

1. Docker

Docker Toolbox 설치 (10분) - <https://www.docker.com/products/docker-toolbox>

Docker 명령어 실습

| 명령어 | 설명 | 옵션 | 실습 명령어 |
|---------|---------------------|----|---|
| search | Docker Hub에서 이미지 검색 | | docker search helloworld (도커허브 사이트에서도 helloworld 검색해 보기) |
| pull | | | docker pull karthequian/helloworld:latest |
| images | | | docker images |
| run | 도커 컨테이너 생성 및 실행 | | docker run -p 80:80/tcp "karthequian/helloworld:latest" (웹브라우저에서 PORT번호 입력하여 실행되는지 확인) |
| ps | 컨테이너 목록 보기 | -a | docker ps |
| start | 정지된 컨테이너 재실행 | | docker start 컨테이너 ID (또는 names) |
| stop | 실행 중인 컨테이너 정지 | | docker stop hello-world |
| restart | 실행 중인 컨테이너 재부팅 | | docker restart hello-world |
| rm | 도커 컨테이너 삭제 | | docker rm 컨테이너 ID (또는 names) |
| rmi | 도커 이미지 삭제 | -f | docker rm 이미지 ID |

도커 명령어 : <https://docs.docker.com/engine/reference/commandline/cli/>

Docker 환경설정

인공지능 실습을 위한 도커 이미지 받기 :

```
$ docker pull continuumio/miniconda3
```

인공지능 실습을 위한 도커 컨테이너 실행하기 :

```
$ docker run -i -t continuumio/miniconda3
```

리눅스 명령어 실행해 보기

```
$ ls
```

파이썬 버전 확인하기

```
$ python --version
```

파이썬 실행해 보기

```
$ python3
```

```
$ 1+1
```

파이썬에서 나가기

```
$ quit()
```

도커 컨테이너 정지시키고 나가기

```
$ exit
```

인공지능 실습을 위한 도커 컨테이너 실행하기 :

```
$ docker run -i -t continuumio/miniconda3
```

파이썬 패키지 설치

```
$ pip install requests
```

나가기

```
$ exit
```

업데이트된 컨테이너를 새로운 도커 이미지로 저장

```
$ docker commit 컨테이너 ID mlearn:init
```

신규 생성된 도커 이미지 확인

```
$ docker images
```

인공지능 실습을 위한 도커 컨테이너 실행하기 :

```
$ docker run -i -t continuumio/miniconda3
```

내 폴더를 도커 컨테이너에 마운트(연결)하기

- Windows에 마운트용 폴더 생성 : C>사용자>내이름 에 'sample' 폴더 만들기
- Windows에 마운트용 파일 생성 : 'sample' 폴더에서 마우스 오른쪽클릭 > 새로만들기 > 텍스트 문서 > test.txt 파일 만들기
- 생성된 폴더를 도커에 마운트하기 :

```
$ docker run -i -t -v /c/Users/내이름/sample:/sample mlearn:init
```
- 도커 컨테이너에서 마운트 결과 확인하기 :
 - ```
$ ls -al
```
  - ```
$ cat /sample/test.txt
```

(참고) 도커 컨테이너 실행시킬 때마다 내 작업폴더 마운트해줘야 함

Windows의 경우 :

```
$ docker run -i -t -v /c/Users/내이름/sample:/sample mlearn:init
```

Mac의 경우 :

```
$ docker run -i -t -v /Users/daejinan/sample:/sample mlearn:init
```

파이썬 코드 작성하기

인공지능 실습을 위한 도커 컨테이너 실행하기 :

Windows의 경우 :

```
$ docker run -i -t -v /c/Users/내이름/sample:/sample mlearn:init
```

Mac의 경우 :

```
$ docker run -i -t -v /Users/daejinan/sample:/sample mlearn:init
```

내PC의 파이썬 폴더(마운트된 sample폴더)을 도커 컨테이너로 복사

```
$ docker cp /Users/내이름/sample 컨테이너ID:/usr/
```

(MAC 명령어) \$ docker cp /Users/daejinan/sample 6c1a8ff53f51:/usr/

docker commit 컨테이너ID 이름:태그

컨테이너 저장

docker run -i -t 이름:태그

컨테이너 실행

nano 텍스트 에디터 설치하기

```
$ apt-get install nano
```

리눅스에 내 폴더 만들기

```
$ mkdir /usr/code
```

파이썬 파일 만들기

```
$ nano week1.py
```

파이썬 코드 작성하기

```
print ("Sejong Python AI")
```

파이썬 코드 저장하고 나가기

Ctrl + O

엔터

Ctrl + X

엔터

도커 이미지 생성하기

목표 : 나만의 도커 이미지 만들기

```
$ cd ~
```

```
$ mkdir mydockerbuild
```

```
$ cd mydockerbuild
```

```
$ vi Dockerfile (붙여넣기: +p 엔터, 나가기: esc : w q 엔터, 글자 삽입하기: i )
```

```
FROM docker/whalesay:latest
RUN apt-get -y update && apt-get install -y fortunes
CMD /usr/games/fortune -a | cowsay
```

```
$ docker build -t docker-whale .
```

```
$ docker images
```

```
$ docker run docker-whale
```

(참고 사이트 : <https://github.com/oracle/cloud-native-devops-workshop/blob/master/containers/docker001/Participant-Guide.md#intro-to-basic-container-concepts> 13/59)

도커 허브 가입하기

<https://hub.docker.com/>

도커 이미지 배포하기

```
$ docker login
```

```
$ docker tag docker-whale:latest username/docker-whale:latest
```

```
$ docker push username/docker-whale:latest
```

```
$ docker rm short_id
```

```
$ docker rmi docker-whale
```

```
$ docker images
```

```
$ docker run username/docker-whale
```

모든 컨테이너 멈추고 삭제하기

```
$ docker stop $(docker ps -a -q)
```

```
$ docker rm $(docker ps -a -q)
```

도커 Compose를 이용해 홈페이지 만들기 (건너뛰어도 됨)

링크 : <https://github.com/oracle/cloud-native-devops-workshop/blob/master/containers/docker001/Participant-Guide.md#install-docker-compose>

깃허브 리포지터리 만들기

상단우측 + 클릭 > new repository > Repository name 입력 (2017-AI)

Readme.md 클릭 > 본문 작성 > Commit new file

Create new file 클릭 > 상단 파일제목을 Dockerfile로 입력 > 본문 작성 (아래 박스의 내용 복사해서 붙이기)

```
FROM docker/whalesay:latest  
RUN apt-get -y update && apt-get install -y fortunes  
CMD /usr/games/fortune -a | cowsay
```

> Commit new file 클릭 > 완료

도커 허브와 깃허브(GitHub) 연결하기 (도커 이미지 생성하고 컨테이너 실행하기 위해)

링크 : <https://github.com/oracle/cloud-native-devops-workshop/blob/master/containers/docker001/Participant-Guide.md#use-github-and-docker-hub-together-to-build-an-image-and-run-the-container>