**Flask-GET**

**1. Flask 설치하기**

-> Flask의 경우 기본적으로 내장되어 있지 않기 때문에 따로 설치를 해줘야 한다.

-> terminal을 열어서 pip install flask라는 명령어를 실행한다.



Figure 1. pip install flask

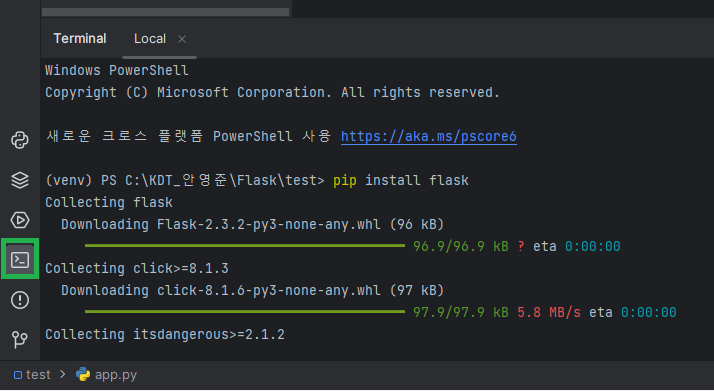


Figure 2. terminal에서 flask 다운로드

-> 녹색 버튼을 누르면 terminal이 열린다.

**※ 만약에 terminal에서 설치하기가 싫은 경우**

**1. File -> Settings로 들어간다.**

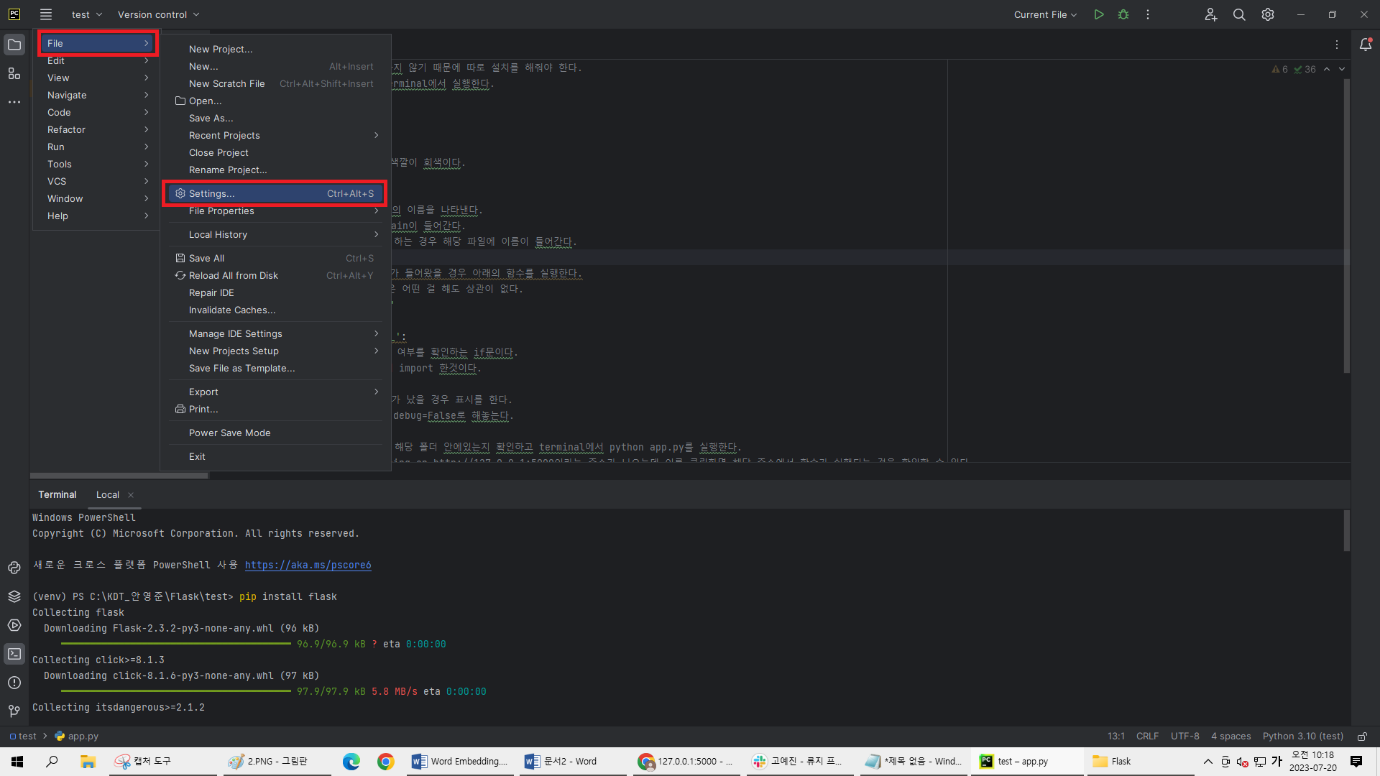


Figure 3. File -> Settings

**2. Project: test를 열고 Python Interpreter를 누른다.**

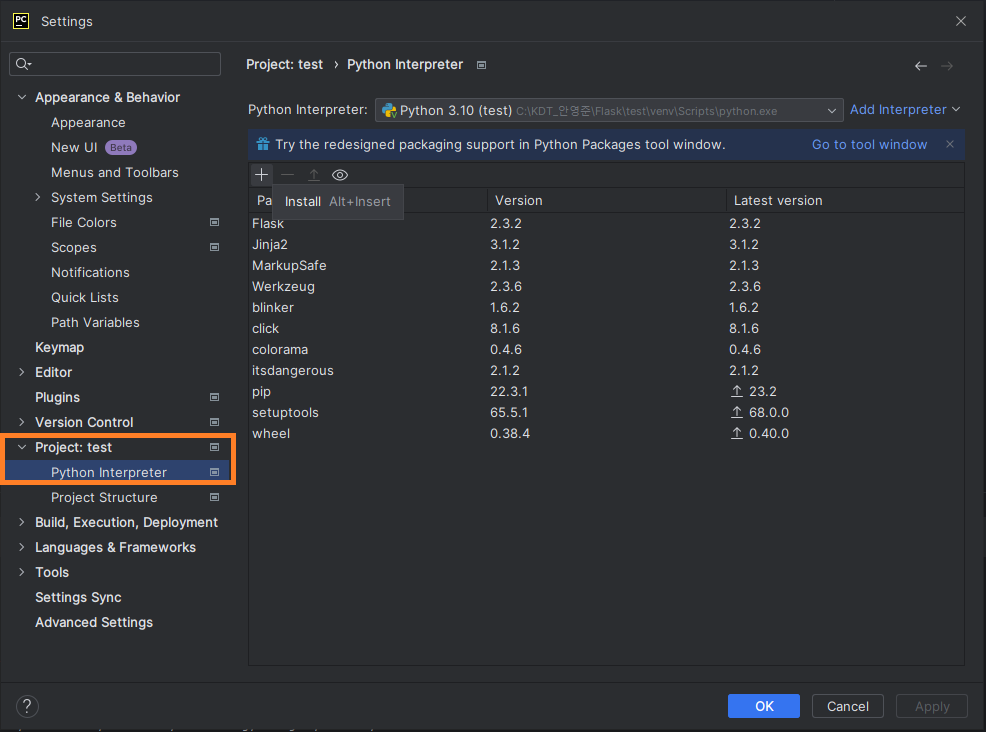


Figure 4. Python Interpreter 눌러보기

**3. 다운 받고 싶은 package의 이름을 입력한다.**

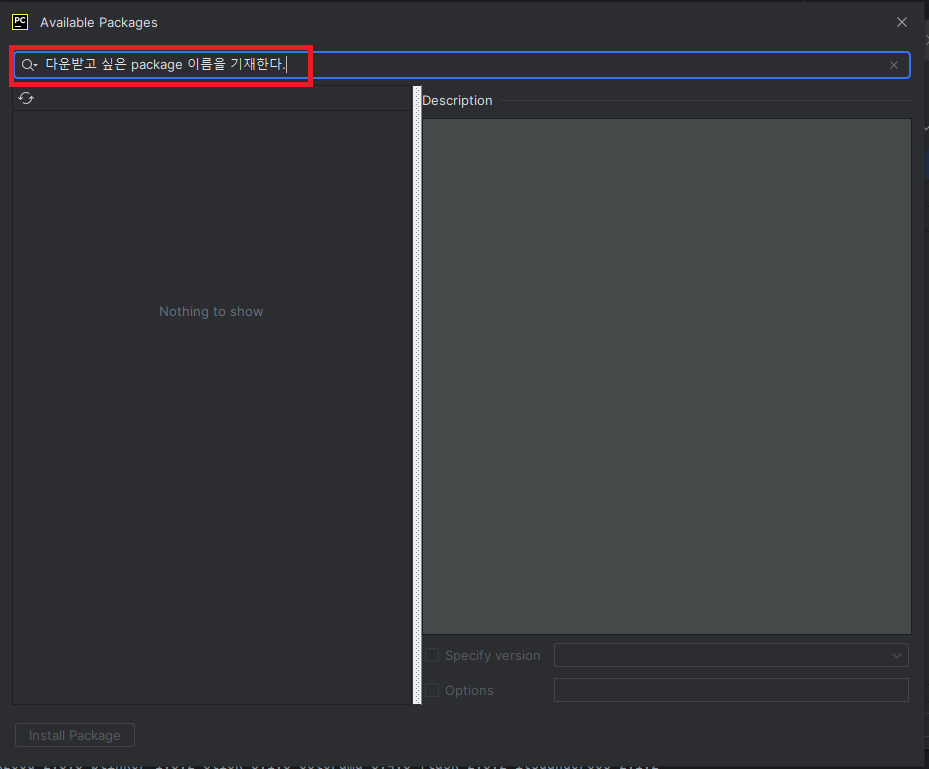


Figure 5. 다운 받고 싶은 Package의 이름을 입력한다.

-> 붉은색 상자에 있는 검색창에 다운받고 싶은 package의 이름을 입력하면 된다.

**2. Flask 사용해보기**

**2-1. app.py**



Figure 1. app.py 내부

-> Python에서 Flask 웹 애플리케이션을 생성하는 코드

-> Flask 클래스의 인스턴스를 생성한다.

-> 자기 자체가 실행이 되면 \_\_name\_\_에 \_\_main\_\_이 들어간다.

-> 다른 py 파일에서 import 하는 경우 해당 파일의 이름이 \_\_name\_\_ 대신에 들어간다.

-> router 부분(노드와 유사하다)

-> router가 들어왔을 경우(해당 url로 이동했을 경우) 아래의 함수를 실행한다.

-> 이때 함수의 이름은 상관이 없다.

-> return 부분이 결과창에 나타난다.

-> @app.route(‘/’)에 원래는 methods라는 parameter가 존재하는데 이 methods를 따로 설정하지 않으면 default 값인 GET이 된다.

-> if \_\_name\_\_ == ‘\_\_main\_\_’은 py 파일이 자체적으로 실행되있는지 여부를 묻는 if문이다.

-> if문이 true일 경우 app.py 파일을 실행한다.

-> debug=True로 하면 나중에 에러가 발생할 때 어떠한 종류의 에러가 발생했는지 알 수 있다.

=> 나중에 웹 혹은 앱을 배포할 경우 debug=False로 하면 된다.

**※※ 파일 실행**

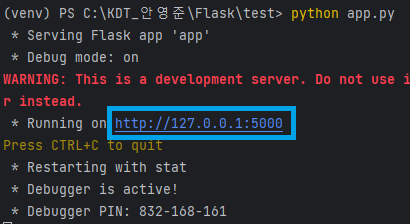


Figure 2. python app.py를 terminal에서 실행

-> 파일을 실행할 때는 파일이 해당 폴더 안에 있는지 확인하고 Figure2와 같이 terminal에서 python app.py라는 명령어를 입력하고 실행하면 된다.

-> 실행이 성공적일 경우 Running on http://127.0.0.1:5000이라는 주소가 나오는데 이를 클릭하면 해당 주소에서 함수가 실행되는 것을 확인할 수 있다.

-> 해당 주소를 누를 경우 Figure 3과 같이 해당 url로 이동을 하며 결과를 확인할 수 있다.

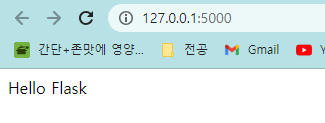


Figure 3. 결과 확인