

## 第 4 章课程辅助资料

为了便于大家更好地理解课程中的知识，学院教研组配套整理了课程所需的辅助资料，供大家参考使用。

### 1. 指令包

```
# 启动 Docker
```

```
# FROM 语法
```

```
touch Dockerfile
```

```
=====
```

```
FROM alpine:3.17
```

```
=====
```

```
docker build -t alpine-image-001 .
```

```
docker images
```

```
docker run alpine-image-001 cat /etc/os-release
```

```
docker run alpine-image-001 ls /
```

```
docker run alpine-image-001 df -h
```

```
docker run alpine-image-001 python --version
```

```
=====
```

```
FROM python:3.11
```

```
=====
```

```
docker build -t python-image-001 .
```

```
docker images
```

```
docker run python-image-001 python --version
```

```
=====
```

```
FROM openjdk:21
```

```
=====
```

```
docker build -t java-image-001 .
```

```
docker images
```

```
docker run java-image-001 java --version
```

=====

FROM alpine:3.17

=====

### # ENTRYPOINT 语法

docker container ls

docker images

docker run -d alpine-image-001 tail -f /dev/null

docker container ls

=====

FROM alpine:3.17

ENTRYPOINT ["tail","-f","/dev/null"]

=====

docker build -t alpine-image-002 .

docker images

docker run -d alpine-image-002

docker container ls

### # RUN 语法 - part 1

docker build -t alpine-image-003 .

docker images

docker run -d -p 8083:80 alpine-image-003

docker container ls

http://localhost:8083

docker exec -it [container\_id] /bin/sh

/ # apk --update add apache2

/ # rm -rf /var/cache/apk/\*

http://localhost:8083

/ # httpd

/ # ps

/ # kill [pid]

/ # httpd -D FOREGROUND

http://localhost:8083

/ # exit

docker container ls

=====

FROM alpine:3.17

```
RUN apk --update add apache2
RUN rm -rf /var/cache/apk/*
ENTRYPOINT ["httpd","-D","FOREGROUND"]
=====
```

```
docker build -t alpine-image-004 .
docker images
docker run -d -p 8084:80 alpine-image-004
docker container ls
http://localhost:8084
```

## # RUN 语法 - part 2

```
docker build -t alpine-image-005 --progress=plain .
```

```
=====
FROM alpine:3.17
RUN apk --update add apache2
RUN rm -rf /var/cache/apk/*
RUN cd /var/www/localhost/htdocs
RUN pwd
RUN ls
ENTRYPOINT ["httpd","-D","FOREGROUND"]
=====
```

```
docker build -t alpine-image-005 --progress=plain .
docker build -t alpine-image-005 --progress=plain --no-cache .
```

```
=====
FROM alpine:3.17
RUN apk --update add apache2
RUN rm -rf /var/cache/apk/*
RUN cd /var/www/localhost/htdocs && pwd
RUN cd /var/www/localhost/htdocs && ls
RUN cd /var/www/localhost/htdocs && cat index.html
ENTRYPOINT ["httpd","-D","FOREGROUND"]
=====
```

```
docker build -t alpine-image-005 --progress=plain --no-cache .
```

```
=====
FROM alpine:3.17
RUN apk --update add apache2
RUN rm -rf /var/cache/apk/*
```

```
RUN cd /var/www/localhost/htdocs && echo "<h3>I'm nobody, and I'm building my first
Docker Image 1</h3>" >> index.html
RUN cd /var/www/localhost/htdocs && echo "<h3>I'm nobody, and I'm building my first
Docker Image 2</h3>" >> index.html
RUN cd /var/www/localhost/htdocs && echo "<h3>I'm nobody, and I'm building my first
Docker Image 3</h3>" >> index.html
ENTRYPOINT ["httpd","-D","FOREGROUND"]
=====
```

```
docker build -t alpine-image-005 --progress=plain --no-cache .
docker images
docker run -d -p 8085:80 alpine-image-005
docker container ls
http://localhost:8085
```

### # WORKDIR 语法

```
=====
FROM alpine:3.17
WORKDIR /var/www/localhost/htdocs
RUN apk --update add apache2
RUN rm -rf /var/cache/apk/*
RUN echo "<h3>I'm nobody, and I'm building my first Docker Image 1</h3>" >> index.html
RUN echo "<h3>I'm nobody, and I'm building my first Docker Image 2</h3>" >> index.html
RUN echo "<h3>I'm nobody, and I'm building my first Docker Image 3</h3>" >> index.html
ENTRYPOINT ["httpd","-D","FOREGROUND"]
=====
```

```
docker build -t alpine-image-006 .
docker images
docker run -d -p 8086:80 alpine-image-006
docker container ls
http://localhost:8086
```

### # ARG 语法

```
=====
FROM alpine:3.17
ARG whoami
WORKDIR /var/www/localhost/htdocs
RUN apk --update add apache2
RUN rm -rf /var/cache/apk/*
RUN echo "<h3>I'm ${whoami}, and I'm building my first Docker Image 1</h3>" >>
```

```
index.html
RUN echo "<h3>I'm ${whoami}, and I'm building my first Docker Image 2<h3>" >>
index.html
RUN echo "<h3>I'm ${whoami}, and I'm building my first Docker Image 3<h3>" >>
index.html
ENTRYPOINT ["httpd","-D","FOREGROUND"]
=====
```

```
docker build -t alpine-image-007 --build-arg whoami=SamTsai .
docker images
docker run -d -p 8087:80 alpine-image-007
docker container ls
http://localhost:8087
```

## # ENV 语法

```
=====
FROM alpine:3.17
ARG whoami
ENV db_user=none
WORKDIR /var/www/localhost/htdocs
RUN apk --update add apache2
RUN rm -rf /var/cache/apk/*
RUN echo "<h3>I'm ${whoami}, and I'm building my first Docker Image 1<h3>" >>
index.html
RUN echo "<h3>I'm ${whoami}, and I'm building my first Docker Image 2<h3>" >>
index.html
RUN echo "<h3>I'm ${whoami}, and I'm building my first Docker Image 3<h3>" >>
index.html
ENTRYPOINT ["httpd","-D","FOREGROUND"]
=====
```

```
docker build -t alpine-image-008 --build-arg whoami=SamTsai .
docker images
docker run -d -p 8088:80 -e "db_user=admin001" alpine-image-008
docker container ls
docker exec -it [container_id] /bin/sh
/ # echo ${db_user}
/ # exit
```

```
docker stop [container_id]
docker rm [container_id]
```

```
docker run -d -p 8088:80 -e "db_user=admin002" alpine-image-008
docker container ls
docker exec -it [container_id] /bin/sh
/ # echo ${db_user}
/ # exit
```

## # COPY 语法

```
touch content.txt
===== - =====

<div>
<h3>My Book List</h3>
<ul>
  <li><a href="#">Docker 100</a></li>
  <li><a href="#">Podman 200</a></li>
  <li><a href="#">Kubernetes 300</a></li>
  <li><a href="#">Alibaba 400</a></li>
  <li><a href="#">AWS 500</a></li>
</ul>
</div>
===== - =====

=====
FROM alpine:3.17
ARG whoami
ENV db_user=none
WORKDIR /var/www/localhost/htdocs
RUN apk --update add apache2
RUN rm -rf /var/cache/apk/*
RUN echo "<h3>I'm ${whoami}, and I'm building my first Docker Image 1<h3>" >>
index.html
RUN echo "<h3>I'm ${whoami}, and I'm building my first Docker Image 2<h3>" >>
index.html
RUN echo "<h3>I'm ${whoami}, and I'm building my first Docker Image 3<h3>" >>
index.html
COPY ./content.txt ./
RUN ls
RUN cat ./content.txt >> index.html
ENTRYPOINT ["httpd","-D","FOREGROUND"]
=====
```

```
docker build -t alpine-image-009 --build-arg whoami=SamTsai --progress=plain .
docker images
docker run -d -p 8089:80 -e "db_user=admin003" alpine-image-009
```

docker container ls  
http://localhost:8089

=====

```
FROM alpine:3.17
ARG whoami
ENV db_user=none
WORKDIR /var/www/localhost/htdocs
RUN apk --update add apache2
RUN rm -rf /var/cache/apk/*
RUN echo "<h3>I'm ${whoami}, and I'm building my first Docker Image 1<h3>" >>
index.html
RUN echo "<h3>I'm ${whoami}, and I'm building my first Docker Image 2<h3>" >>
index.html
RUN echo "<h3>I'm ${whoami}, and I'm building my first Docker Image 3<h3>" >>
index.html
COPY ./content.txt ./
RUN cat ./content.txt >> index.html
ENTRYPOINT ["httpd","-D","FOREGROUND"]
=====
```

Done!