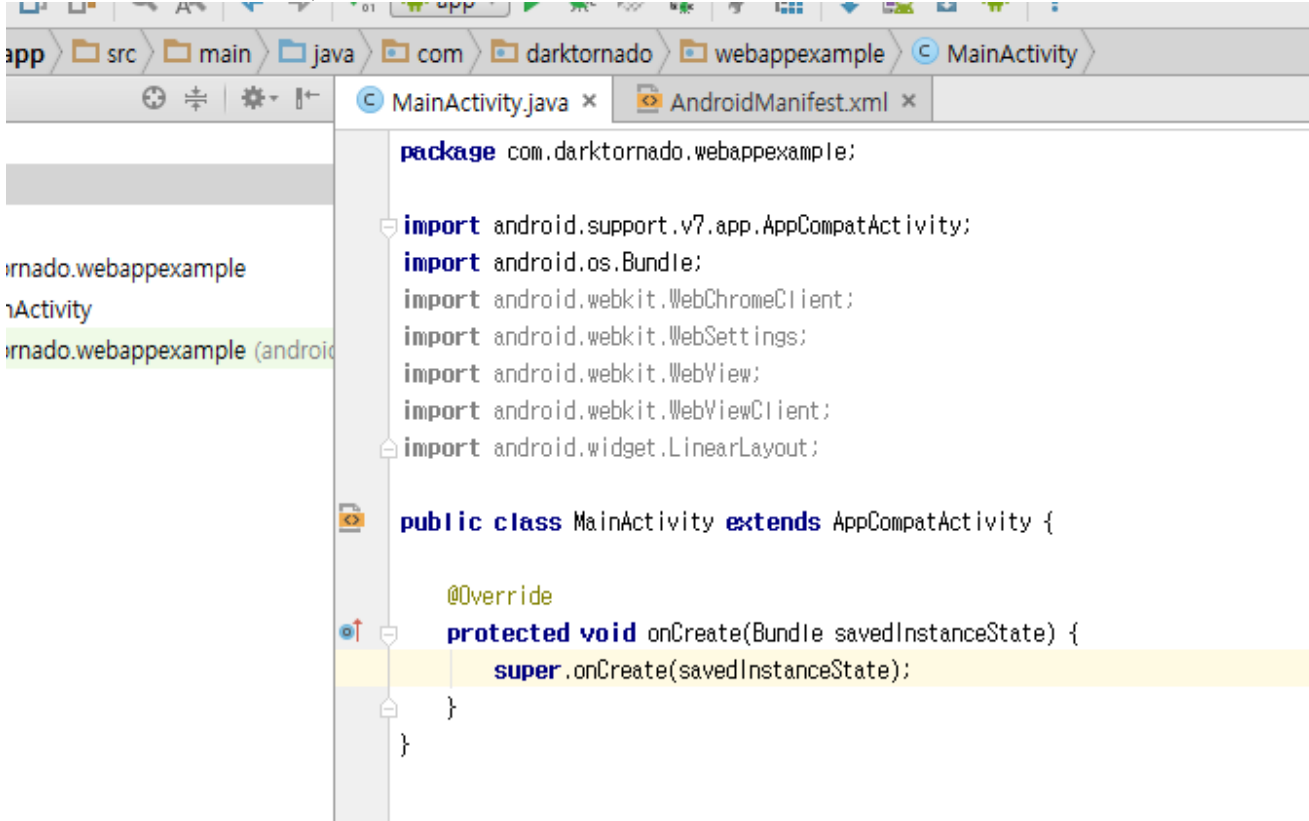


일단 이 정도만 만지고, MainActivity.java 파일을 열어봅시다.  
일단 import 부분에 저런 내용들을 적어줍니다.  
스크립트 만드셨던 분들은 뭔가 익숙한 것들이 많이 보일겁니다.

특히 android.widget.LinearLayout

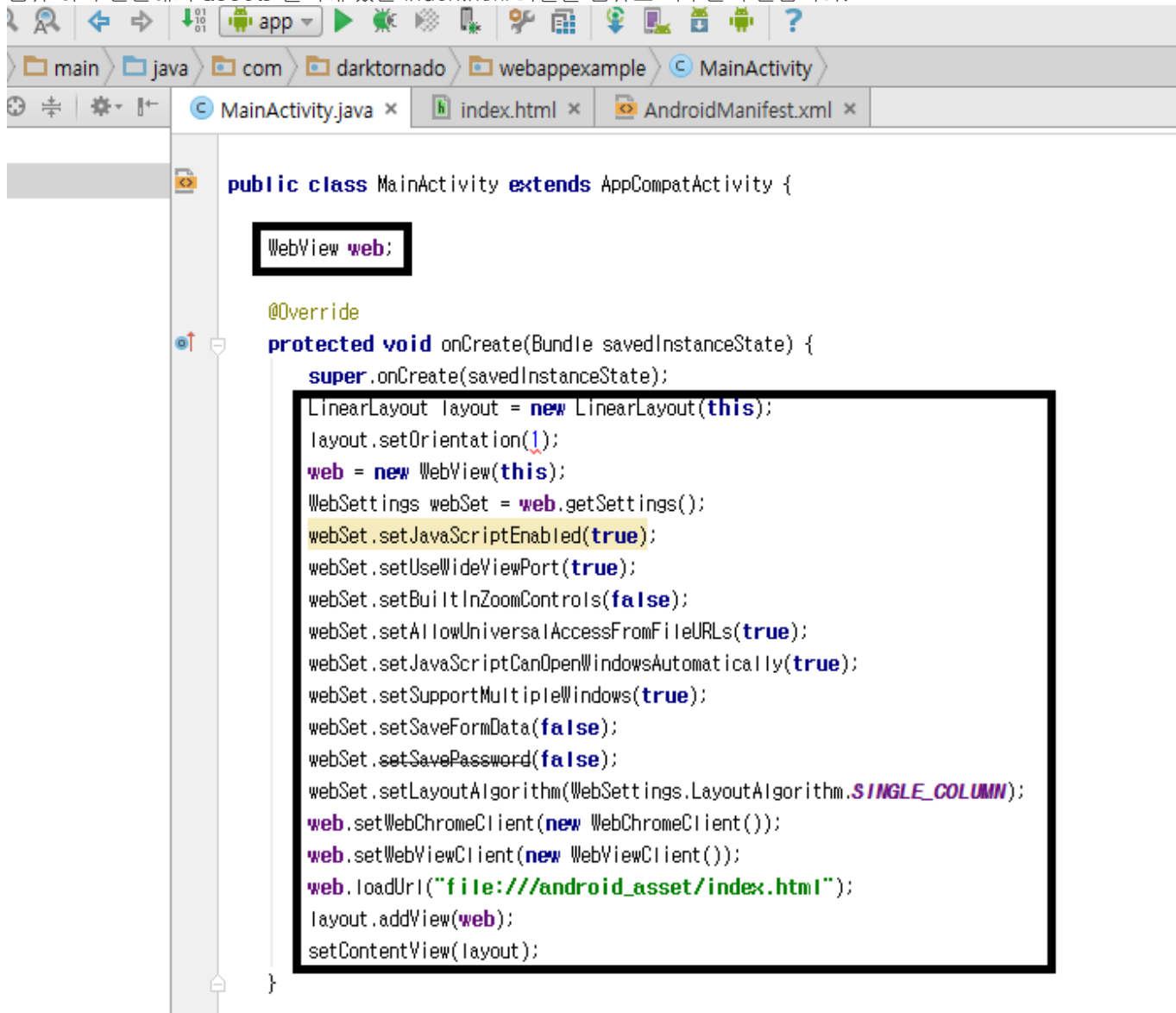


이렇게 흐려지는 이유는 import는 했는데, 아래에서는 안썼다고 나름 빼애액 거리는 겁니다.



이렇게도 적어줍니다.

웹뷰 하나 선언해서 assets 폴더에 있는 index.html 파일을 웹뷰로 띄우는 부분입니다.



The screenshot shows an IDE window with the following code in MainActivity.java:

```
public class MainActivity extends AppCompatActivity {  
    WebView web;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        LinearLayout layout = new LinearLayout(this);  
        layout.setOrientation(1);  
        web = new WebView(this);  
        WebSettings webSet = web.getSettings();  
        webSet.setJavaScriptEnabled(true);  
        webSet.setUseWideViewPort(true);  
        webSet.setBuiltInZoomControls(false);  
        webSet.setAllowUniversalAccessFromFileURLs(true);  
        webSet.setJavaScriptCanOpenWindowsAutomatically(true);  
        webSet.setSupportMultipleWindows(true);  
        webSet.setSaveFormData(false);  
        webSet.setSavePassword(false);  
        webSet.setLayoutAlgorithm(WebSettings.LayoutAlgorithm.SINGLE_COLUMN);  
        web.setWebChromeClient(new WebChromeClient());  
        web.setWebViewClient(new WebViewClient());  
        web.loadUrl("file:///android_asset/index.html");  
        layout.addView(web);  
        setContentView(layout);  
    }  
}
```

Annotations in the code include a red box around `WebView web;`, a red arrow pointing to `web` in `web = new WebView(this);`, and a large black box around the entire `onCreate` method body.

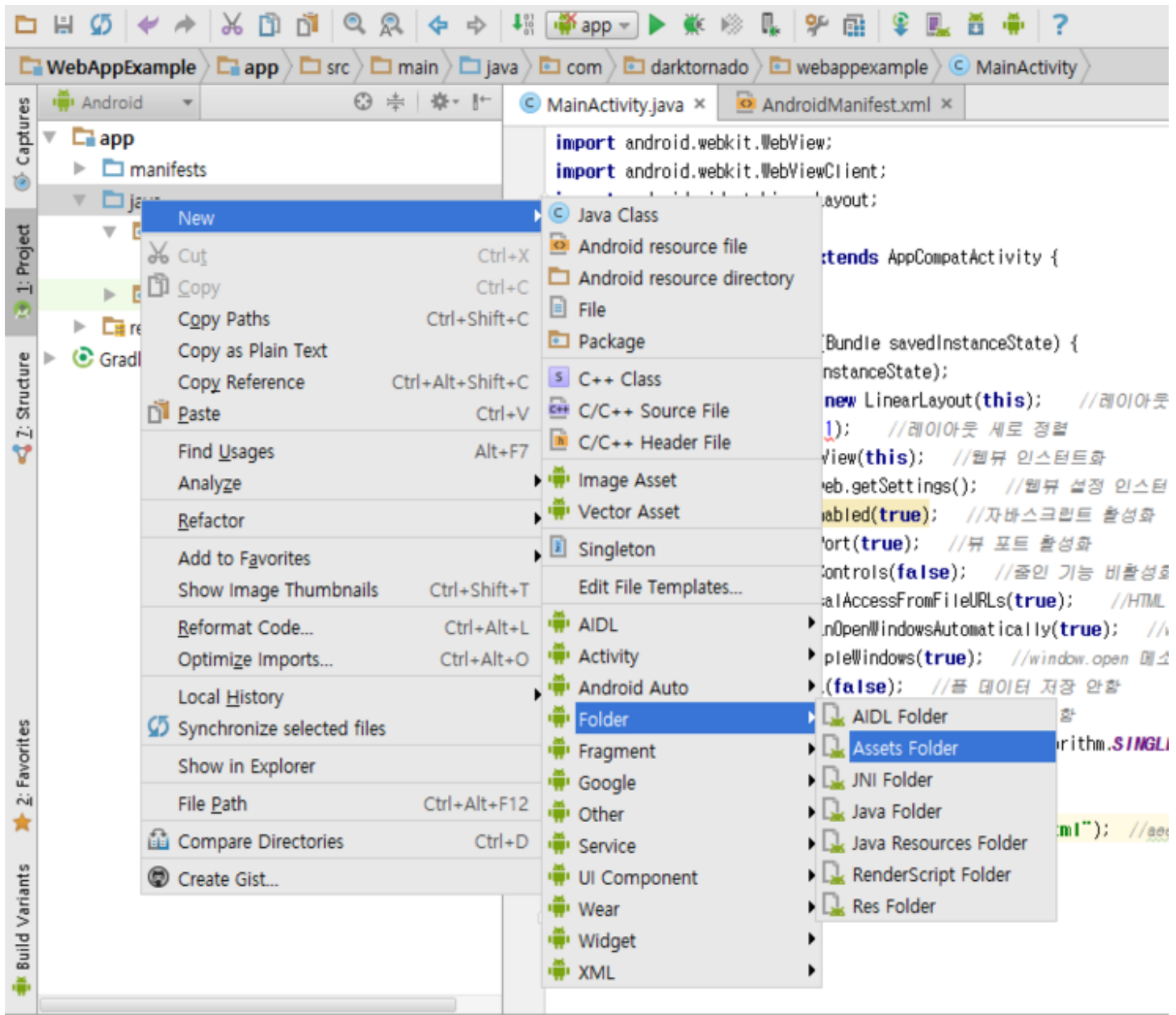
이것도 적어줍니다.  
휴대폰의 이전 키를 눌렀을 때, 웹뷰에서 뒤로 갈 수 있으면 뒤로 가고,

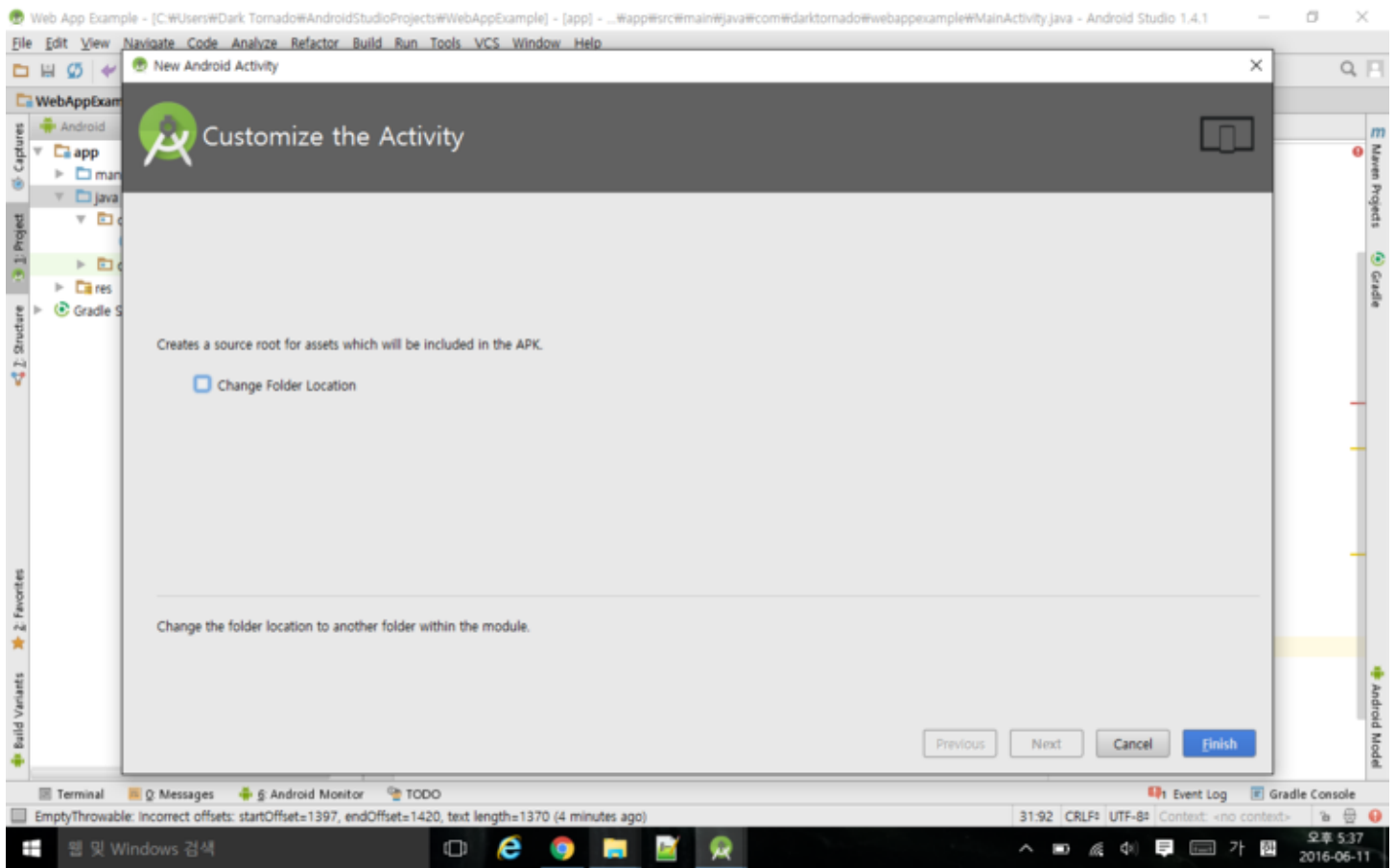
뒤로 가지 못하면 앱을 끄는 부분입니다.

```
webSet.setJavaScriptCanOpenWindowsAutomatically(true); //window
webSet.setSupportMultipleWindows(true); //window.open 메소드 사
webSet.setSaveFormData(false); //폼 데이터 저장 안함
webSet.setSavePassword(false); //비번 저장 안함
webSet.setLayoutAlgorithm(WebSettings.LayoutAlgorithm.SINGLE_COLUMN);
web.setWebChromeClient(new WebChromeClient());
web.setWebViewClient(new WebViewClient());
web.loadUrl("file:///android_asset/index.html"); //assets 폴더
layout.addView(web);
setContentView(layout);
}

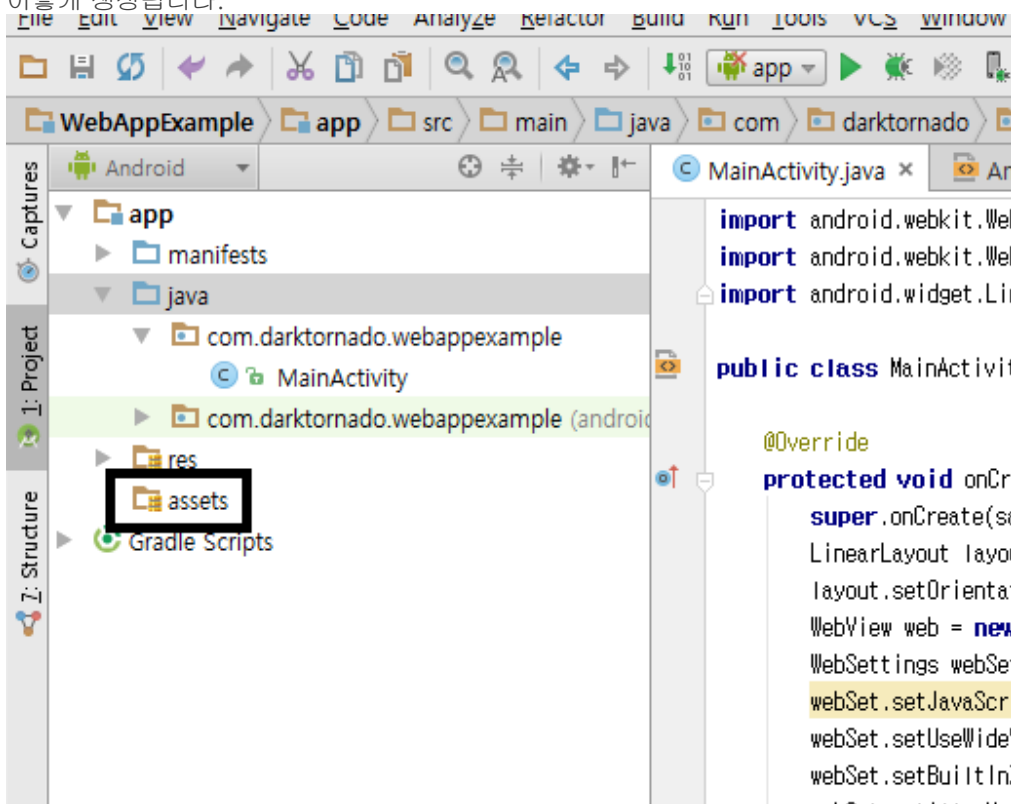
public void onBackPressed() {
    if (web.canGoBack()) web.goBack();
    else finish();
}
}
```

이제, assets 폴더를 만들어봅시다.

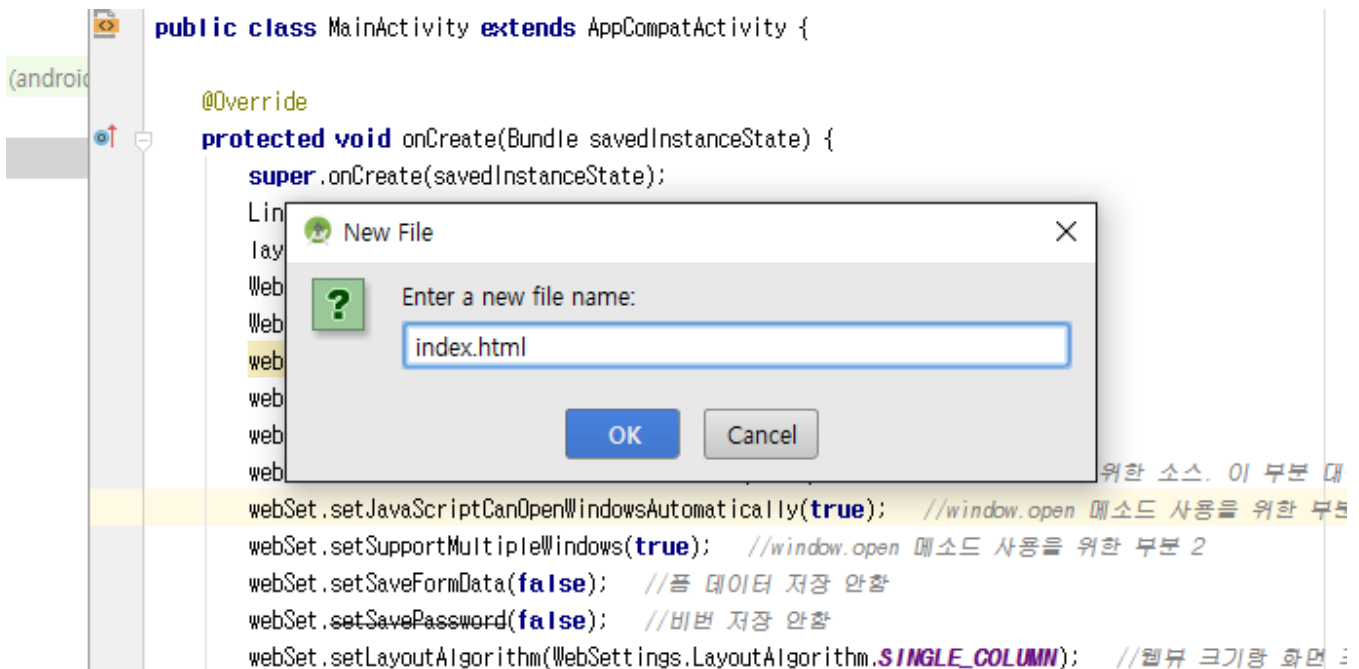
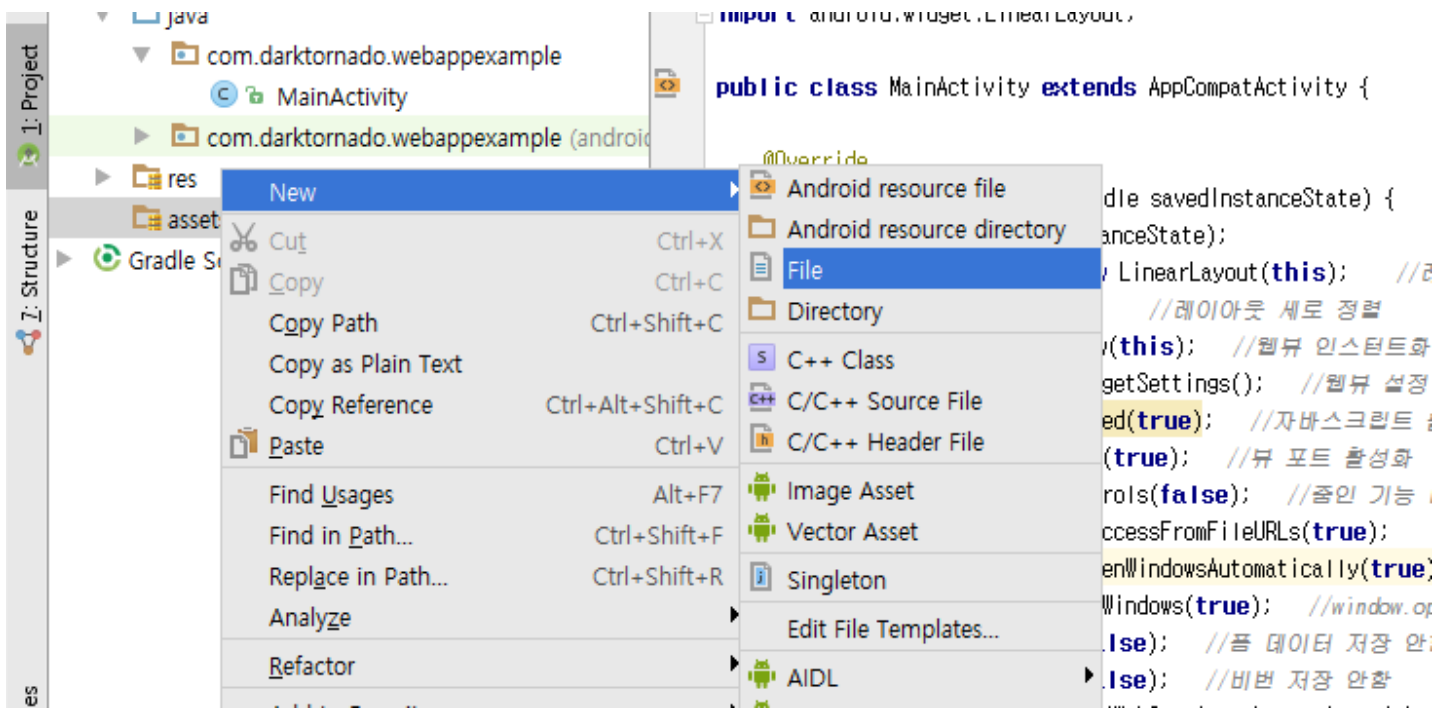




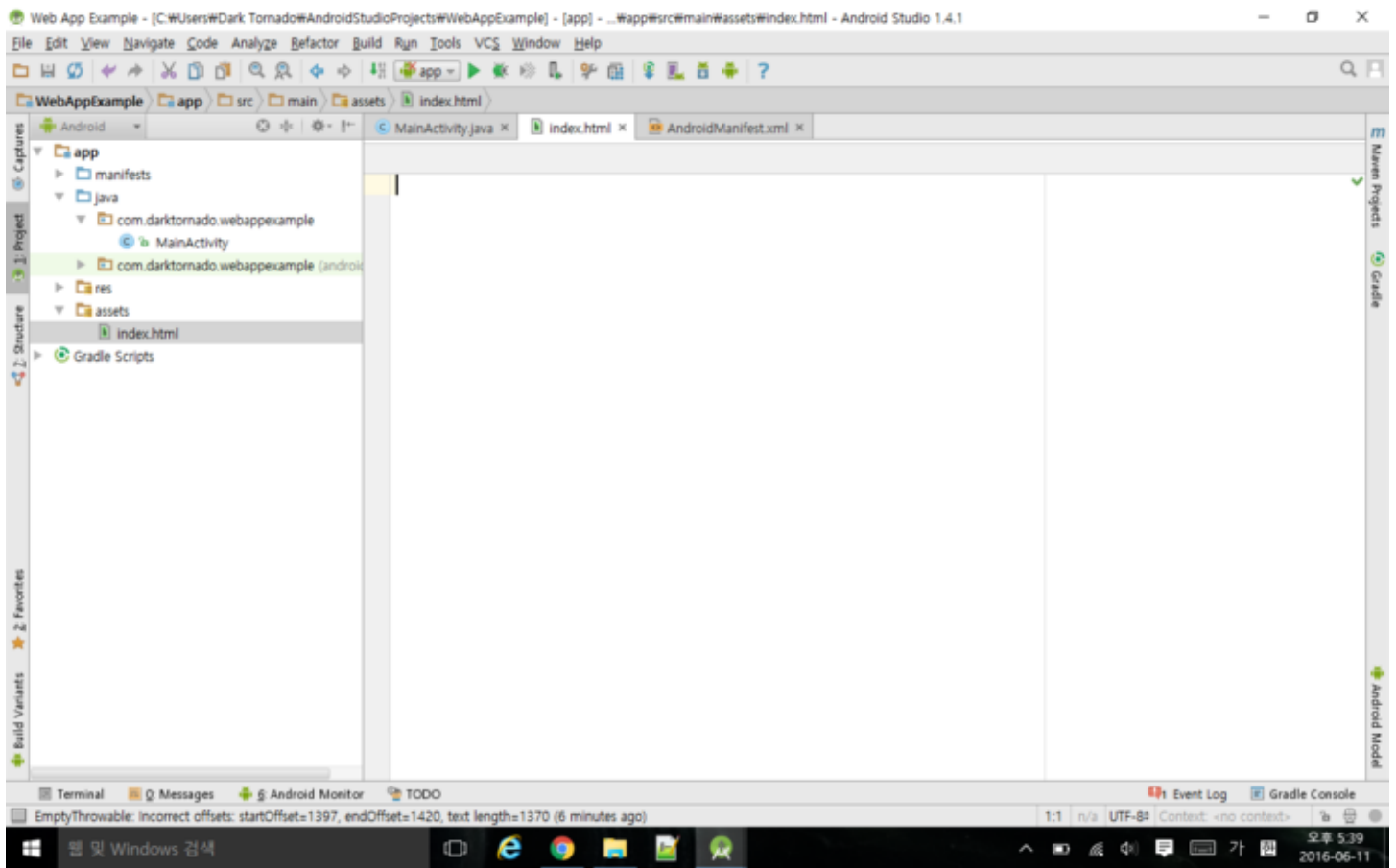
이렇게 생성됩니다.



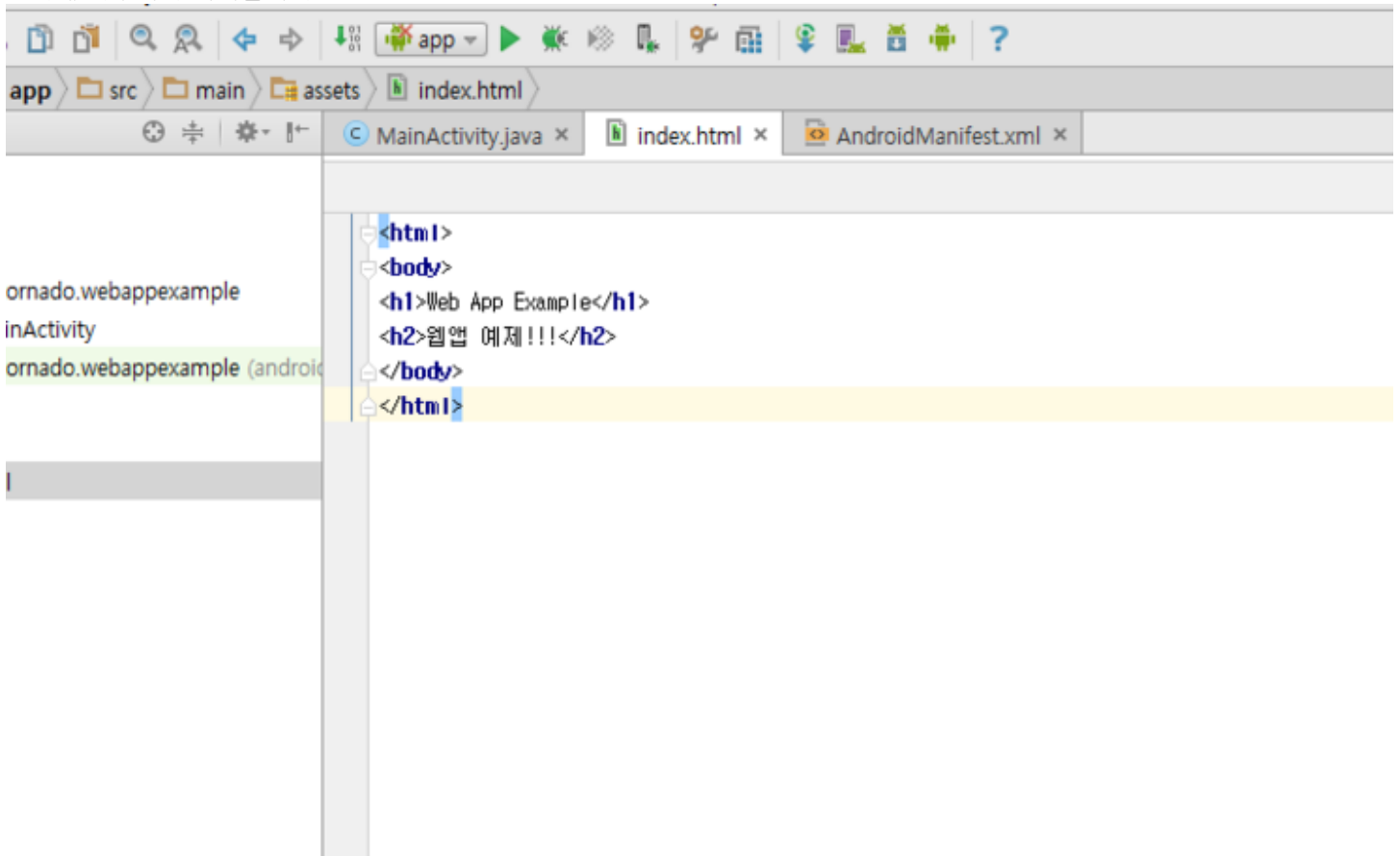
이제, index.html 파일을 생성해봅시다.



이렇게 생성됩니다.

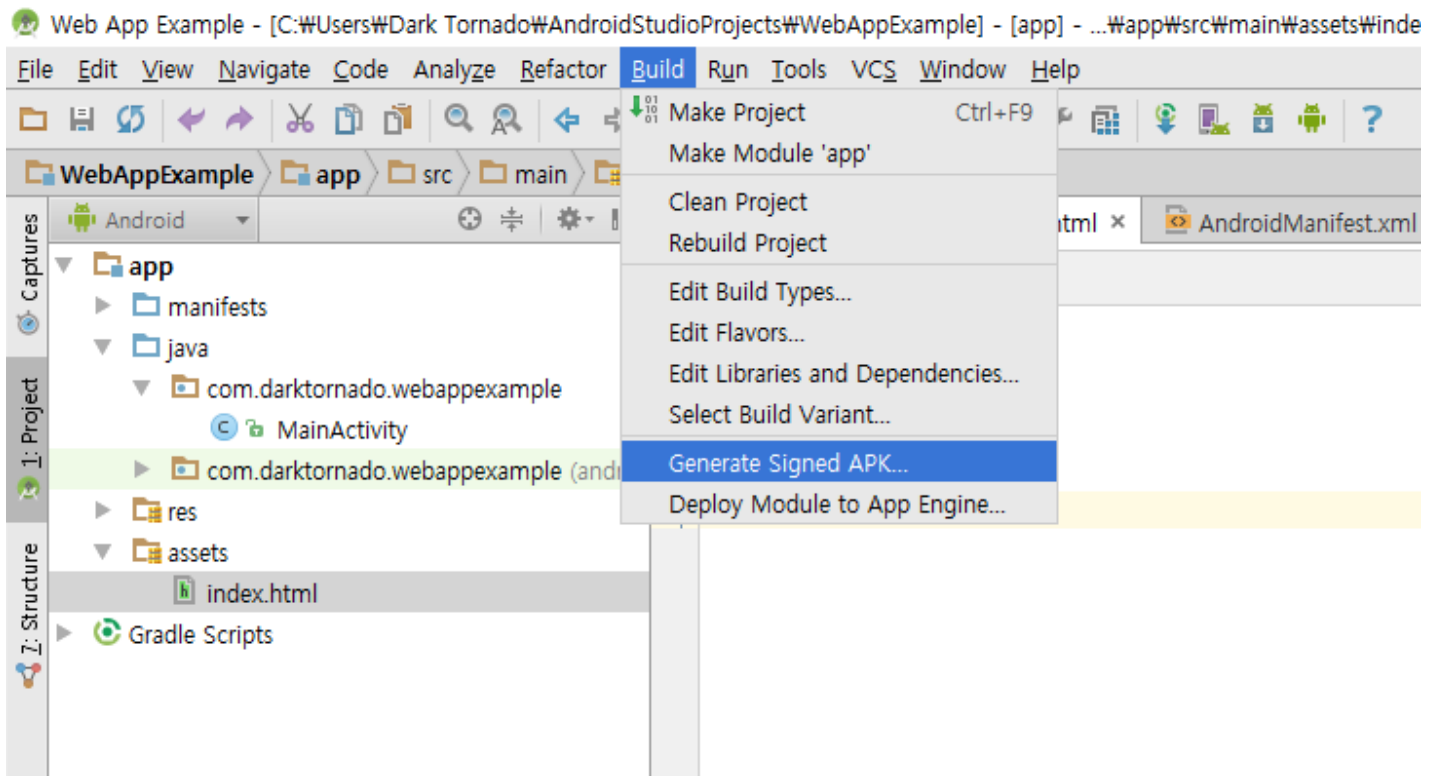


신나게 아무거나 적어봅시다.

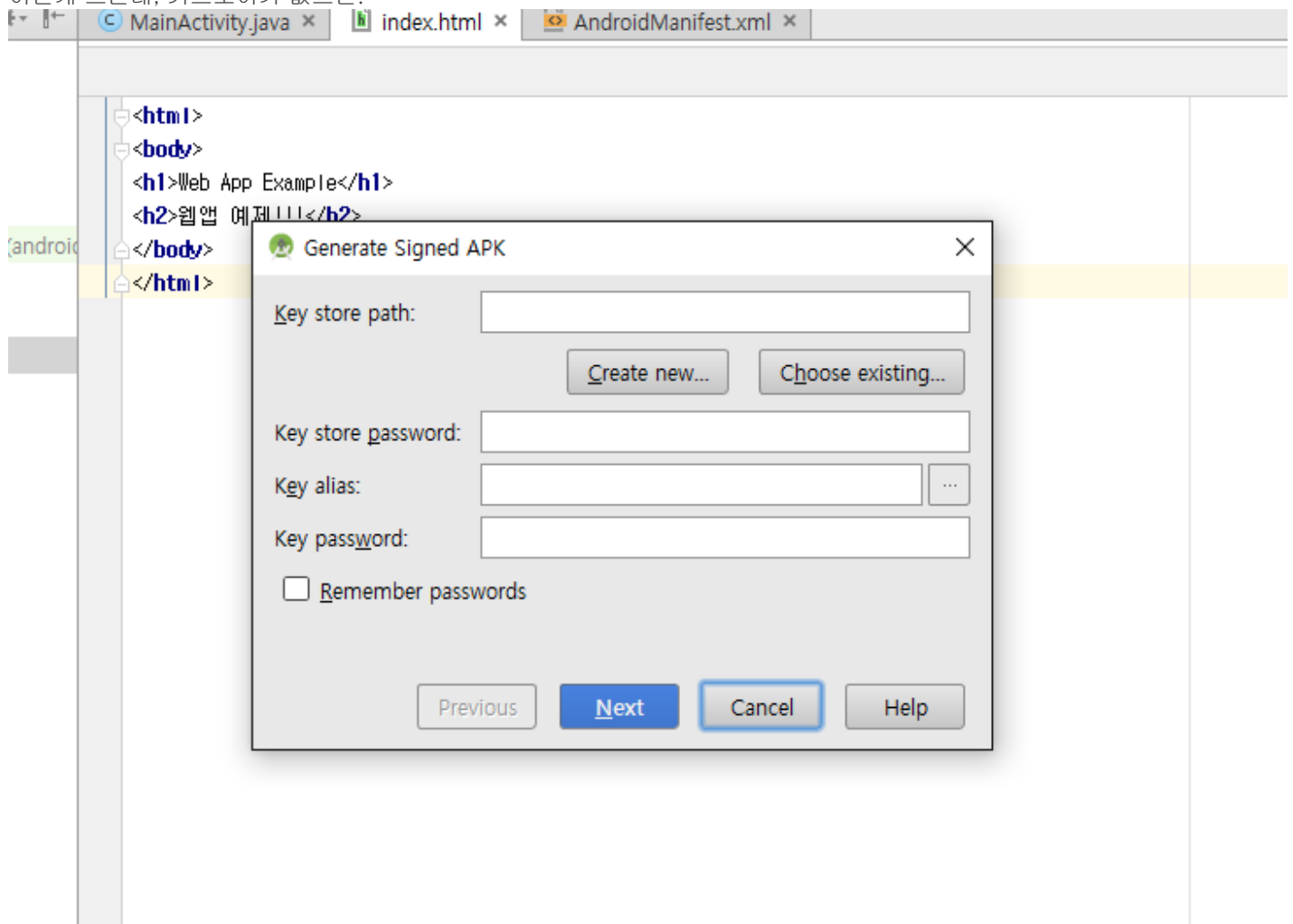


.apk 파일로 뽑아봅시다.



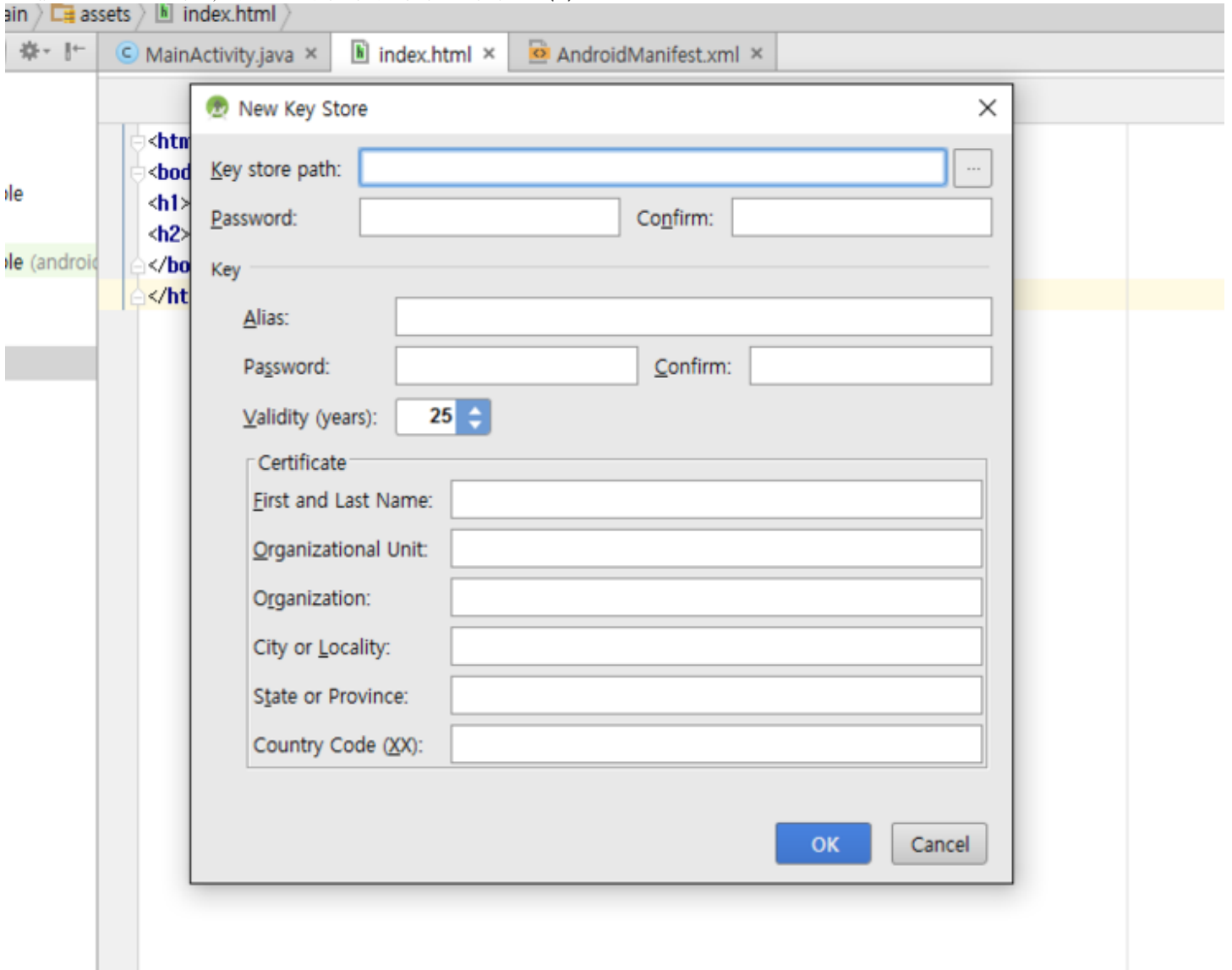


이런게 뜨는데, 키스토어가 없으면.

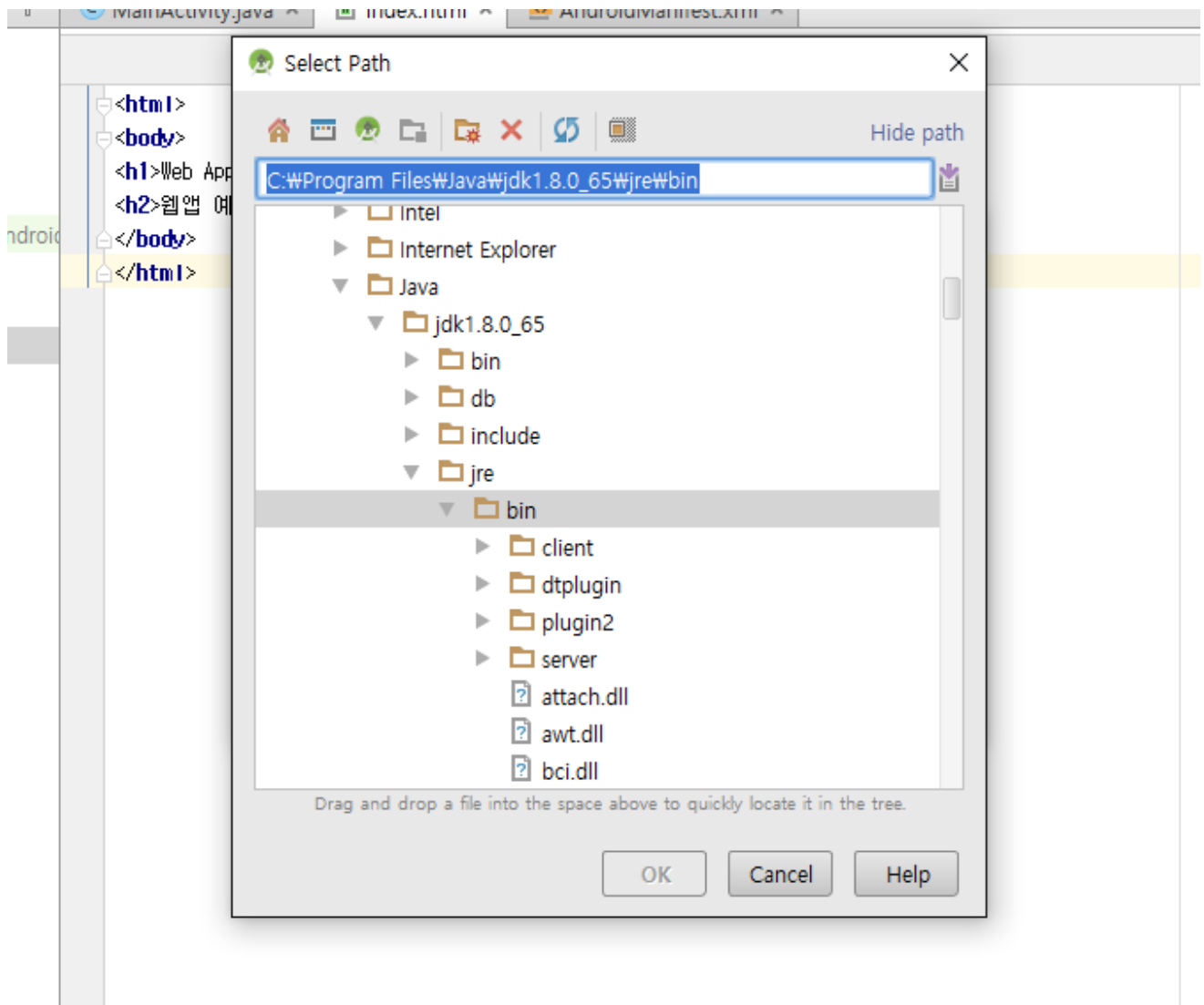


Create new 눌러서 만드시면 됩니다.

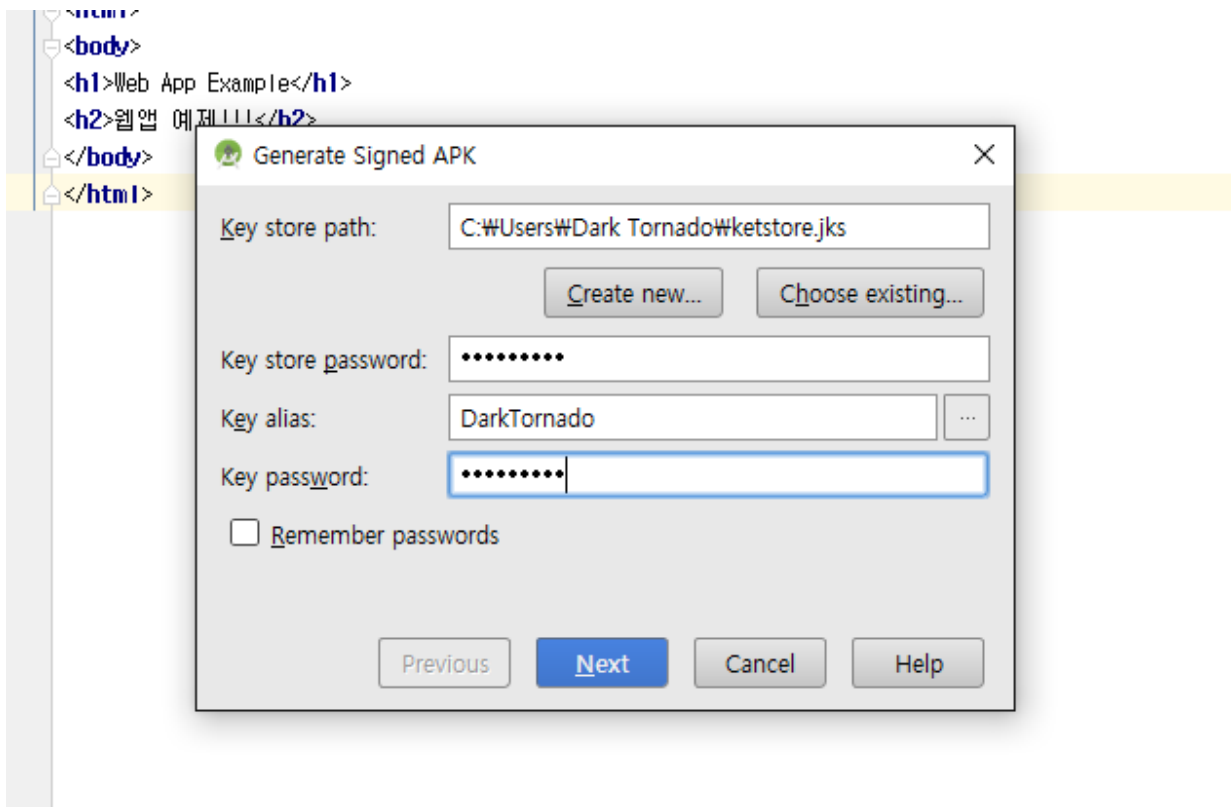
맘에 안드는건 안적고, 다른건 아무거나 적어도 되더군요(?).



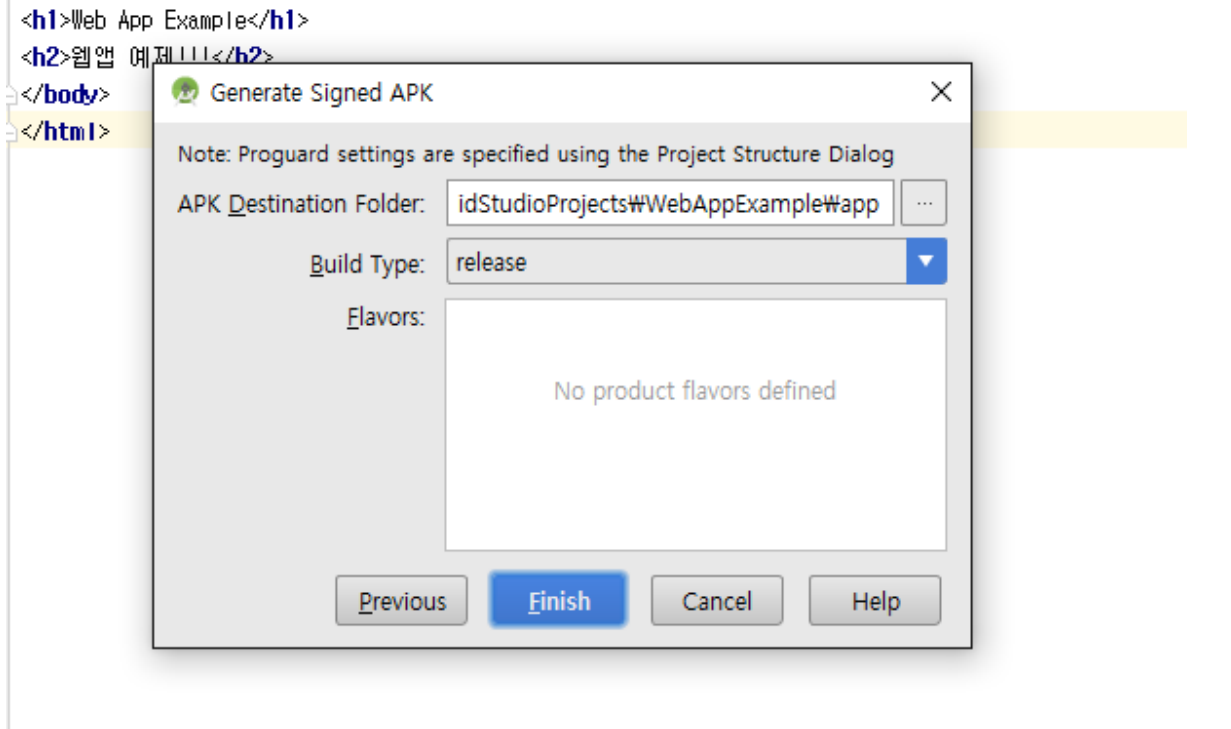
저는 Choose existing 버튼으로 기존에 있던걸 불러왔습니다.



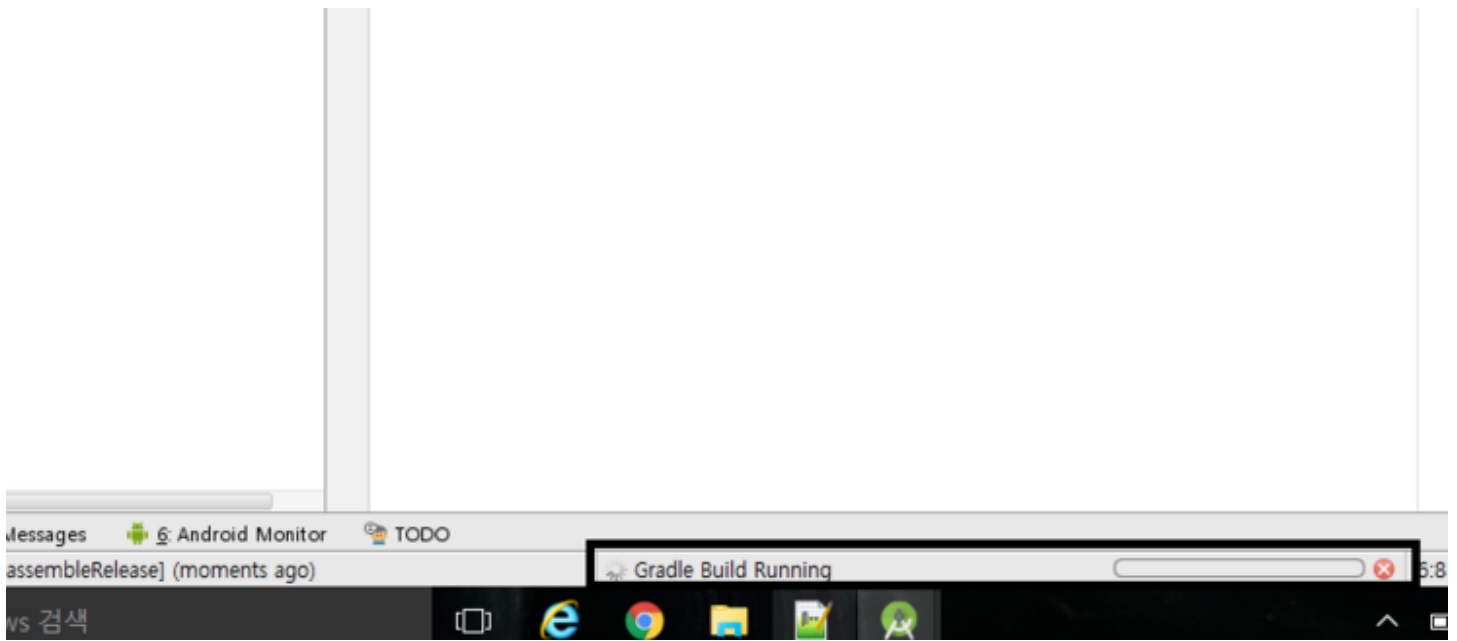
똑같이 따라 적지 마세요?!



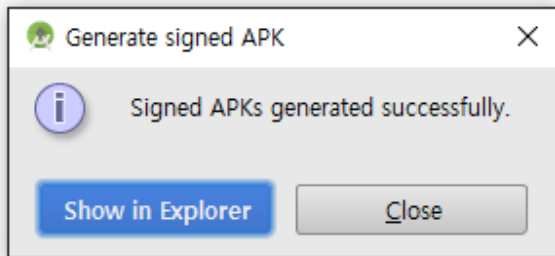
Next 버튼을 누르면 이런 창이 뜹니다.  
디버그 모드와 릴리즈 모드 중 선택하는 부분인데, 사실상 그게 그거예요.



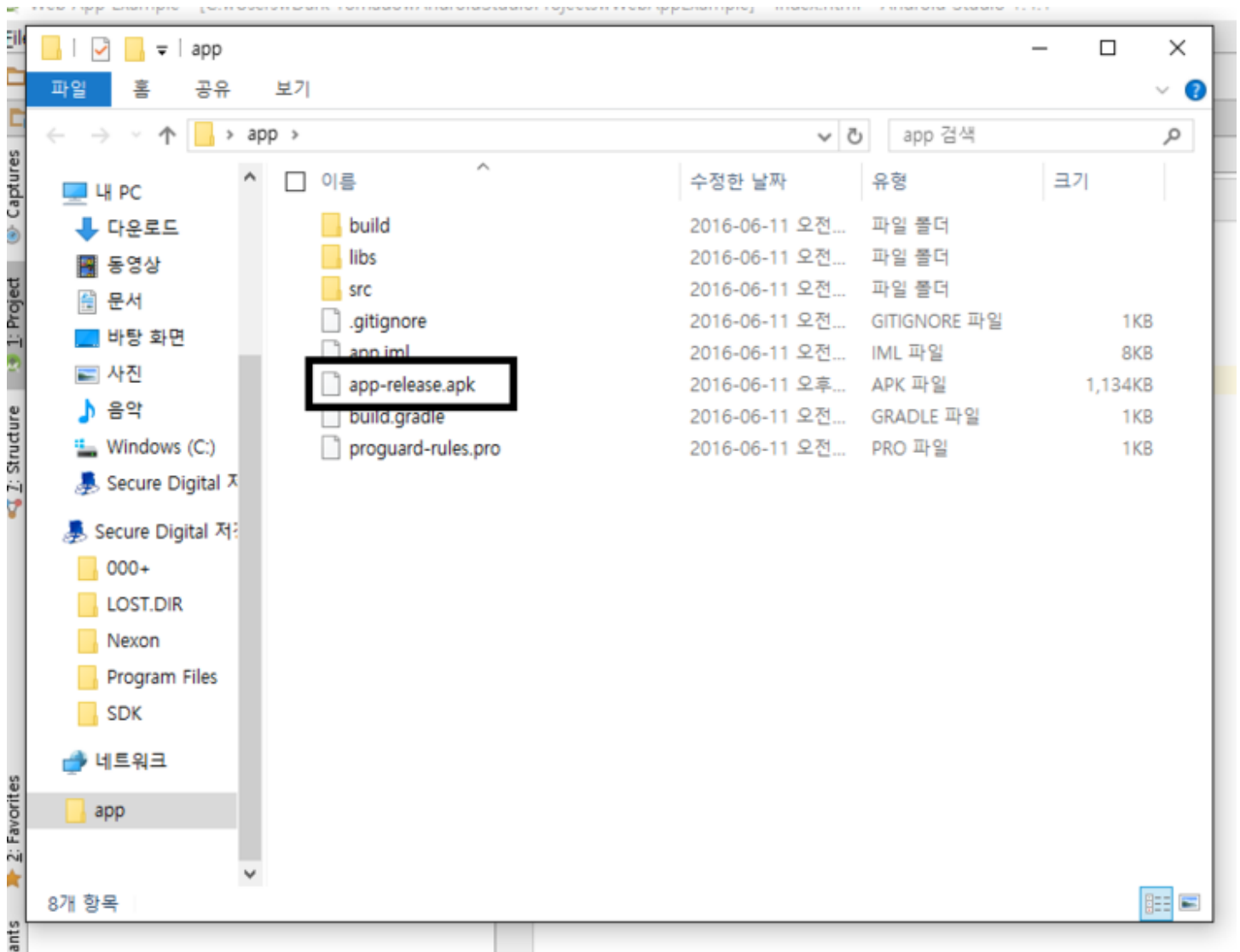
Finish를 누르면, 컴파일이 시작되고,



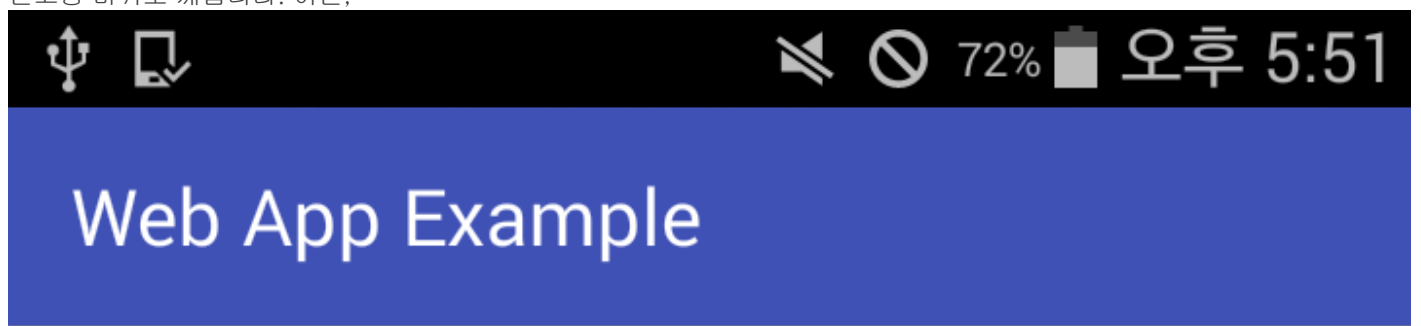
끝나면 끝났다고 뜹니다.  
Show in Explorer 버튼을 누르면,



이렇게 .apk 파일의 위치가 뜹니다. 이걸 설치하시면 됩니다.



설치해보면 한글이 깨집니다. 아~신난다.  
인코딩 바꿔도 깨집니다. 이걸,

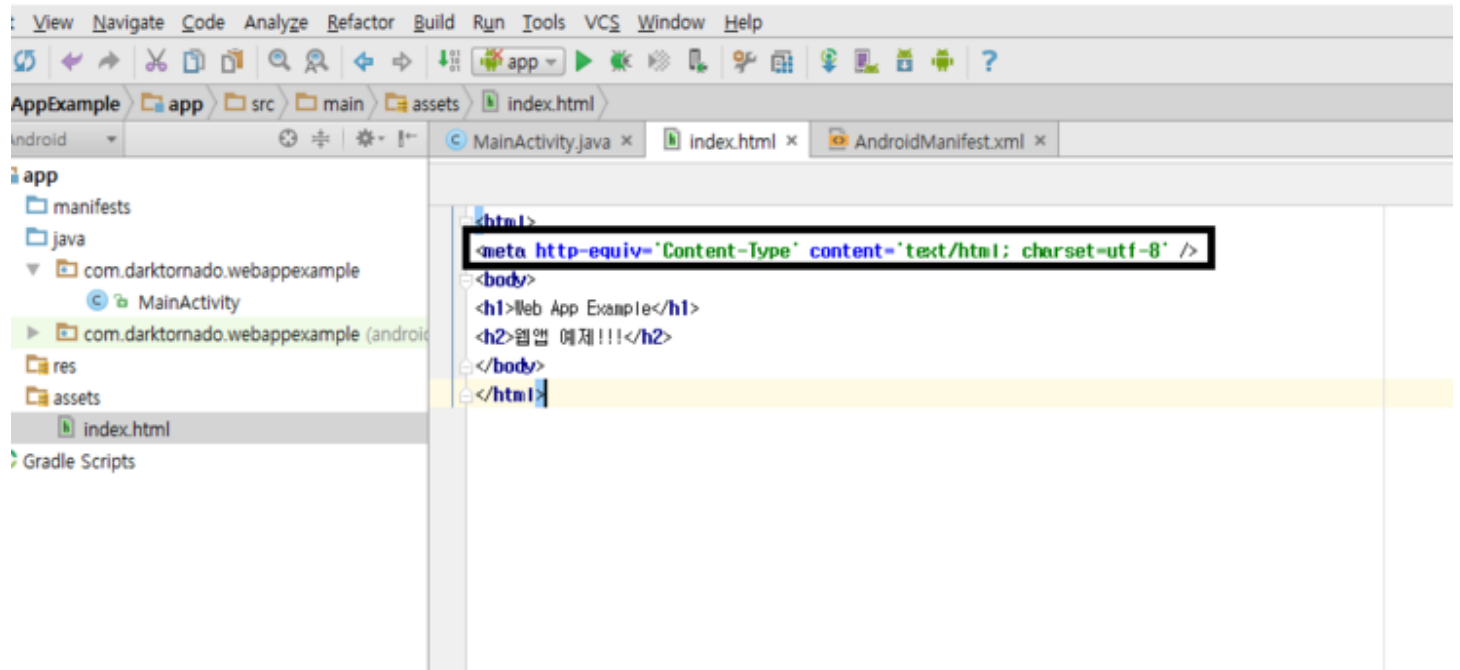


# Web App Example

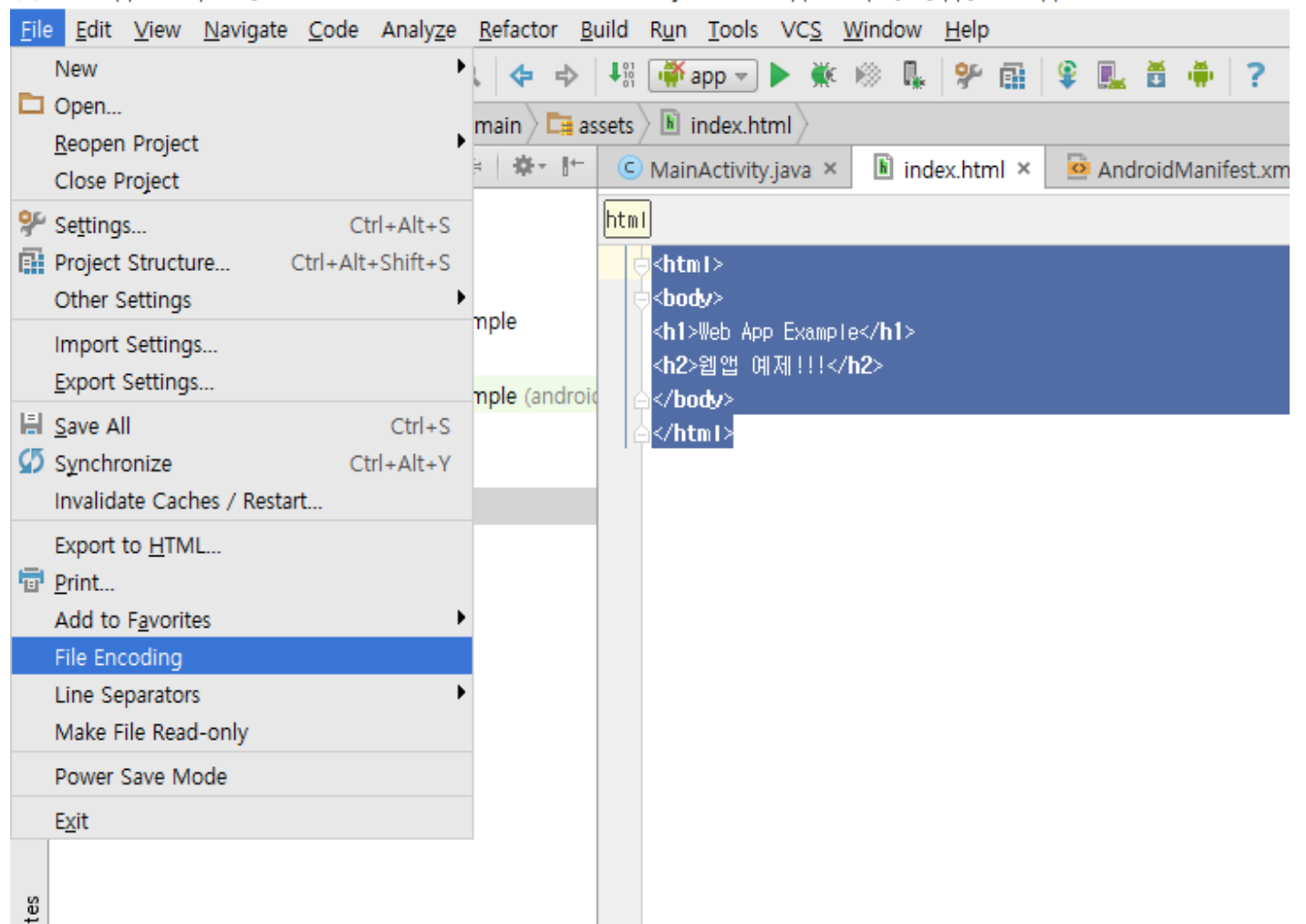
ì>¹ì•± ì~^ì œ!!!



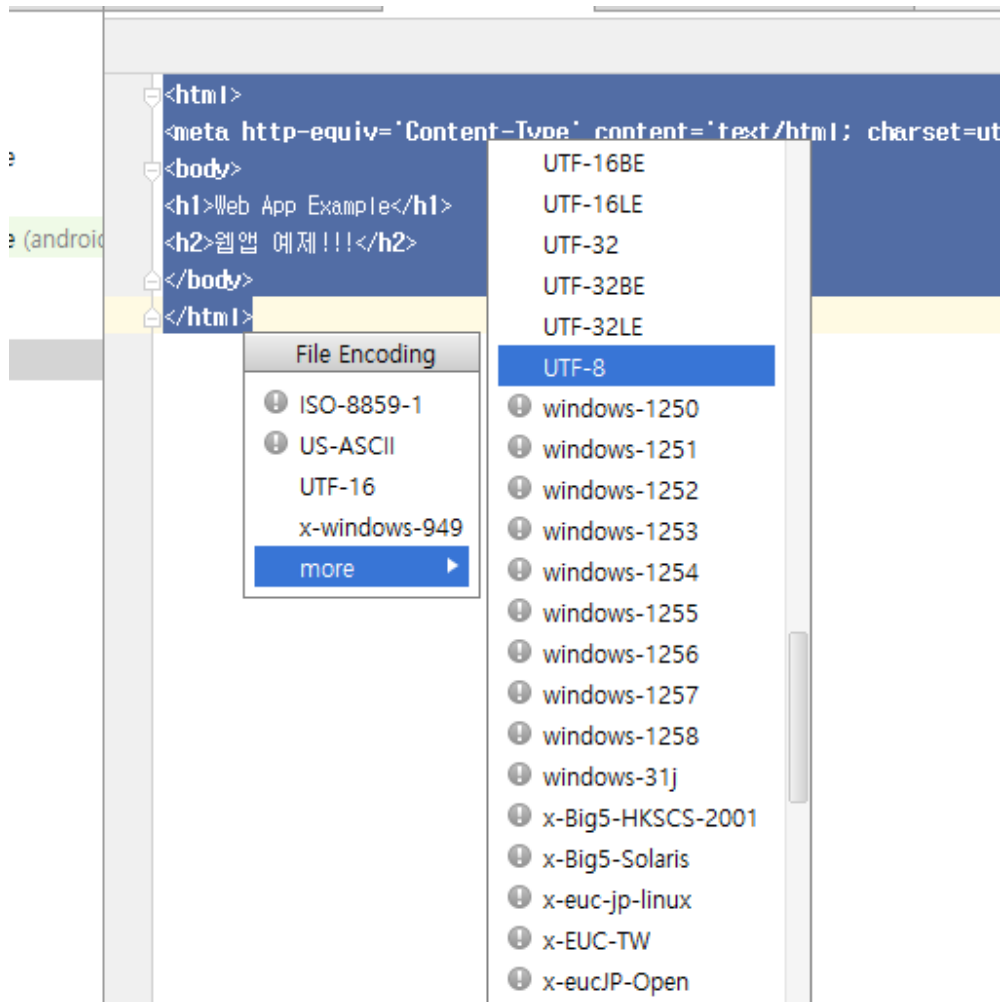
HTML 파일에서 <meta> 태그로 이상한 짓 하시면 됩니다.



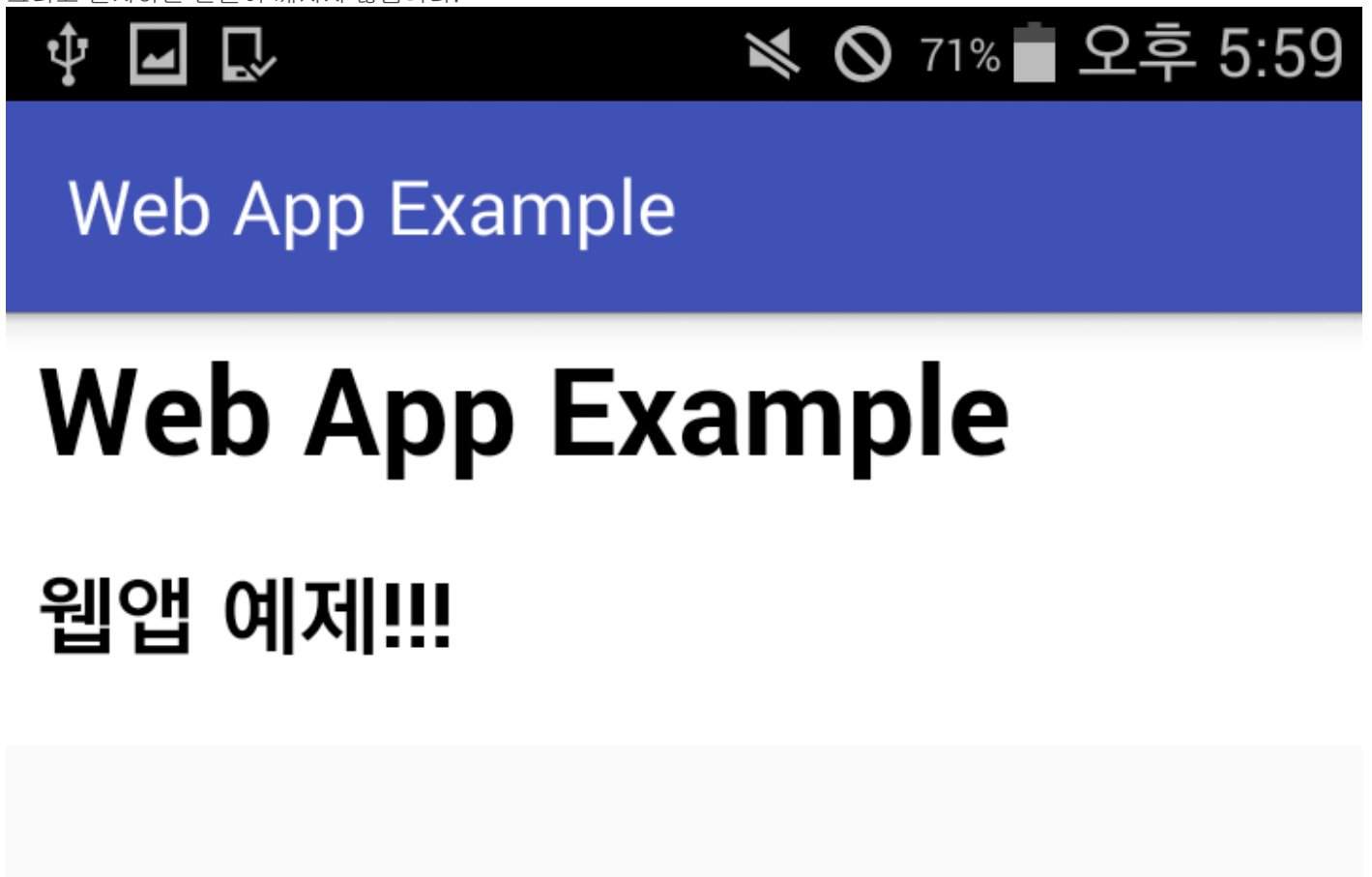
전 html 파일의 인코딩도 utf-8로 바꾸었습니다.





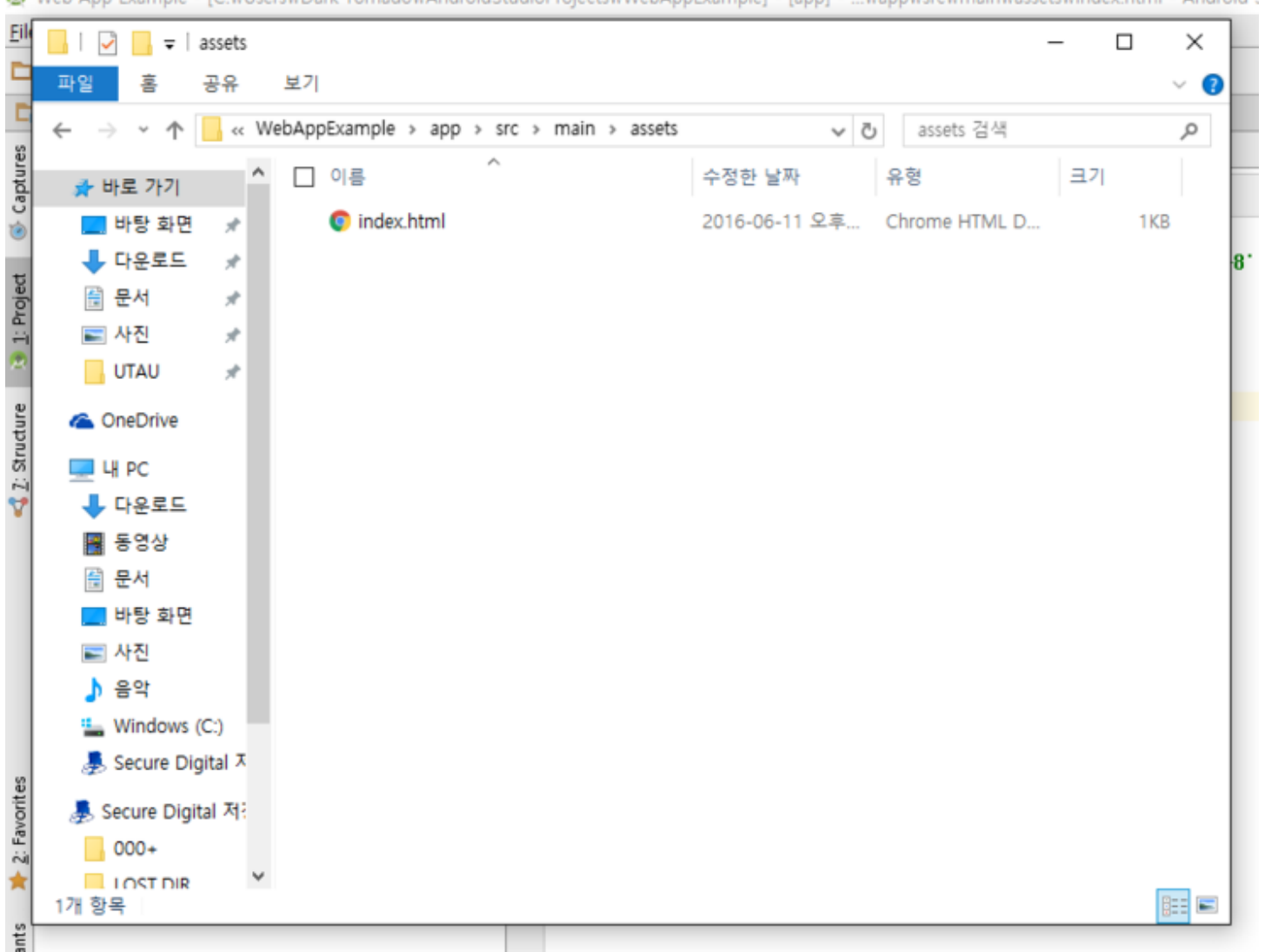


그리고 설치하면 한글이 깨지지 않습니다.



HTML 파일을 직접  
프로젝트 경로WappWsrcWmainWassets

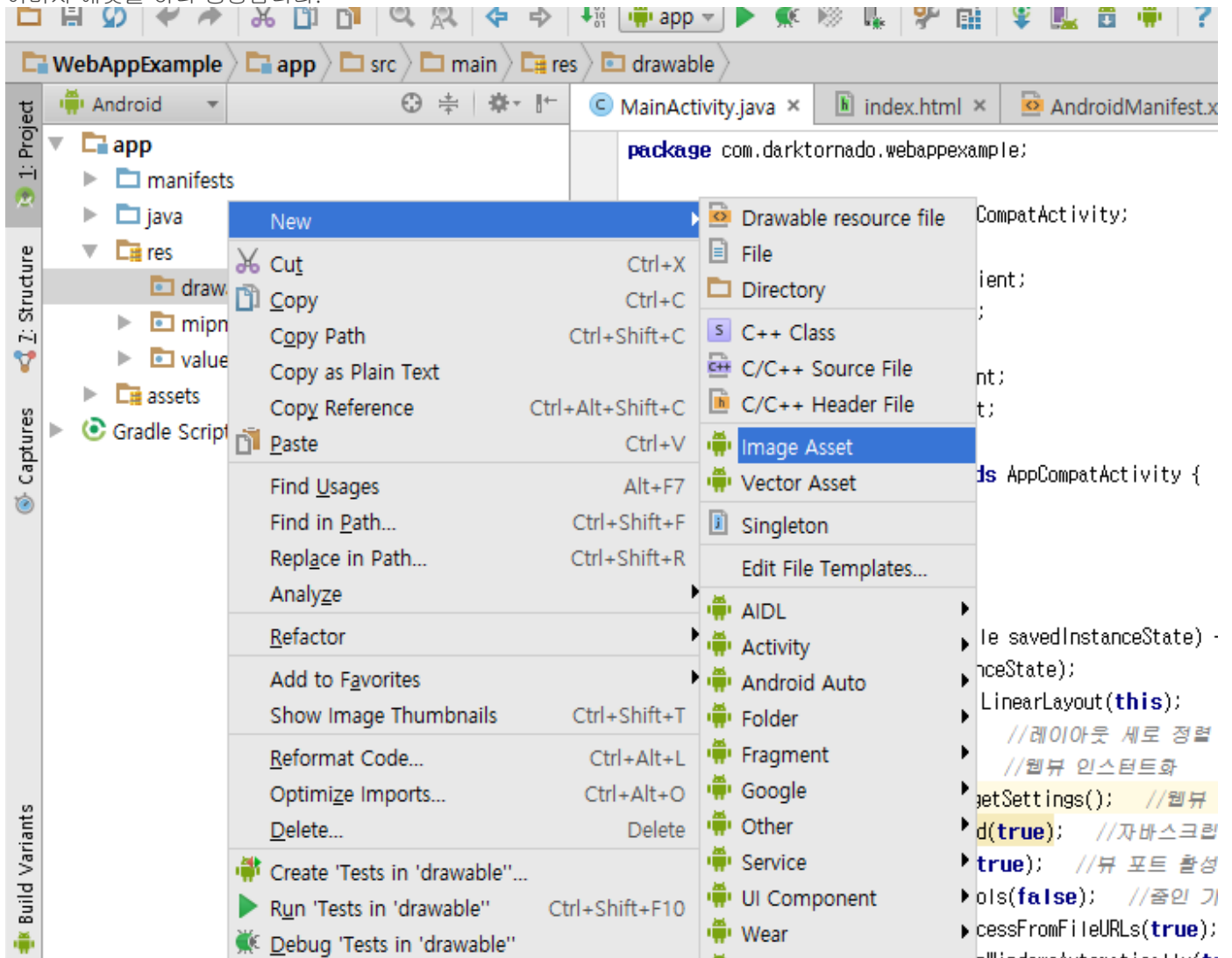
더 안에 넣어드립니다.



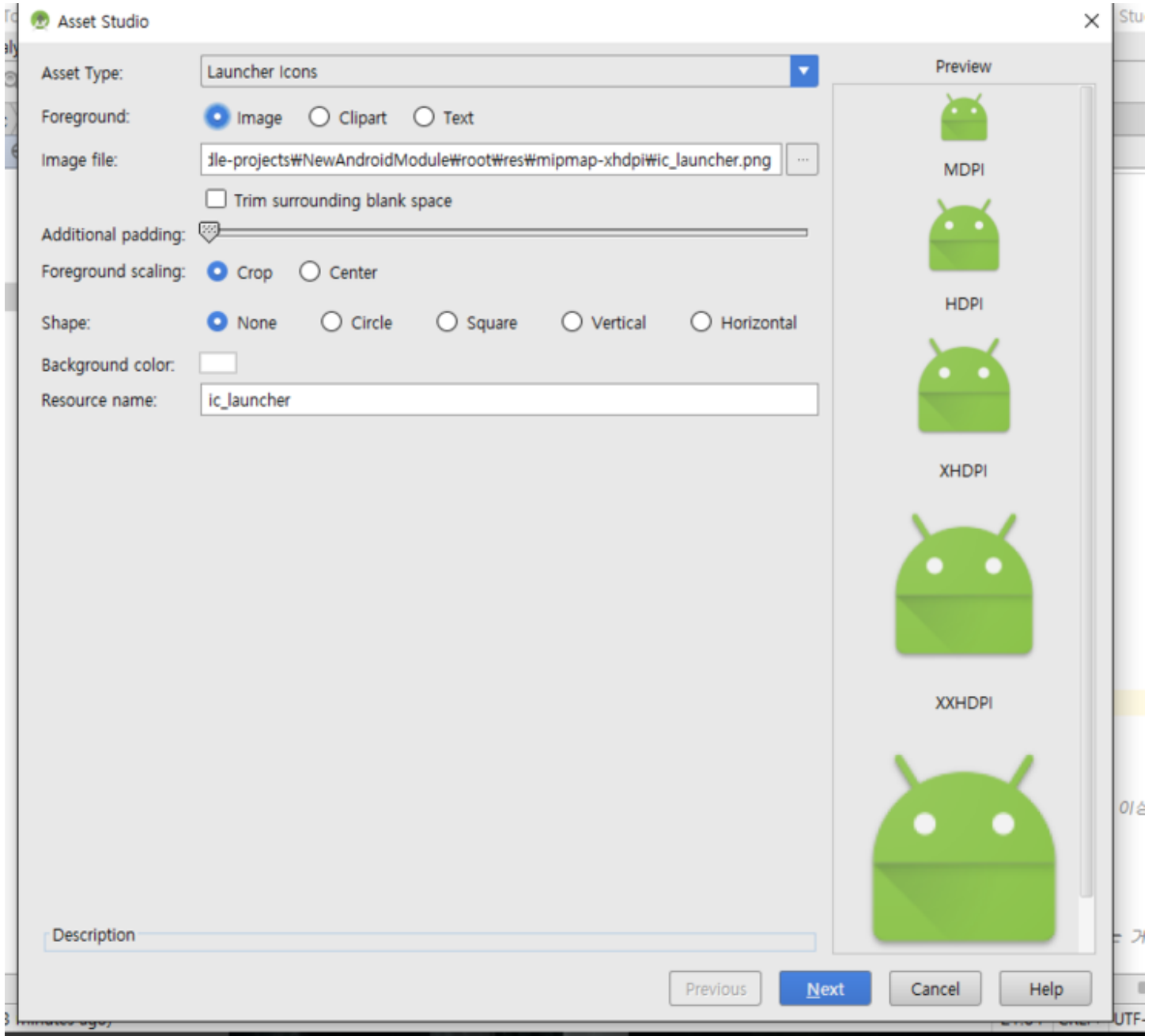
여기서 부터 아마도 웹 개발 하듯이 하시면 됩니다.

+  
아이콘 바꾸기.

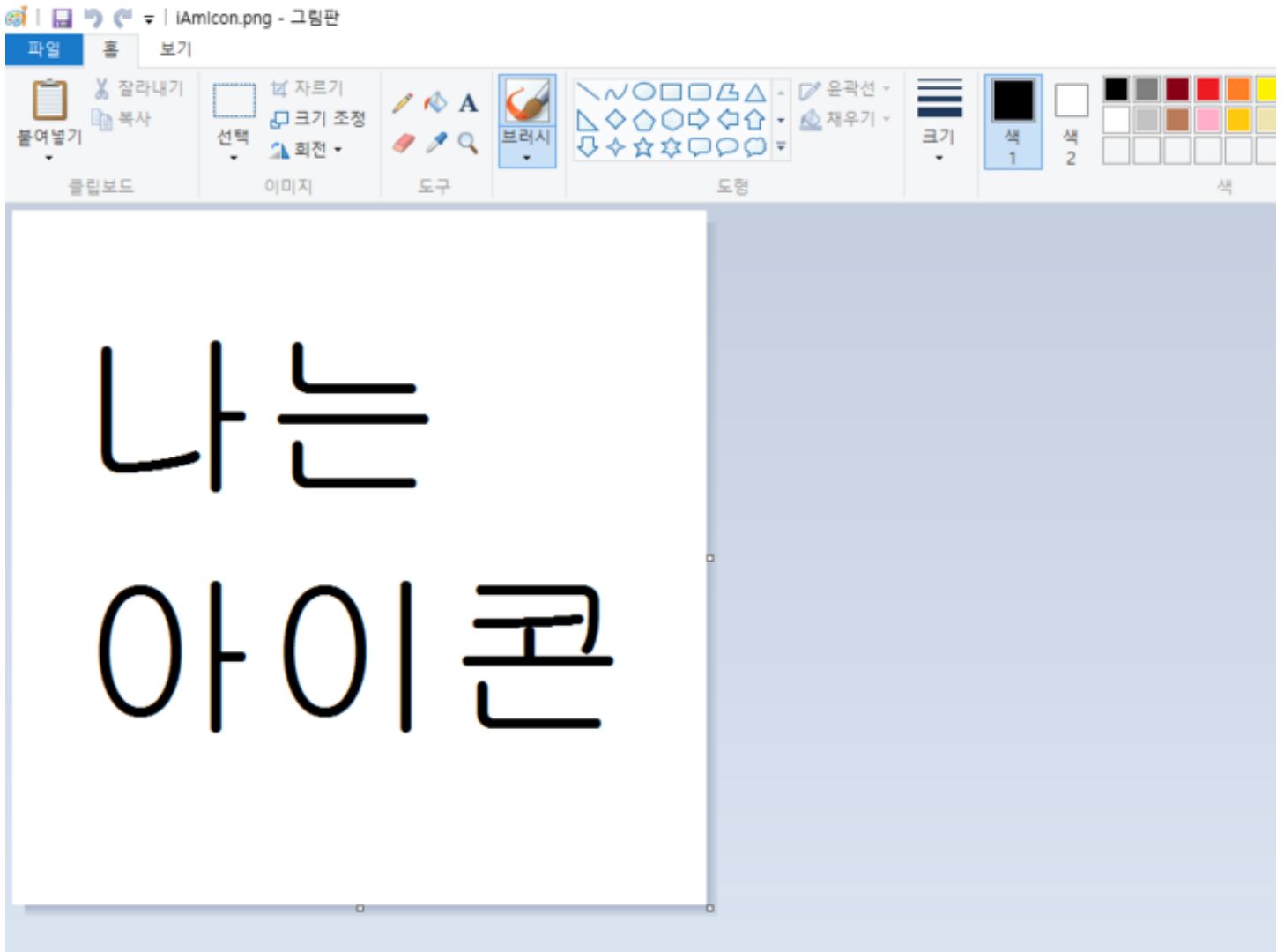
이미지 애셋을 하나 생성합니다.



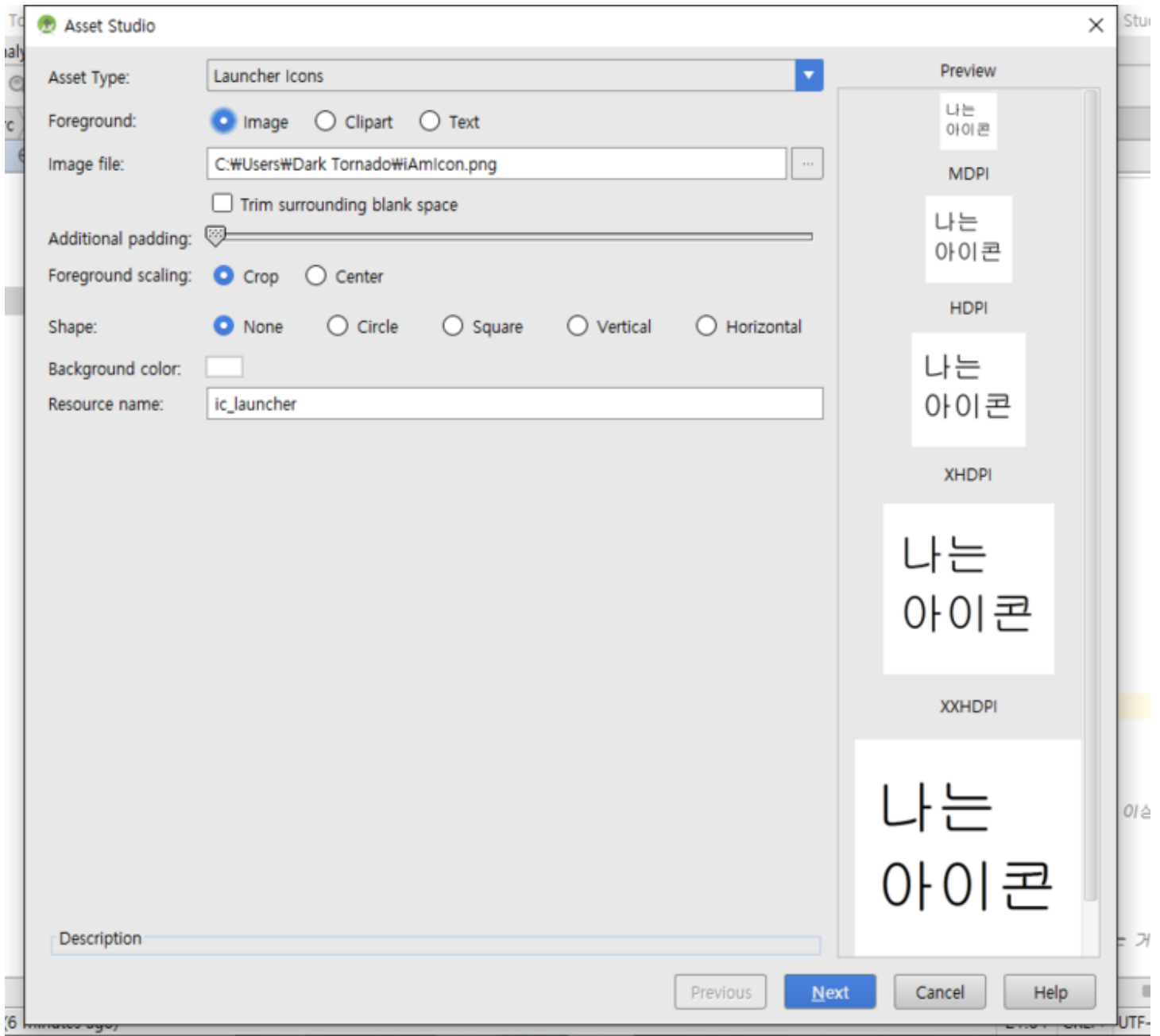
이런 창이 뜹니다.



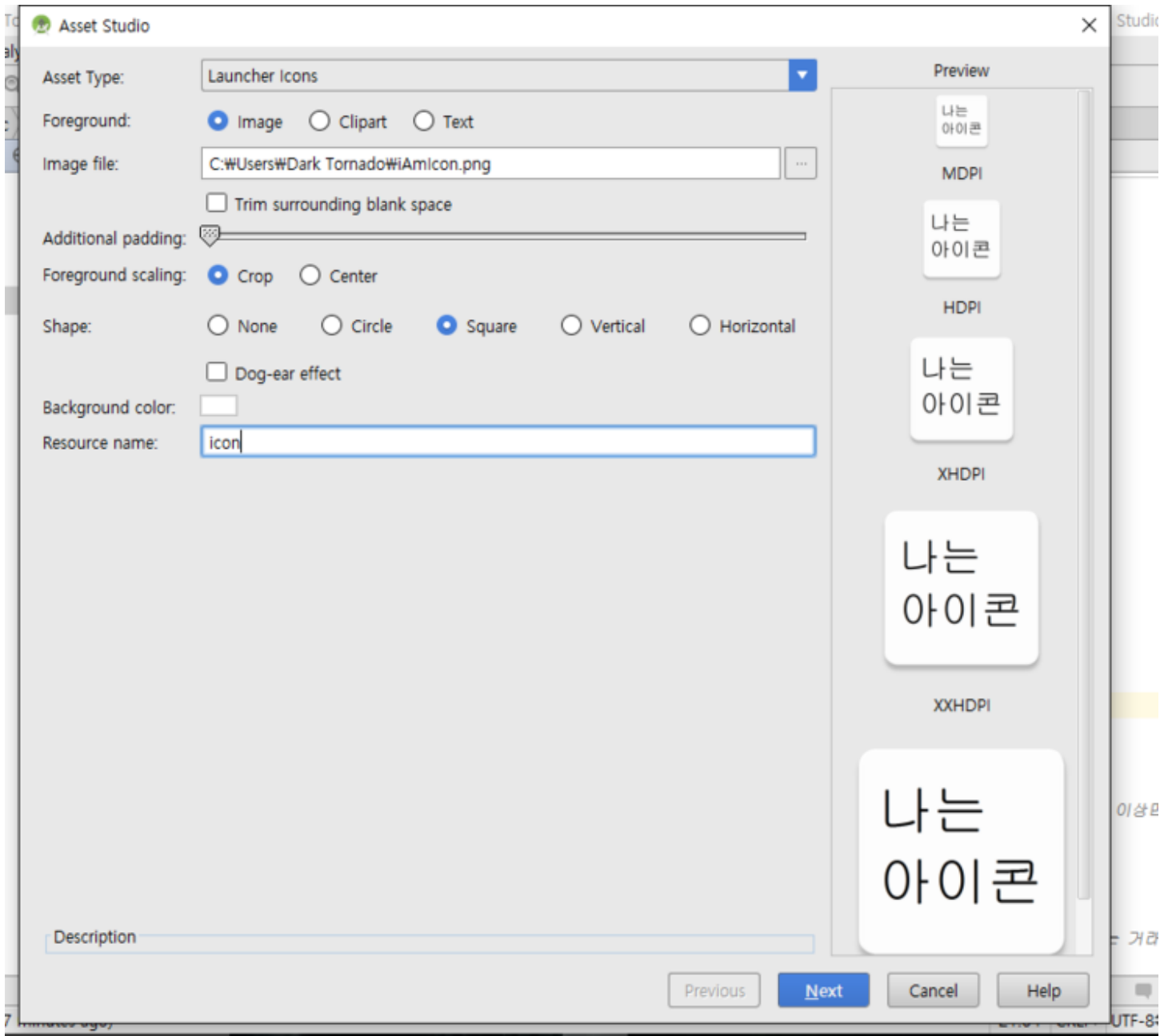
일단 아이콘 하나 급조.



그 아이콘을 불러오시면 됩니다.

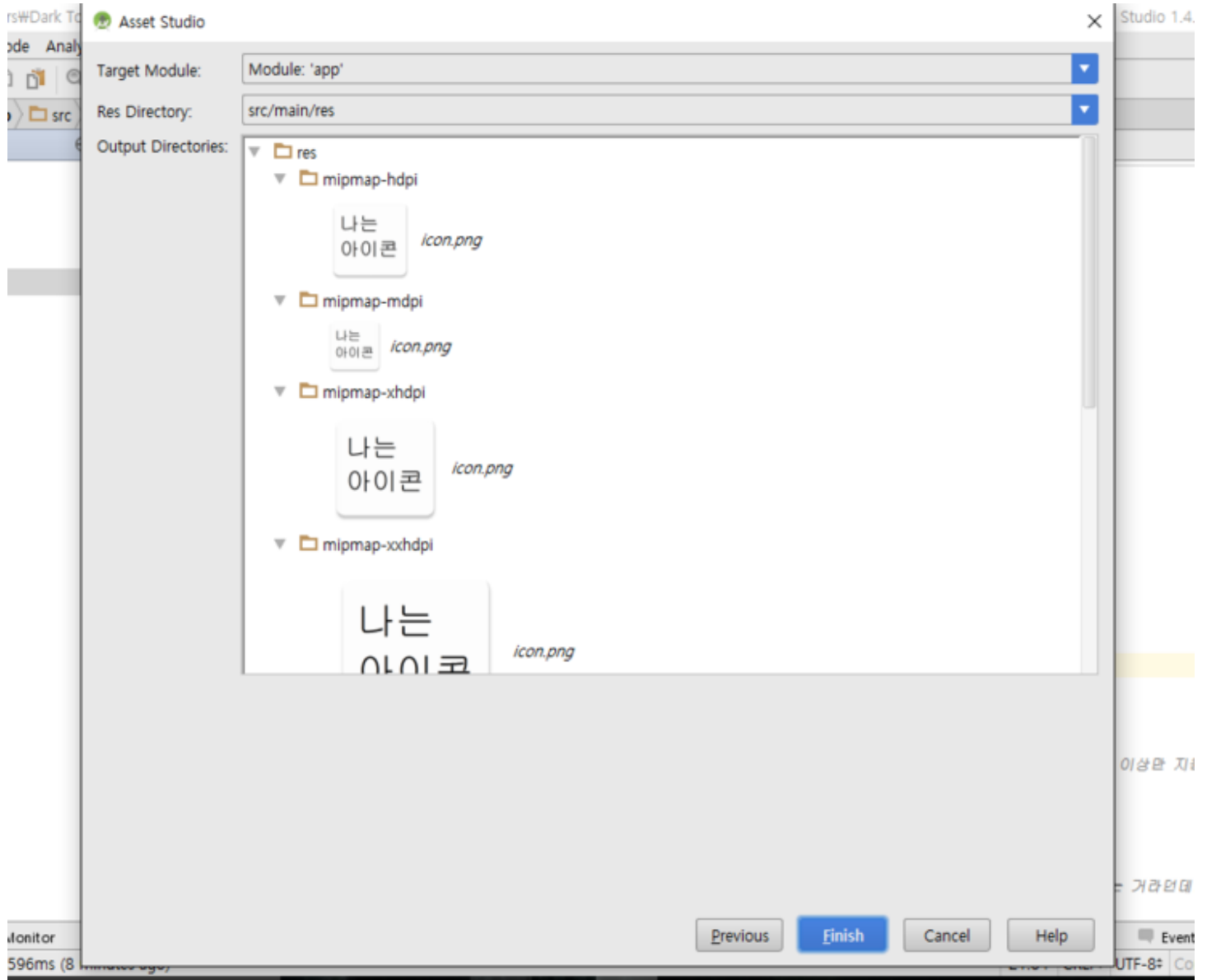


모양이랑 리소스 이름 정하시고,

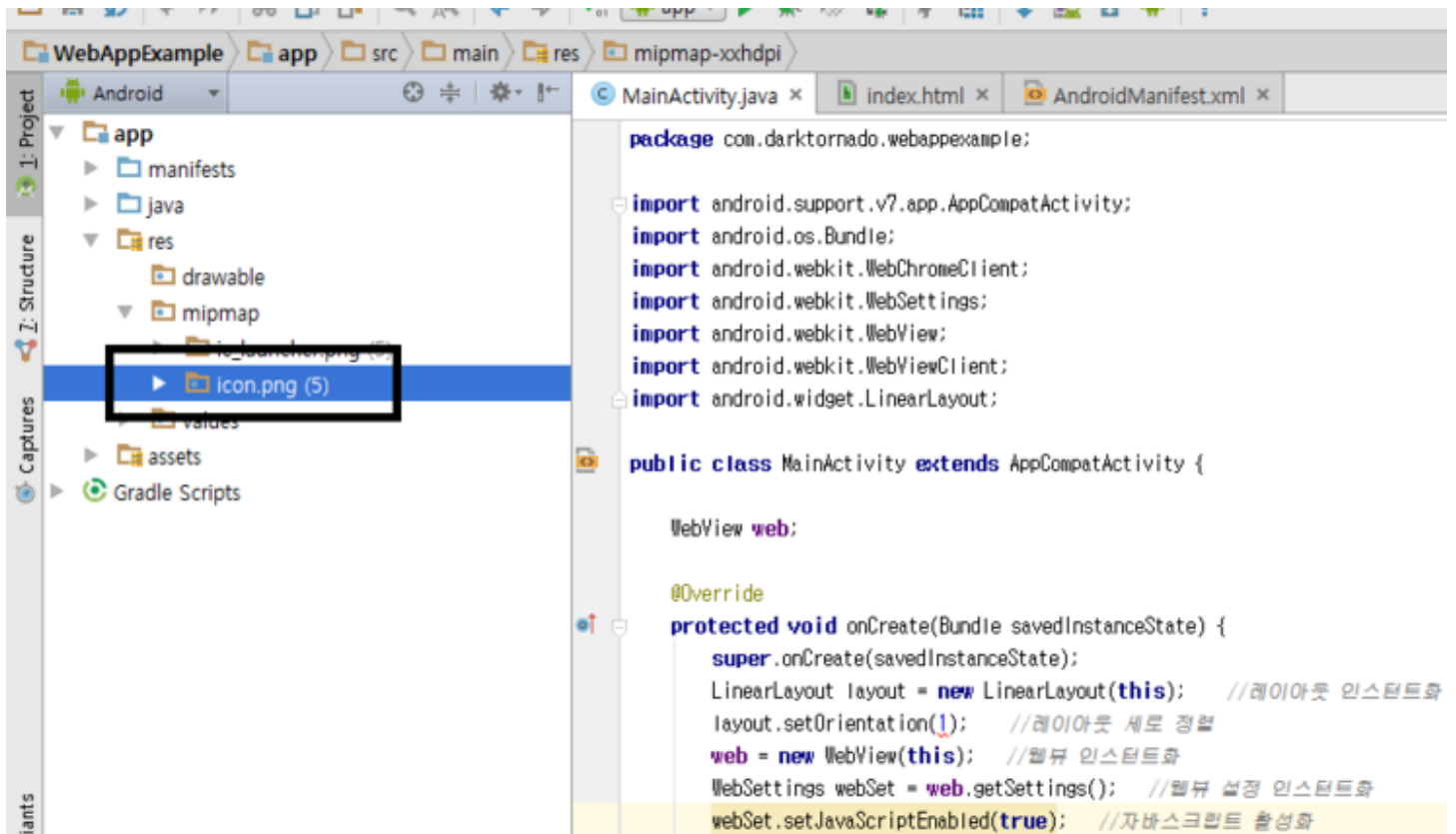


Next 누르고 Finish 누르면 됩니다.

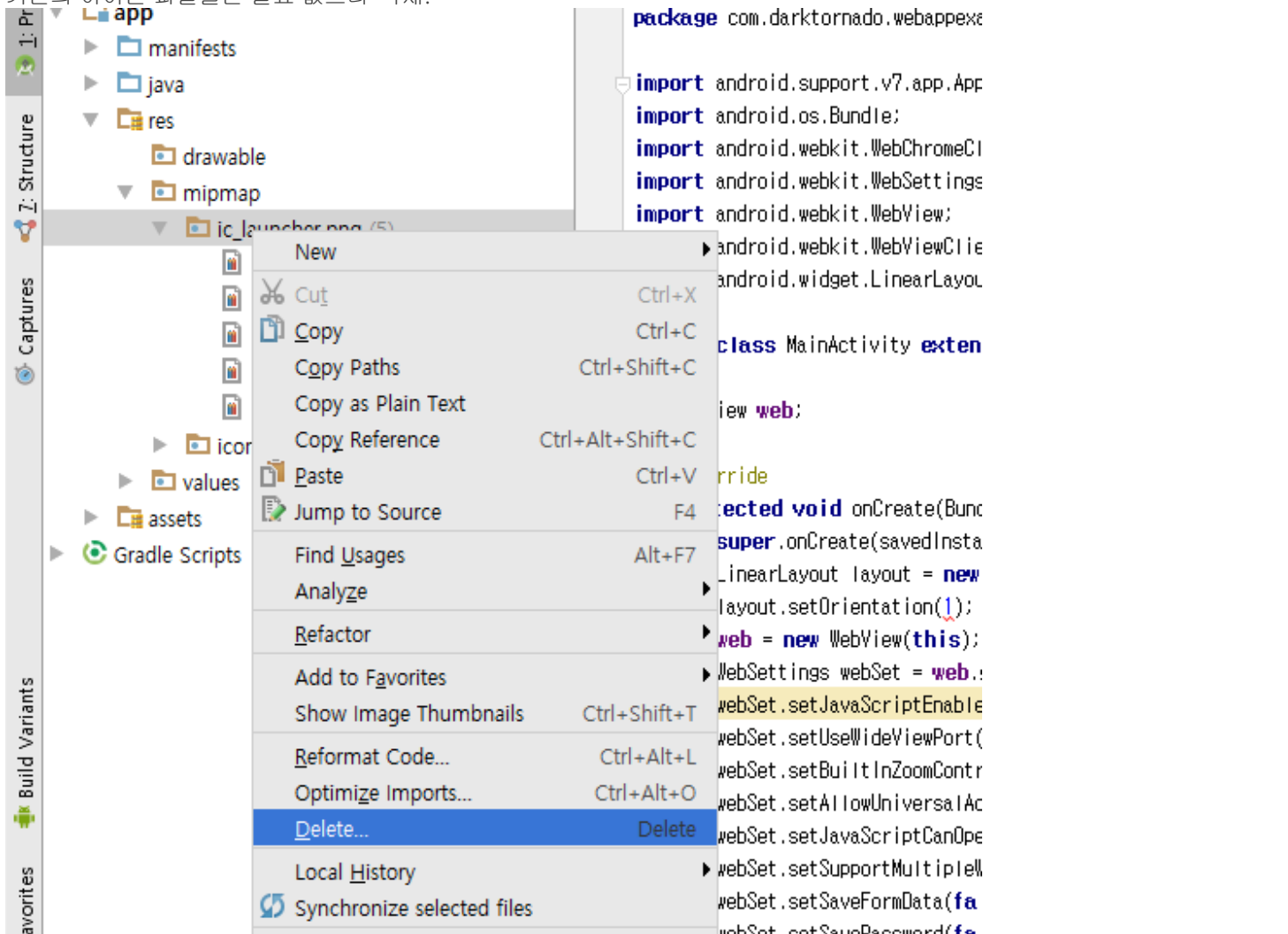




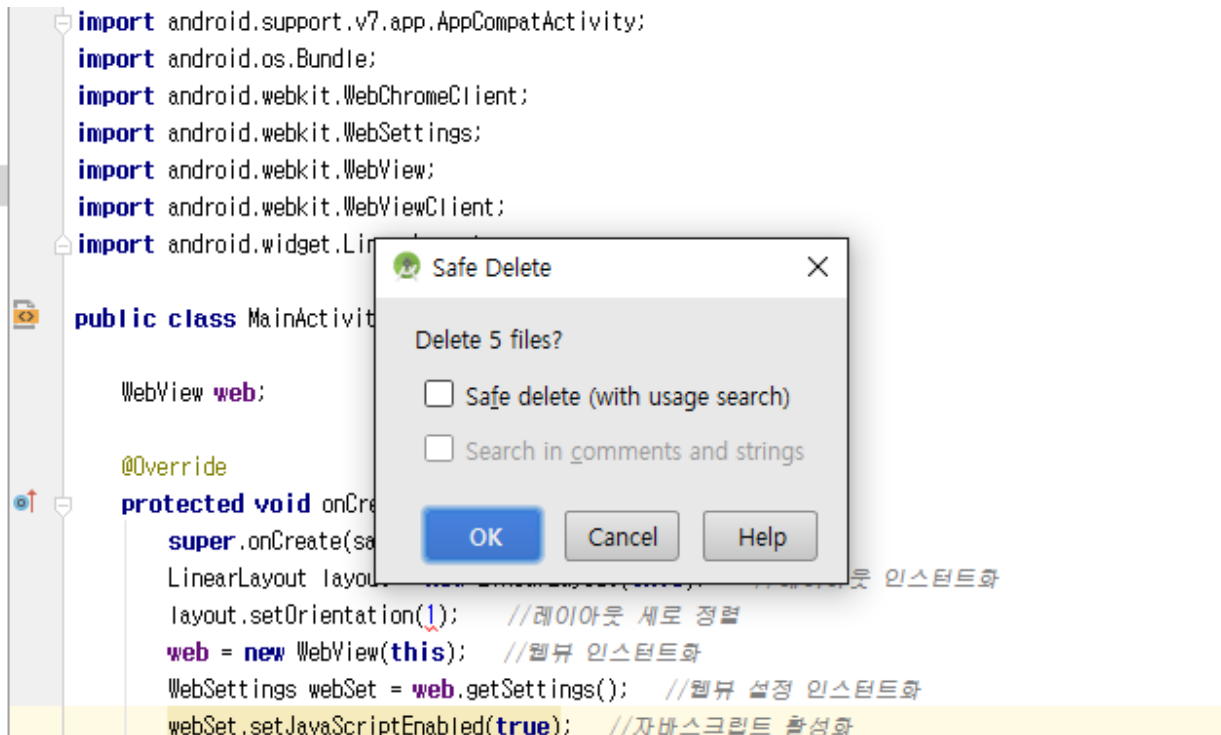
그럼 이렇게 추가되죠.



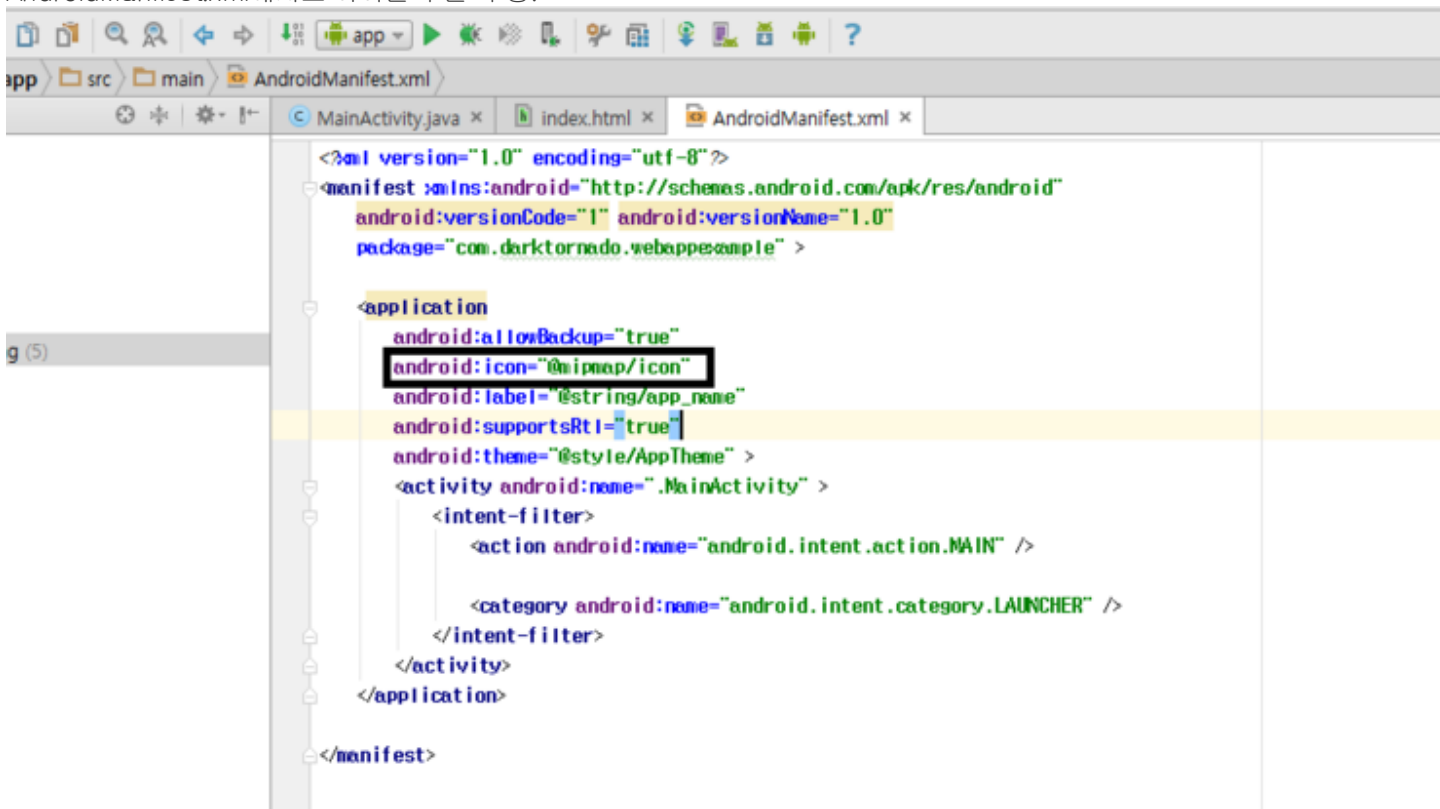
기존의 아이콘 파일들은 필요 없으니 삭제.



뽕

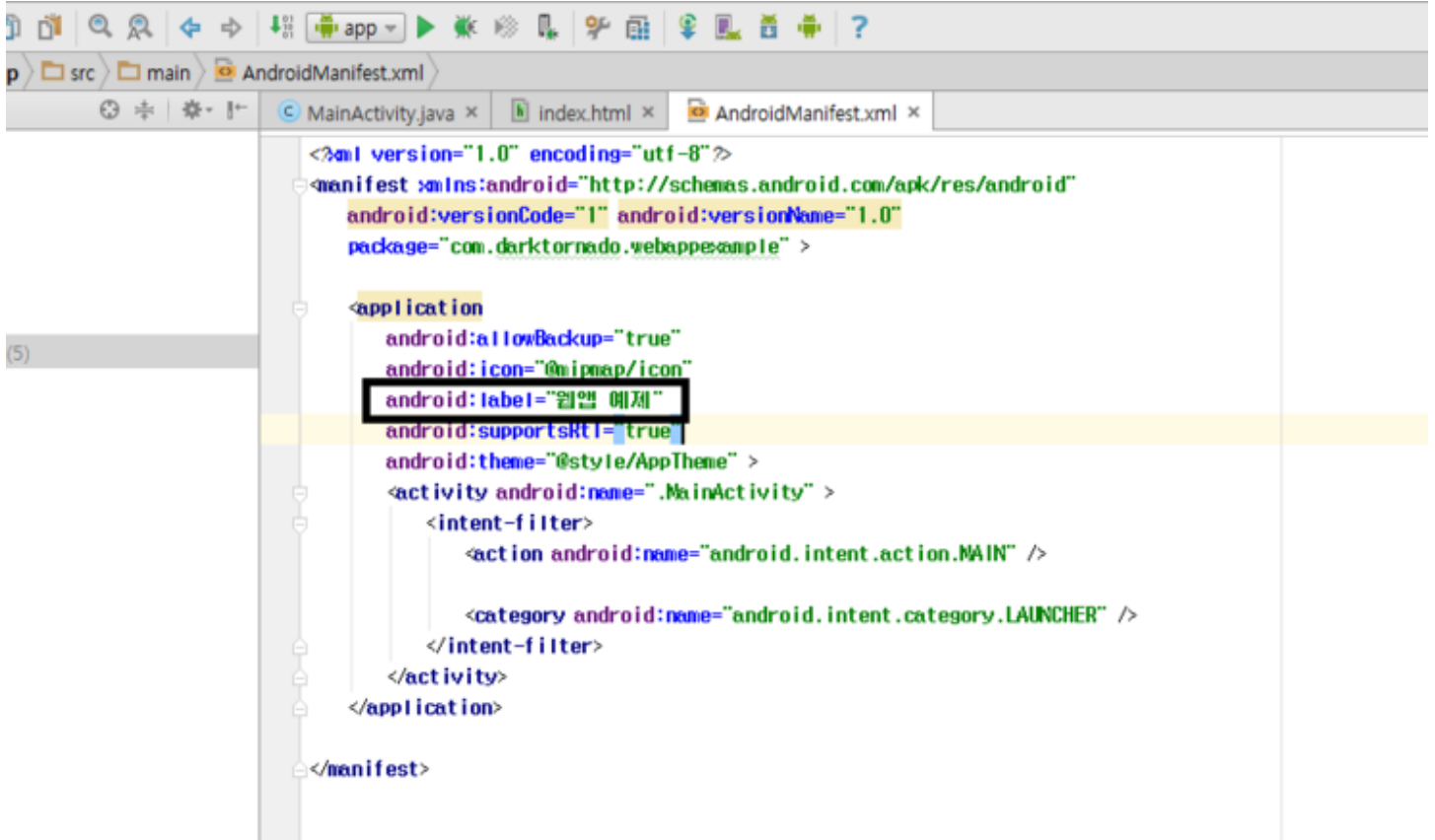


AndroidManifest.xml에서도 아이콘 부분 수정.

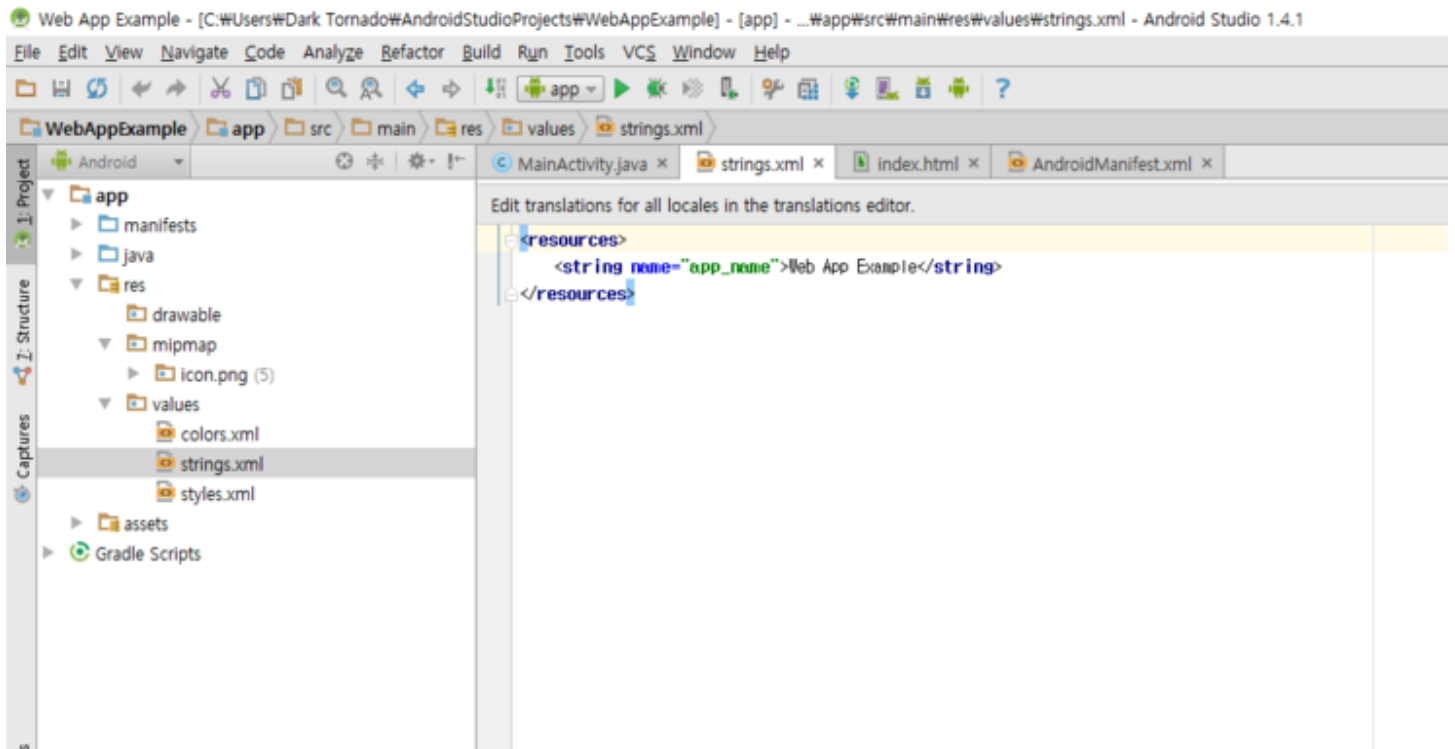


++  
앱 이름 변경

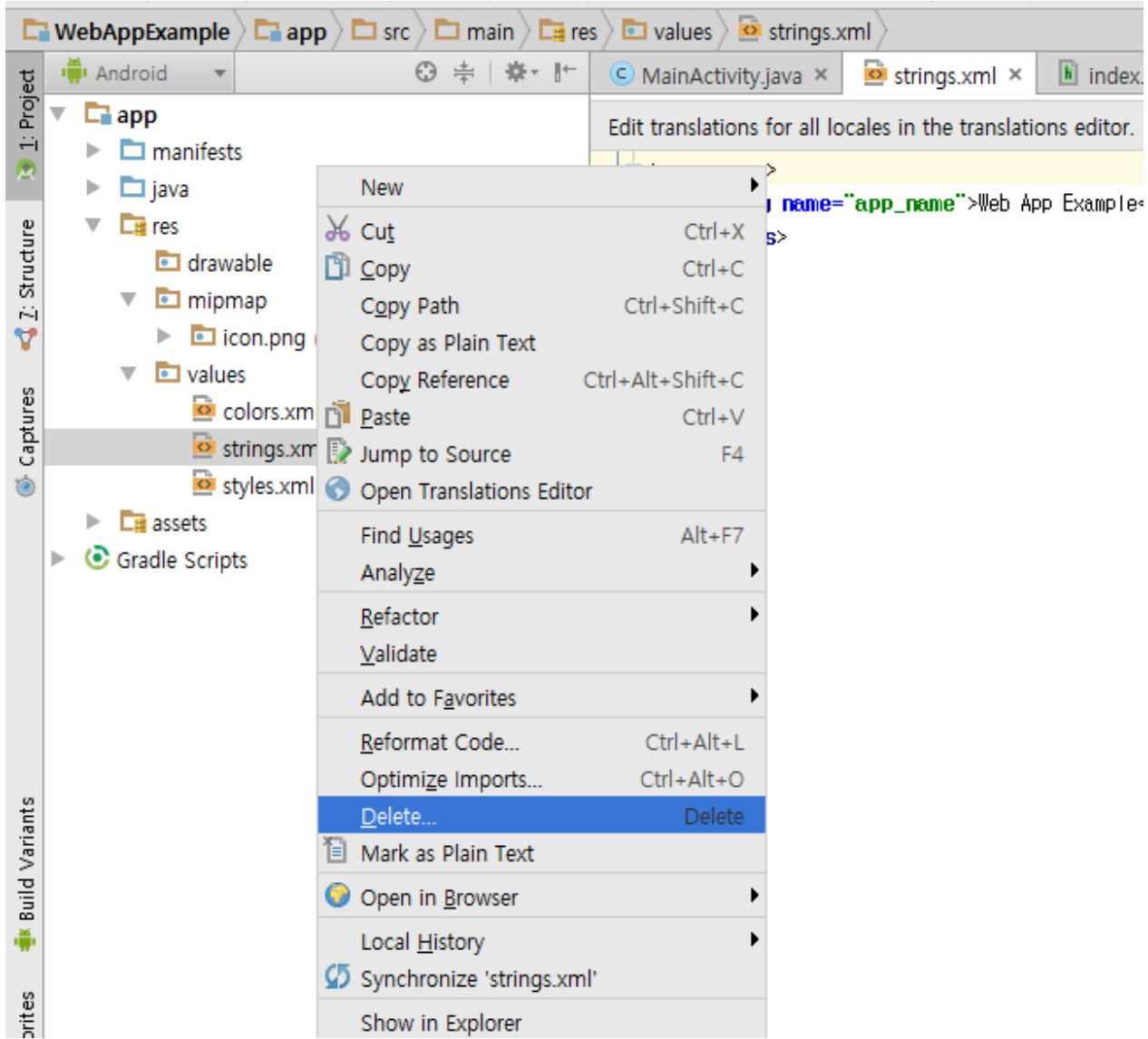
저 부분 수정하시면 됩니다.



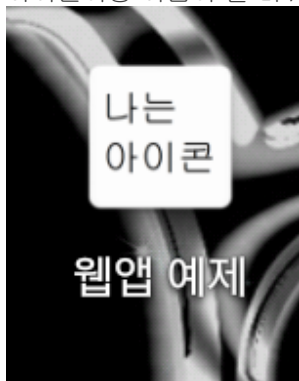
직접 값을 때려넣었으니, 필요가 없어진 이 파일도



삭제



아이콘이랑 이름이 잘 바뀌네요.



앱의 이름을 바꾸면 액션 바의 내용도 바뀝니다.

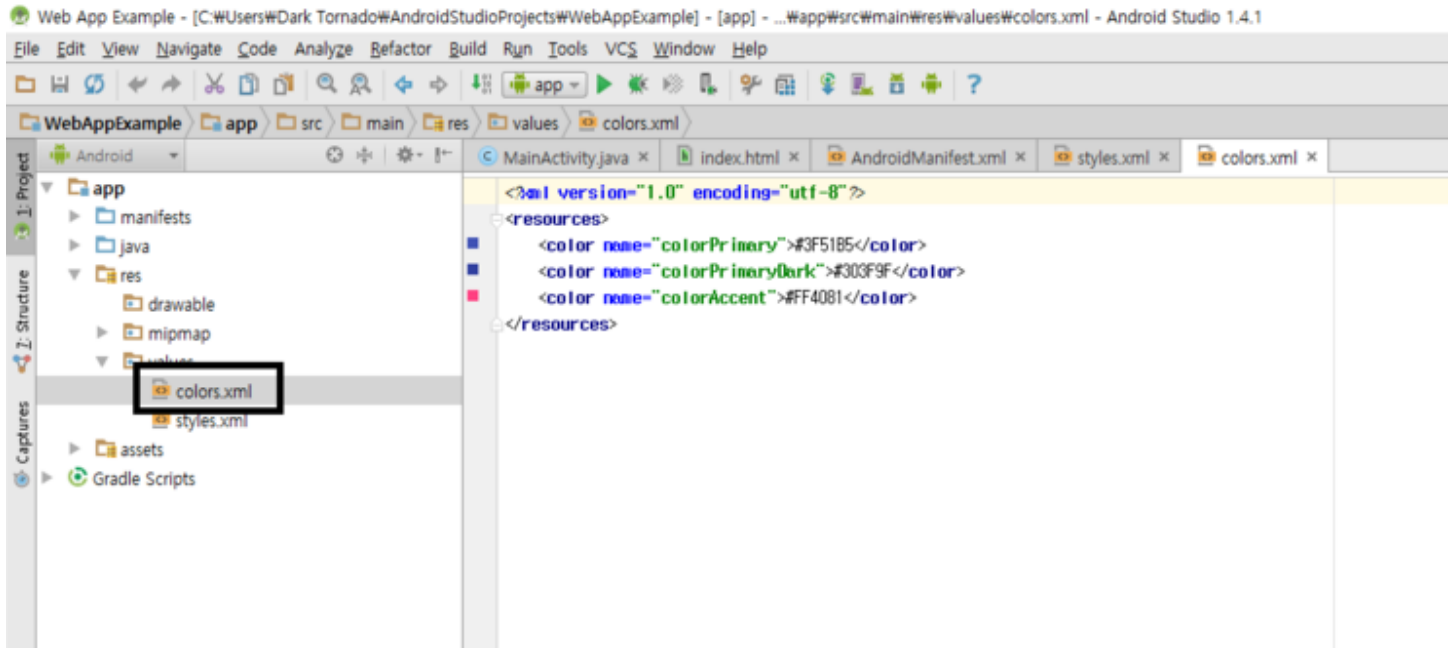


# Web App Example

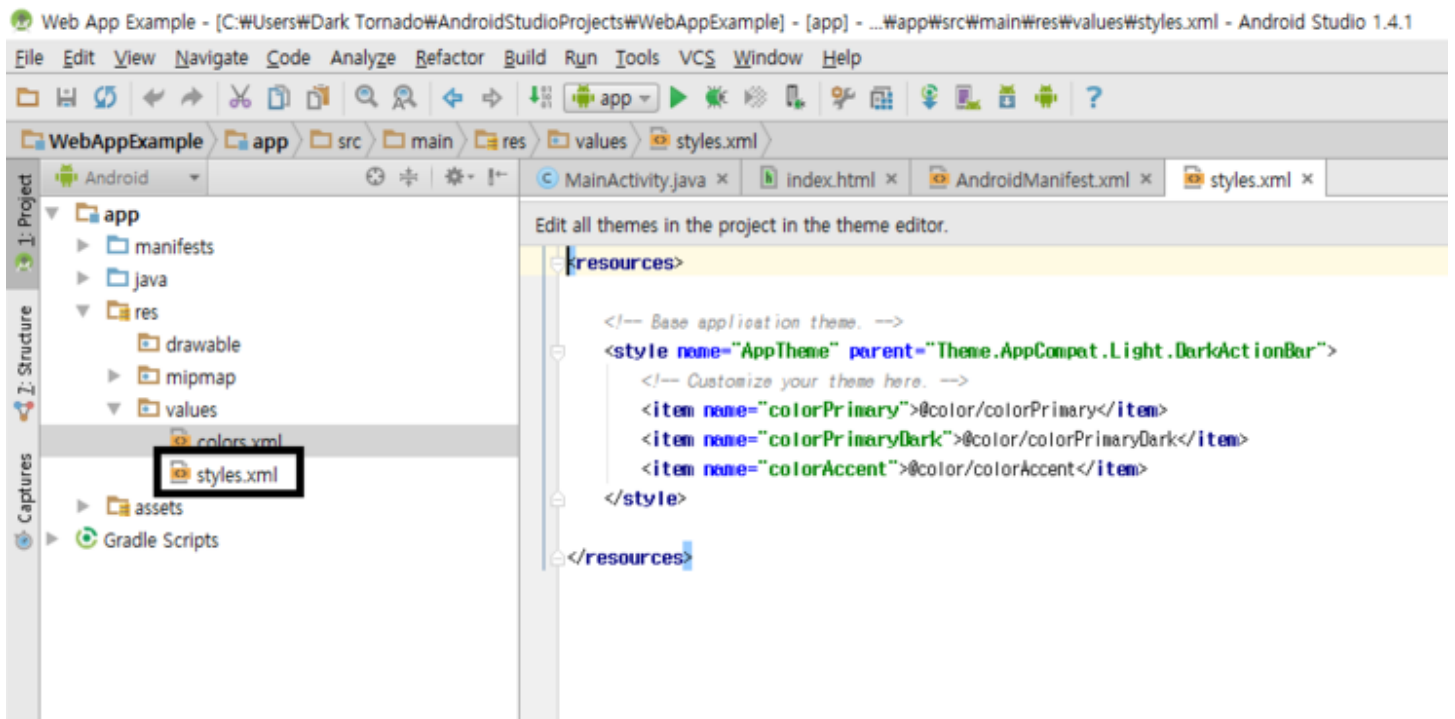
웹앱 예제!!!

++  
액션바 색상 변경 or 삭제.

액션바 색은 이 파일의  
<color name="colorPrimary">#3F51B5</color>  
부분을 잘 건들면 되고,



액션바 삭제는 이 파일의  
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
부분을 잘 건들면 됩니다.



++++

저 프로젝트 파일 첨부합니다.  
참고하라고 첨부한거지, 마음대로 꿀꺽하라고 올린거 아닙니다.

벌써 누구님이 아무 말도 없이 꿀꺽해가지고 패키지명도 안바꾸고 Play 스토어에 올리셨더군요.  
그래놓고 그 앱을 지식IN에서 신나게 홍보. 그리고 그 계정 정지.  
(지식IN에서의 홍보 행위는 신고 대상입니다.)  
이용약관을 어기는 행위에 저의 닉네임이 포함되어 있는게 상당히 불쾌하더군요.

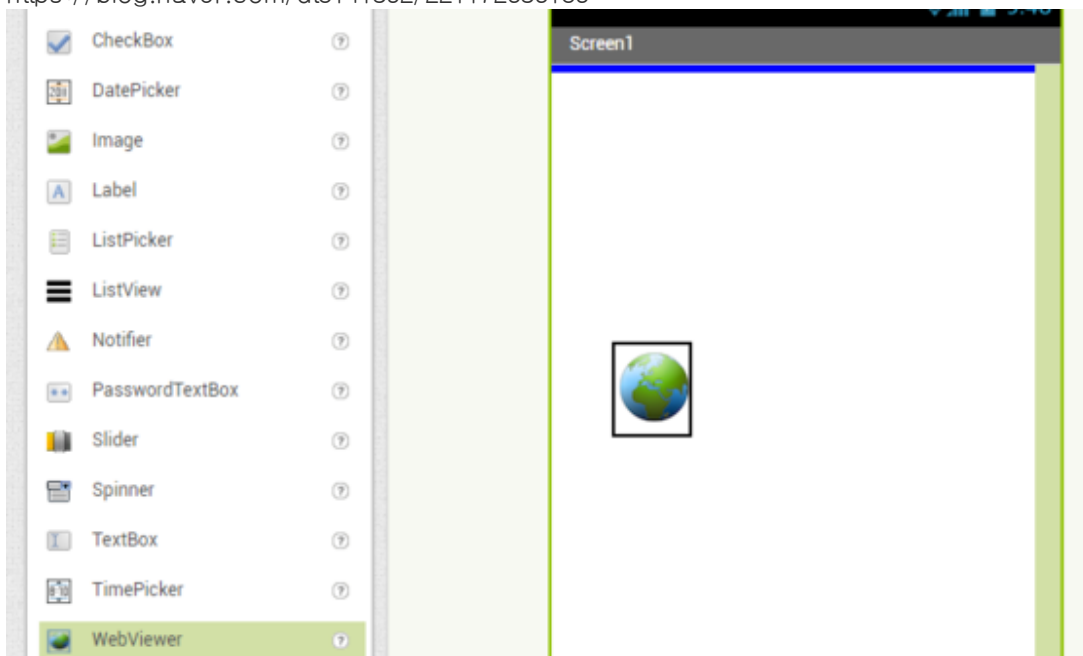
+++++

게시글 스크랩 등은 자유입니다..!

+++++

앱인벤터로 웹앱 만들기

<https://blog.naver.com/dt3141592/221172656136>



앱인벤터로 안드로이드 웹앱 만들기



작년에 안드로이드 스튜디오를 이용하여 웹앱을 만드는 방법을 올렸었는데, 이번에는 앱 인벤터로 만드는걸...  
[blog.naver.com](http://blog.naver.com)