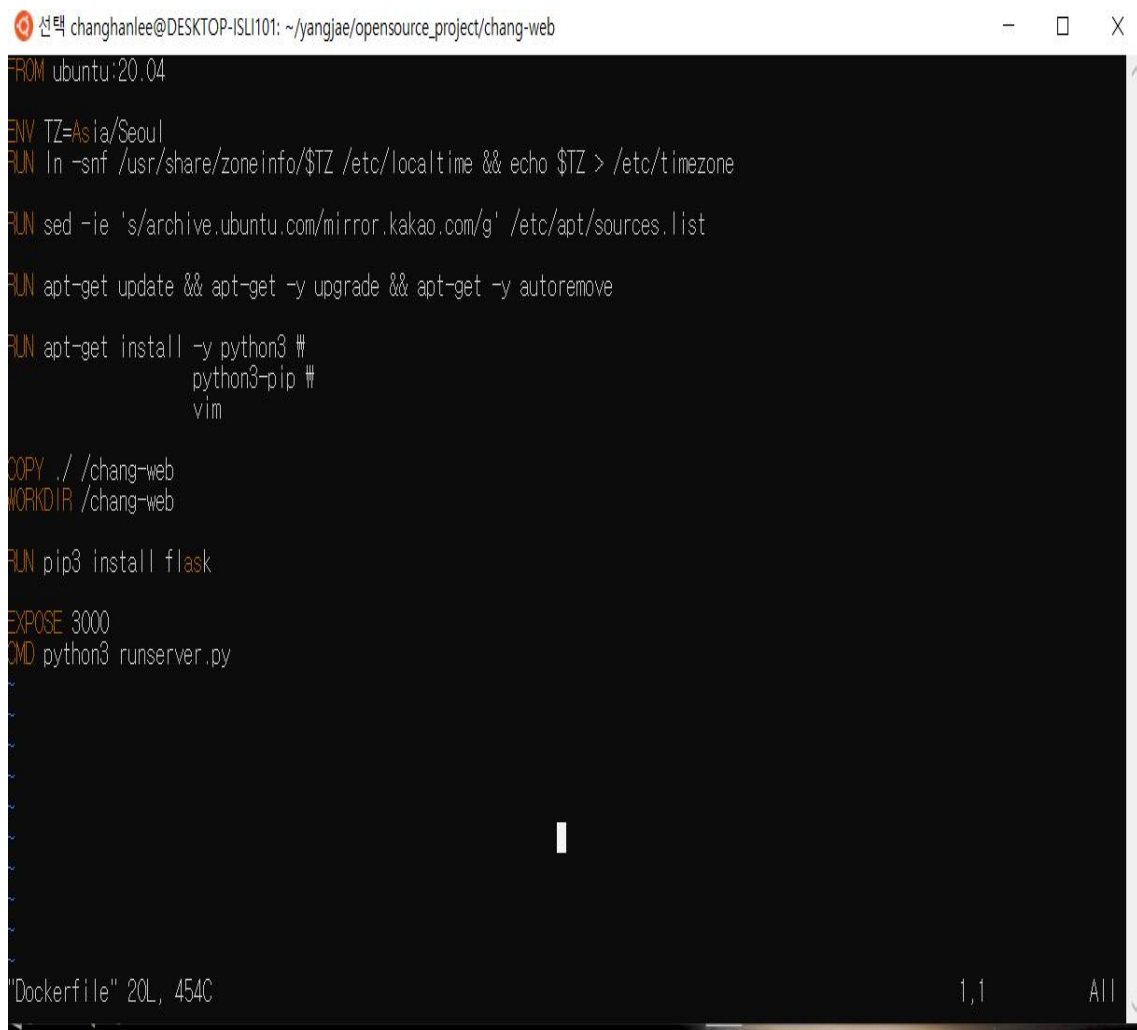


Docker 실습 내용 정리

1. Dockerfile 작성 및 이미지 빌드



```
FROM ubuntu:20.04

ENV TZ=Asia/Seoul
RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone
RUN sed -ie 's/archive.ubuntu.com/mirror.kakao.com/g' /etc/apt/sources.list
RUN apt-get update && apt-get -y upgrade && apt-get -y autoremove
RUN apt-get install -y python3 #
    python3-pip #
    vim #
COPY ./ /chang-web
WORKDIR /chang-web
RUN pip3 install flask
EXPOSE 3000
CMD python3 runserver.py
```

Dockerfile 텍스트 파일을 만들어 줍니다.

다음 그림은 위의 Dockerfile안에 들어있는 명령어들에 대한 설명입니다.

```
// Ubuntu 20.04 버전을 베이스 이미지 지정
FROM ubuntu:20.04

// Time zone을 서울로 수정
ENV TZ=Asia/Seoul
RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone

// apt 패키지 다운로드 서버를 카카오로 바꾸기
RUN sed -ie 's/archive.ubuntu.com/mirror.kakao.com/g' /etc/apt/sources.list

// apt 업데이트 및 업그레이드
RUN apt-get update && apt-get -y upgrade && apt-get -y autoremove

// 필요한 프로그램 설치
RUN apt-get install -y python3 \#
    python3-pip \#
    vim

// 현재 디렉토리를 도커 이미지 환경에 복사
COPY ./ /chang-web

// 현재 작업 디렉토리를 /chang-web으로 변경
WORKDIR /chang-web

// flask 파이썬 라이브러리 설치
RUN pip3 install flask

// 3000 포트를 개방하겠다 명시
EXPOSE 3000

// runserver.py을 실행 -> 웹서버 실행
CMD python3 runserver.py
```

-- INSERT --

60,1

Bot

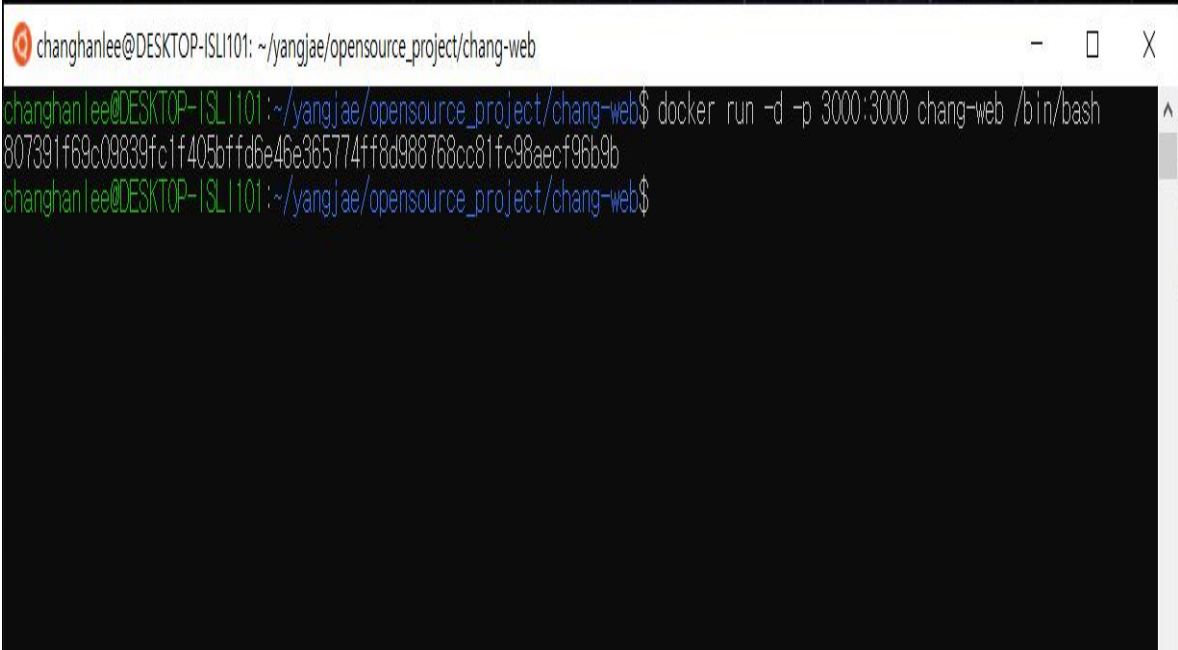
2. docker build -t chang-web .

```
changhanlee@DESKTOP-ISL1101:~/yangjae/opensource_project/chang-web$ docker build -t chang-web .
[+] Building 104.6s (14/14) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 498B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/ubuntu:20.04
=> [auth] library/ubuntu:pull token for registry-1.docker.io
=> [1/8] FROM docker.io/library/ubuntu:20.04@sha256:626ffe58f6e7566e00254b638eb7e0f3b11d4da9675088f4781a50ae288f3322
=> [internal] load build context
=> => transferring context: 16.58kB
=> CACHED [2/8] RUN ln -snf /usr/share/zoneinfo/Asia/Seoul /etc/localtime && echo Asia/Seoul > /etc/timezone
=> CACHED [3/8] RUN sed -ie 's/archive.ubuntu.com/mirror.kakao.com/g' /etc/apt/sources.list
=> CACHED [4/8] RUN apt-get update && apt-get -y upgrade && apt-get -y autoremove
=> [5/8] RUN apt-get install -y python3 python3-pip vim
=> [6/8] COPY ./ /chang-web
=> [7/8] WORKDIR /chang-web
=> [8/8] RUN pip3 install flask
=> exporting to image
=> => exporting layers
=> => writing image sha256:b9b67bad06ab6c18158ea8b6840e0c23722a99abc0974c3f641d179bdfd9781f
=> => naming to docker.io/library/chang-web

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
changhanlee@DESKTOP-ISL1101:~/yangjae/opensource_project/chang-web$
```

도커를 빌드합니다.

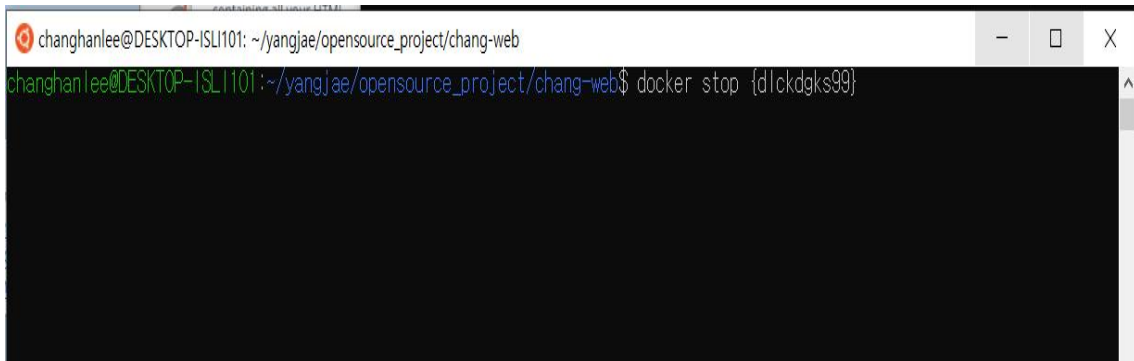
3. docker run -d -p 3000:3000 chang-web /bin/bash .



```
changhanlee@DESKTOP-ISL1101: ~/yangjae/opensource_project/chang-web
changhanlee@DESKTOP-ISL1101:~/yangjae/opensource_project/chang-web$ docker run -d -p 3000:3000 chang-web /bin/bash
807391f69c09839fc1f405bffd6e46e365774ff8d988768cc81fc98aef96b9b
changhanlee@DESKTOP-ISL1101:~/yangjae/opensource_project/chang-web$
```

도커를 실행시킵니다.

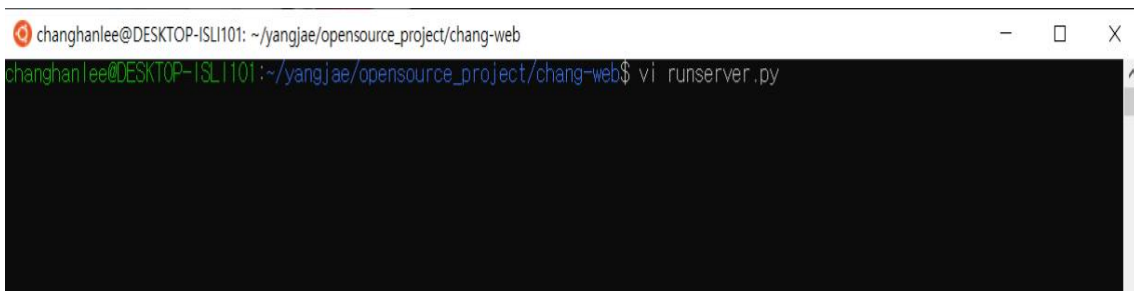
4. docker stop {dlckdgks99} .



```
changhanlee@DESKTOP-ISL1101: ~/yangjae/opensource_project/chang-web
changhanlee@DESKTOP-ISL1101:~/yangjae/opensource_project/chang-web$ docker stop {dlckdgks99}
```

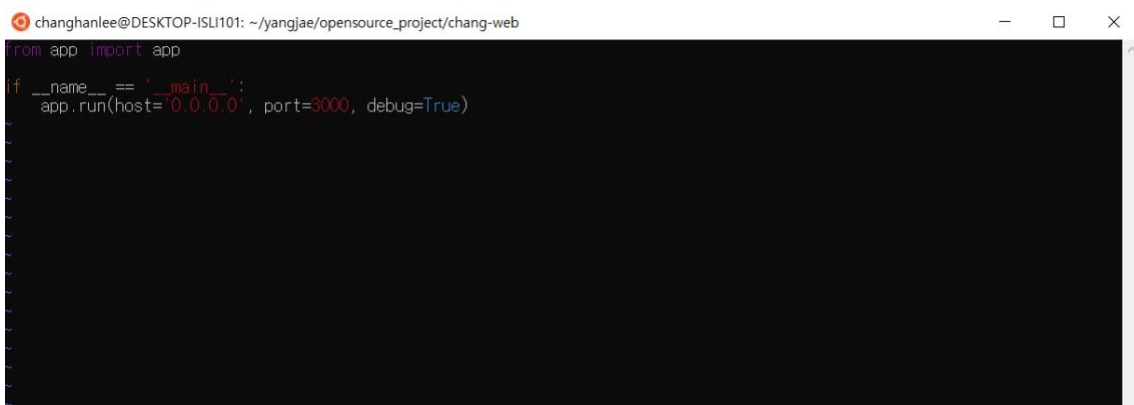
도커 실행을 멈춥니다.

5. vi runserver.py



```
changhanlee@DESKTOP-ISL1101: ~/yangjae/opensource_project/chang-web
changhanlee@DESKTOP-ISL1101:~/yangjae/opensource_project/chang-web$ vi runserver.py
```

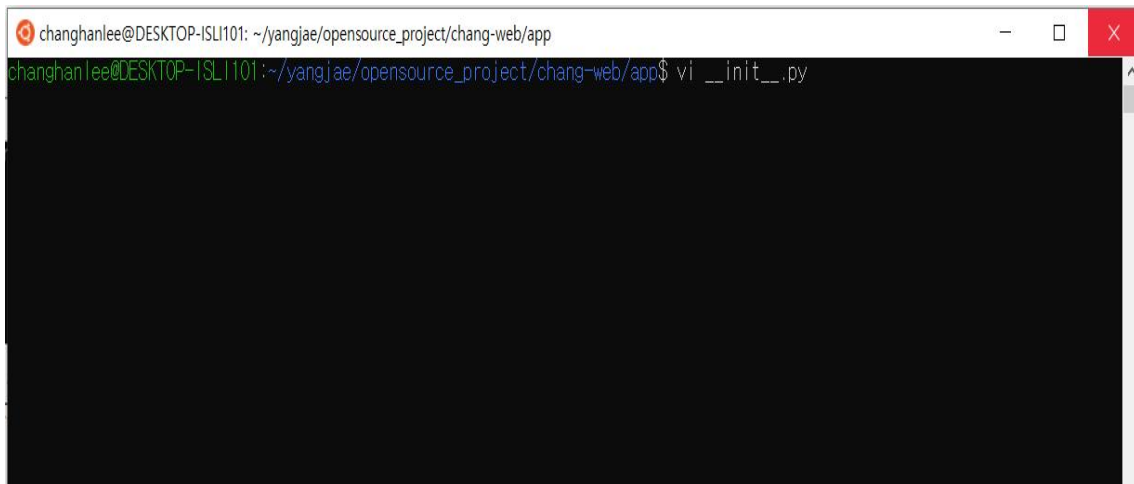
runserver.py 파일을 만든후 코드를 작성했습니다.



```
from app import app

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=3000, debug=True)
```

6. vi __init__.py



```
changanlee@DESKTOP-ISL1101: ~/yangjae/opensource_project/chang-web/app
changanlee@DESKTOP-ISL1101:~/yangjae/opensource_project/chang-web/app$ vi __init__.py
```

vi 명령어를 이용해 아래와 같은 파이썬 파일을 작성했습니다.



```
changanlee@DESKTOP-ISL1101: ~/yangjae/opensource_project/chang-web/app
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def door():
    return render_template('index.html')

~
~
~
```

