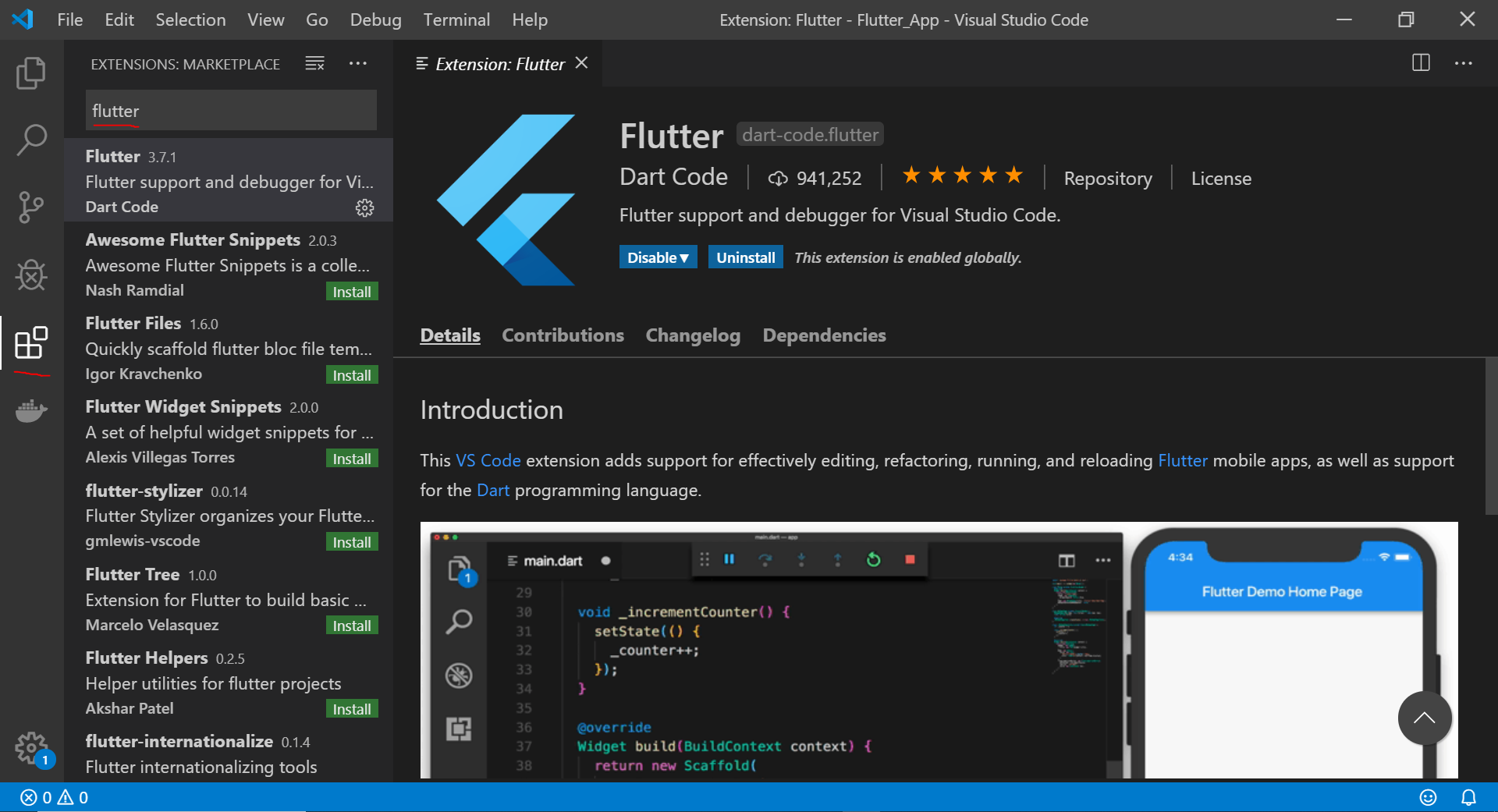
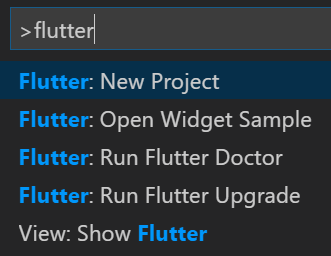
<< Flutter 개발환경 >>

플러터 개발환경으로는 안드로이드 스튜디오, Intelli J, VS Code 등이 있다. 이 중에서 나는 VS Code를 이용하기로 했다. 기존에 깔아 놓기도 했지만 내 컴퓨터의 문제인지는 모르겠지만 안스와 인텔리는 너무 무겁다.

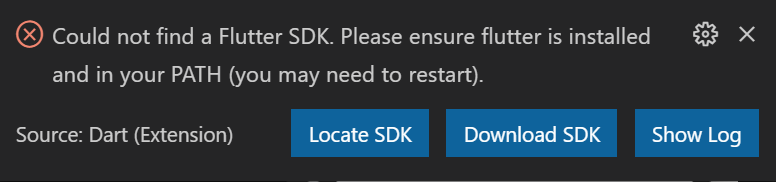
VS Code 설치



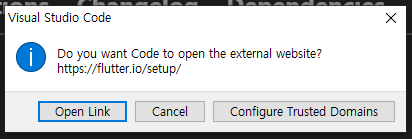
왼쪽 세로 바 메뉴 중에서 블록 모양을 누른다. 해당 메뉴는 Extensions로 필요한 확장프로그램을 설치 할 수 있다. 검색창에 flutter를 입력하여 설치해준다. (해당 사진은 이미 설치된 상태이기에 Uninstall이 나온다) 설치에 상당한 시간이 걸린다. 설치가 다 될 때 까지 기다린다.



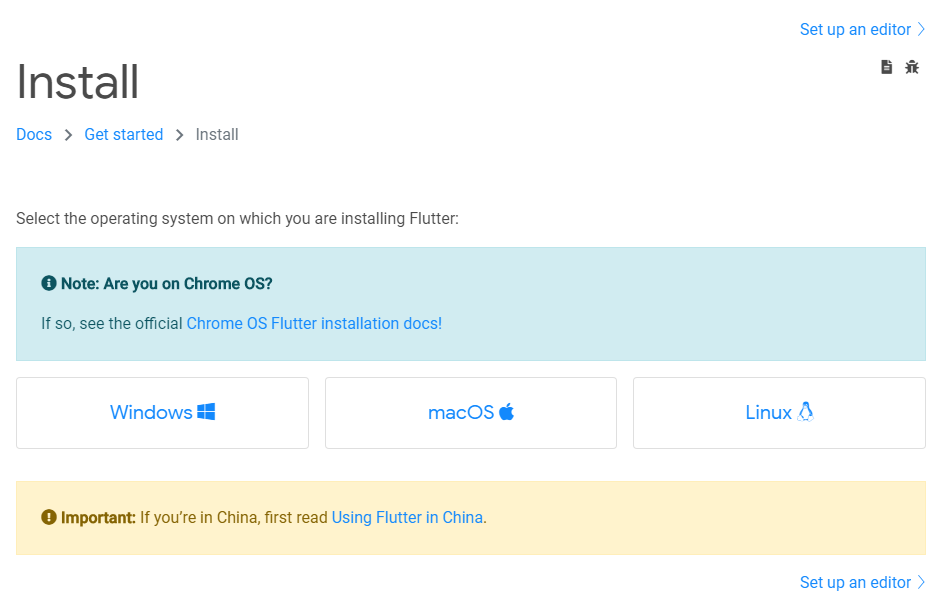
View > Command Palette 혹은 Ctrl+Shift+p를 이용해 명령입력창을 켜서 flutter를 입력하자. 그러면 밑에 flutter로 실행할 수 있는 여러 명령들이 있다. 그 중 Flutter: New Project를 클릭하자.



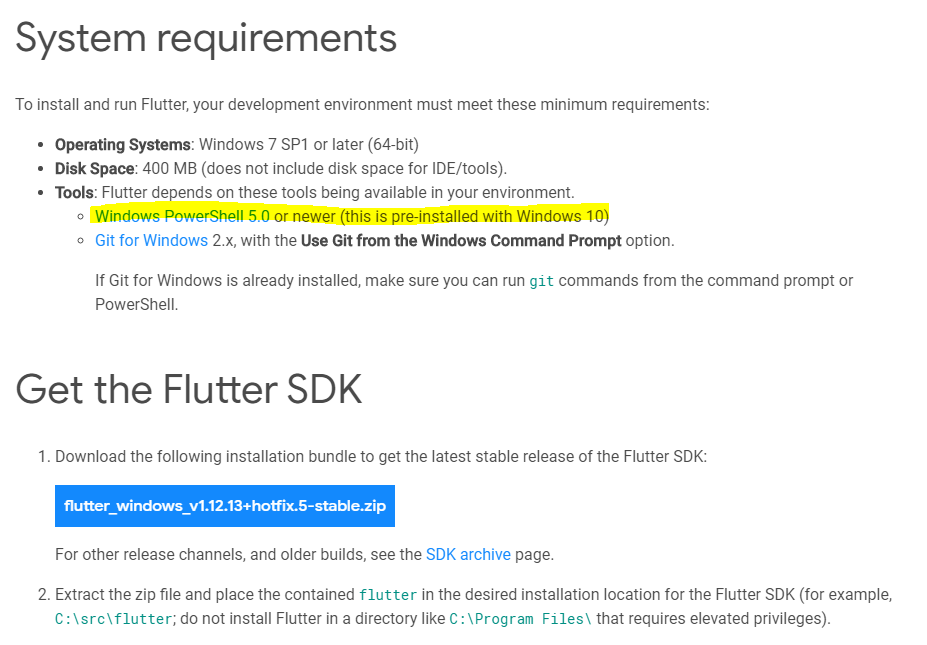
클릭하면 Flutter SDK를 설치하지 않은 상태라 추가 설치를 하라고 경고가 뜬다. 해당 경고에서 Download SDK를 누르면 SDK 설치 파일이 있는 주소로 갈 수 있는 링크로 연결된다.



VS Code에서 해당 URL을 정말 열 것인지 물어본다. 외부 링크가 걸린것이기에 보안 문제상 뜨는 듯 하다. Open Link를 클릭하면 해당 홈페이지가 열린다.



홈페이지가 열리면 각자 설치하는 OS에 맞게 선택하라고 뜬다. 여기서 나는 Window에서 개발을 하기에 Window를 클릭했다. 맥을 이용하거나 리눅스를 이용한다면 해당 버튼을 클릭해서 이용하면 되겠다.



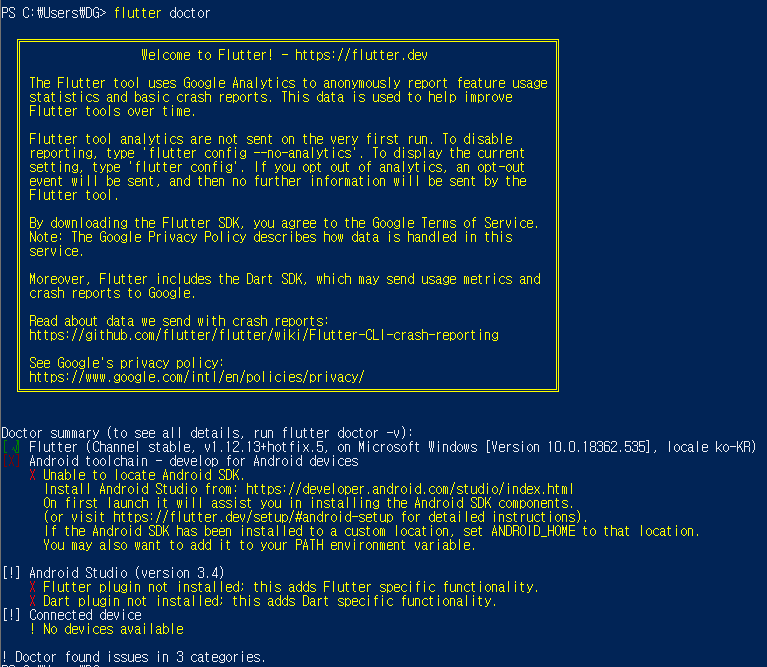
윈도우를 클릭하고 넘어간 페이지에서 보면 요구사항이 적혀있다. PowerShell 5.0 이상, git 2.x 가 필요하다. 파워쉘은 윈도우10에서는 이미 설치되어 있으므로 문제되지 않는다. (윈도우10을 이용하지 않으시는 분들이라면 별도로 검색하여 추가하기 바란다) Git의 경우는 git으로 해당 sdk를 클론하여 받을 경우에만 필요하지 아래에서 제공하는 zip을 이용할 경우는 필요하지 않다. 여기서 나는 zip을 이용해 설치할 것이기에 git 설치는 넘어가기로 한다.



Zip을 풀어서 안에 있는 flutter파일을 원하는 경로로 옮긴다. 나의 경우는 D:\flutter 로 옮겨놨다.



D:\flutter\bin 환경변수를 추가하여 어디서든 flutter를 쓸 수 있게 해줘야 한다.

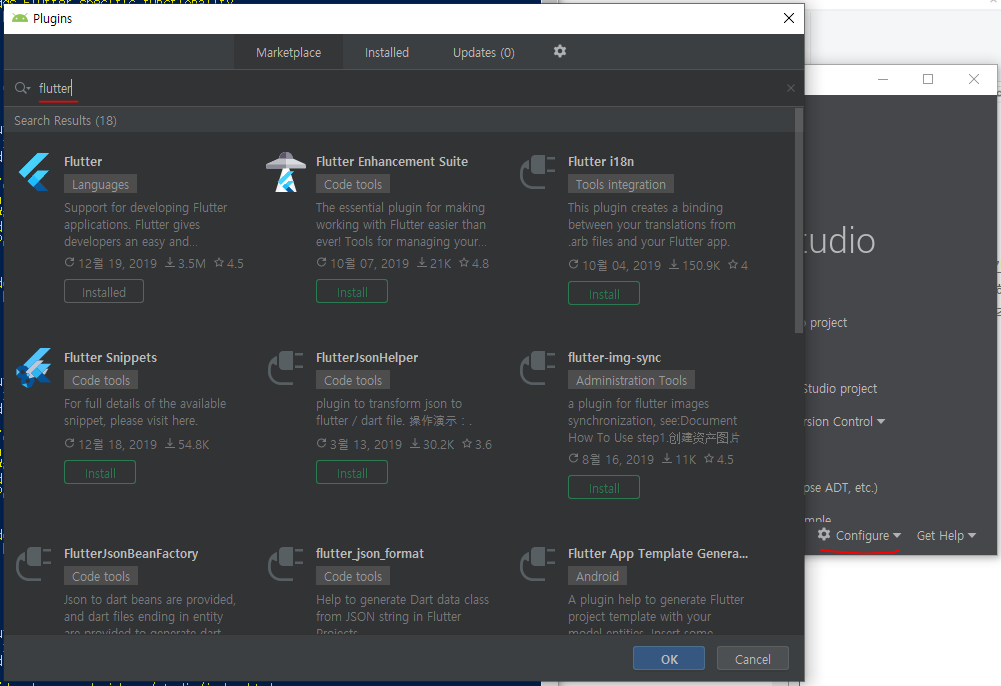


설치가 제대로 됐는지 확인하기 위해서 flutter doctor이라는 명령어를 실행해야한다. PowerShell을 켜서 flutter doctor을 입력해서 확인해보자. 해당 기능은 flutter의 환경세팅에서 빠진 것이 있는지 알려주는 명령이다.

초록색 체크 표시는 성공적으로 설정된 항목, 빨간색 X표시는 추가로 설치해야 할 항목, 흰색 느낌표는 꼭 설치 할 필요는 없는 항목을 뜻한다.



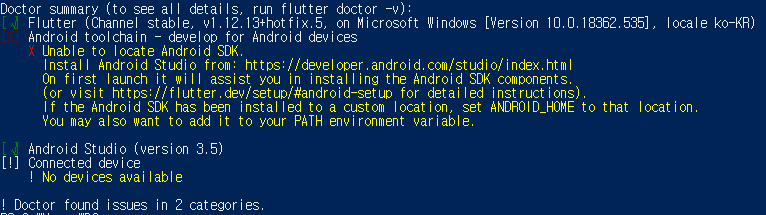
안드로이드 에뮬레이터를 위해서 안드로이드 스튜디오를 설치해야한다. 나 같은 경우엔 기존에 설치해둔 안드로이드 스튜디오가 있는데 인식이 안되는 상황이라 추가 작업이 필요했다.



안드로이드 스튜디오를 실행시키고 plugin을 추가해 줘야 한다. 워크스페이스 고르는 창에서 Configure > Plugins 에 들어가서 flutter와 dart를 검색해서 설치한다. 그리고 안드로이드 스튜디오는 종료한다. 만약 안드로이드 스튜디오를 설치했는데 인식이 안된다면 따라 해보길 바란다.

워크스페이스 선택이 아니라 워크스페이스로 들어갔다면, File-Settings-plugins에서 추가하면 된다. 설치를 안했기에, search in repositories 를 클릭해서 설치하면 된다. 다른 것 설치할 필요없이 flutter로만 된 것을 설치하면 된다.

다른 방법으로, 인식이 안될 때 파워쉘에서 flutter doctor --android-licenses 라는 명령어를 실행하면 해결 될 수 있다고 한다. (나의 경우엔 이걸로 해결되지 않음)

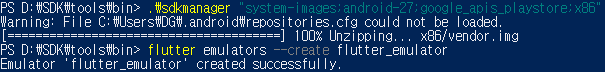


다시 PowerShell에서 flutter doctor을 실행해보면 안드로이드 스튜디오가 체크된 것이 보인다. 이러면 flutter를 사용하기 위한 기본 세팅은 마무리 된 것이다.



다시 command palette를 켜서 flutter: new project 누른 후, 원하는 프로젝트 이름을 정하고 프로젝트를 저장할 위치를 정하면 생성된다. 이때는 이전과 똑같이 flutter sdk를 찾을 수 없다고 뜰 수도 있는데 VS Code를 껐다 키면 정상작동 된다.

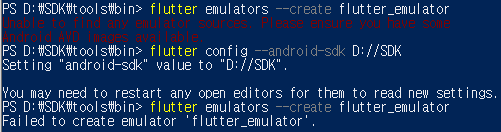
플러터 프로젝트를 실행하기 위해서는 플러터에서 사용할 수 있는 에뮬레이터를 만들어주는 작업이 필요하다. 윈도우에서는 안드로이드밖에 되지 않으며 MAC에서만 ios android 두개 다 가능하다. 따라서 나는 윈도우에서 작업함으로 안드로이드 에뮬레이터를 만드는 방법을 소개 하겠다.



안드로이드SDK위치\tools\bin> .\sdkmanager "system-images;android-27;google\_apis\_playstore;x86"

명령어를 이용해 에뮬레이터를 만들어주고, 아래 명령어를 이용해서 flutter에서 이용할 수 있는 에뮬레이터로 만들어 준다.

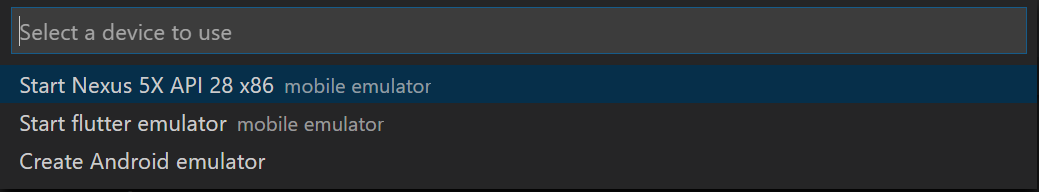
flutter emulators --create flutter\_emulator



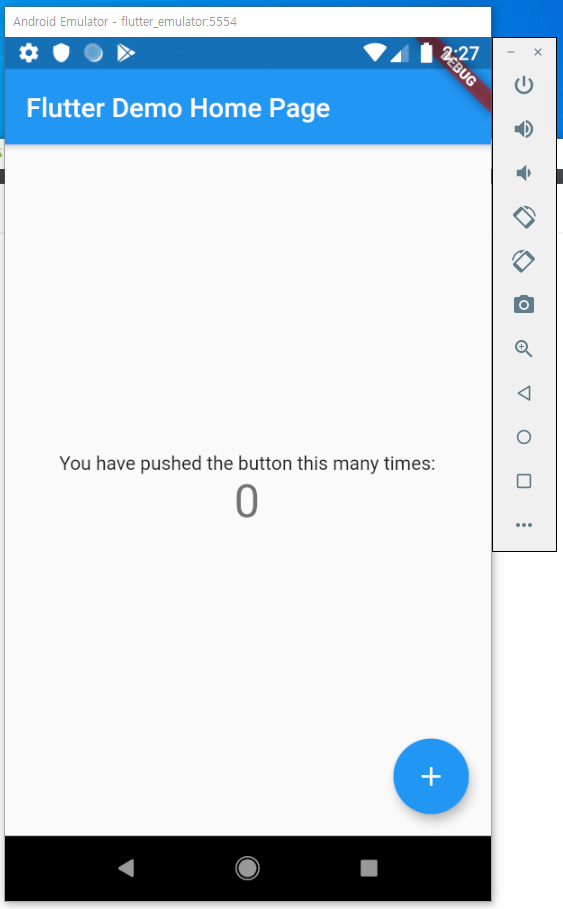
이 때, 에뮬레이터 소스를 찾을 수 없다는 에러가 뜬다면 플러터에서 SDK 위치를 찾지 못하는 것이다. ANDROID\_SDK\_ROOT 라는 환경변수로 sdk가 있는 위치를 설정해주거나 아래 명령어를 통해서 sdk위치를 명시해준다면 해당 오류는 더 이상 나타나지 않는다.

flutter config --android-sdk 안드로이드SDK위치

맨 마지막줄에서 보이는 생성에 실패했다는 문구는 에뮬레이터를 잘못생성해서 플러터에서 쓸 수 있는 에뮬이 없어서 그런 것이니 신경쓰지 않아도 된다.



F5를 누르면 에뮬레이터로 쓸 수 있는 목록이 나온다. Nexus 5X는 내가 안드로이드 앱 제작할 때 썼던 것이고, flutter emulator가 방금 만든 것이다. Flutter emulator를 클릭해서 실행해보자.



처음 실행을 하면 시간이 좀 걸린다. 하지만 첫 실행 이후는 hot-reload를 통해 빠르게 바뀐 부분만 수정해서 보여준다.

이로서 flutter를 사용하기 위한 사전 세팅이 모두 끝났다. 다음은 가계부 앱을 만드는 과정을 포스팅 하도록 하겠다.

<< flutter로 가계부 앱 만들기 >>