

## RUN

보통 이미지에 새로운 패키지를 설치하거나 명령어를 실행시킬 경우 사용됩니다.

```
FROM ubuntu:18.04
RUN apt-get update
RUN apt-get install -y python3 python3-pip wget git less neovim
RUN pip3 install pandas
```

`RUN` 명령은 실행할 때마다 레이어가 생성됩니다. 따라서 `RUN` 명령어 하나에 통합해준다면 보다 깔끔하게 레이어를 관리할 수 있습니다.

```
# example
FROM ubuntu:18.04
RUN apt-get update \
    && apt-get install -y python3 python3-pip wget git less neovim \
    && pip3 install pandas
```

## CMD

`docker run` 명령어 실행 시 실행 될 기본 명령어를 설정하거나, 아래에 있는 `ENTRYPOINT`의 기본 명령어 파라미터를 설정할 때 사용됩니다. `CMD` 명령어는 보통 컨테이너를 실행할 때 사용할 기본 명령어를 설정하는 것입니다.

- `CMD ["executable","param1","param2"]` (exec form, preferred)
- `CMD ["param1","param2"]` (sets additional default parameters for `ENTRYPOINT` in exec form)
- `CMD command param1 param2` (shell form)

```
FROM ubuntu:18.04
CMD echo "Hello"
```

위와 같이 작성하여 만들어진 이미지를 `docker run (옵션 x)` 명령어 실행 시 `CMD` 명령어가 실행되어 "Hello"를 출력하게 됩니다.

✱ 하지만, 두 번째 줄에 작성한 `CMD echo "Hello"`와 별개로 `docker run -it <image_name> echo "Hello world"` 명령어를 주게 되면, `dockerfile`에서 작성한 "Hello"는 무시되고 "hello world"를 출력하게 됩니다.

✱ `CMD`는 여러 번 `dockerfile`에 작성할 수 있지만, 가장 마지막에 작성된 `CMD`만이 실행(override)됩니다.

# ENTRYPOINT

docker run 명령어로 컨테이너를 생성 후 실행되는 명령어입니다.

- ENTRYPOINT ["executable", "param1", "param2"] (exec form, preferred)
- ENTRYPOINT command param1 param2 (shell form)



CMD 와의 차이를 쉽게 설명하자면, ENTRYPOINT 는 docker run 뒤에 명령어 작성과 무관하게 실행되는 명령어입니다.

```
FROM ubuntu:18.04
ENTRYPOINT ["/bin/echo", "Hi"]
CMD echo "Hello"
```

```
$ docker run -it <image_name>
Hi Hello
$ docker run -it <image_name> Mother
Hi Mother
```