Intro to Docker - Cheat Sheet

```
$ run a command from your host shell
$ run a command from inside a running docker container
```

Installing Docker

Go Here : https://docs.docker.com/install/

2. Docker Daemon

```
$ docker ps
CONTAINER ID    IMAGE    COMMAND    CREATED    STATUS    PORTS    NAMES
```

Turning on docker daemon if it's not running:

OSX
Use the docker icon in the status bar.

Linux

```
$ sudo systemctl start docker
```

or

```
$ sudo service docker start
```

2. Docker Images

```
$ docker pull busybox
$ docker images
$ docker pull node:alpine
```

5.a Running Containers

```
$ docker run busybox
$ docker ps busybox
$ docker ps -a busybox
```

5.b Running a Container Interactively

```
$ docker run -it busybox
$ ls
```

5.c Container Instances

```
$ touch HELLO_DOCKER
$ ls
$ exit
$ docker run -it busybox
$ ls
$ exit
```

5.d Understand “CMD”

```
$ docker run -it busybox ls -lh
$ docker run -it busybox bash
```

5.e Managing Containers

```
$ docker ps
$ docker ps -a
$ docker rm [Container Name Here]
```

5.f Named Containers

```
$ docker run -it --name bb busybox
$ docker ps
$ docker rm -f bb
```

5.g Temporary Containers

```
$ docker run -it --rm busybox
$ exit
$ docker ps
```

6.a Expose a container port

```
$ docker run --name web -p 8080:80 nginx
```

6.b Run a webserver daemonized

Ctrl+c to kill a running process / container

```
$ docker run --name web -d -p 8080:80 nginx
$ docker ps
$ docker rm web
$ docker run --name web -d -p 8080:80 nginx
```

6.c Enter a running container

```
$ docker exec -it web bash
$ cat /usr/share/nginx/html/index.html
$ apt update
$ apt install nano
$ nano /usr/share/nginx/html/index.html
$ exit
```

6.d Kill a daemonized container

```
$ docker rm web
$ docker stop web
$ docker rm web
$ docker rm -f web
```

7.a Mount a local directory

```
$ mkdir -p ~/Learn-Docker/static-website
$ cd ~/Learn-Docker/static-website
$ $EDITOR index.html
```

7.b `index.html`

```
<h1>Hello Docker</h1>
```

7.c Mount a local directory

```
$ docker run --name web -d -p 8080:80 \
  -v $PWD:/usr/share/nginx/html/ nginx
```

7.d Run a node script

```
$ mkdir -p ~/Learn-Docker/node-script
$ cd ~/Learn-Docker/node-script
$ $EDITOR script.js
```

7.e `script.js`

```
console.log('Hello Docker');
```

7.f Run a node script

```
$ docker run --name node-script \
  -v $PWD:/srv node:alpine node /srv/script.js
```

8.a Run a node server

```
$ mkdir -p ~/Learn-Docker/node-api
$ cd ~/Learn-Docker/node-api
$ npm init
$ npm i -S fastify
$ $EDITOR index.js
```

8.b `index.js`

- Source : https://github.com/theRemix/node-api

8.c Run a node server

```
$ docker run -d --name node-api -p 3000:3000 \
  -v $PWD:/srv -e API_NAME=yourname node:alpine node /srv
$ docker logs node-api
$ docker logs -f node-api
$ docker inspect node-api
```

8.d Run multiple node servers

```
$ docker run -d --name node-api-2 -p 3000:3000 \
  -v $PWD:/srv -e API_NAME=yourname-2 node:alpine node /srv
$ docker run -d --name node-api-2 -p 3002:3000 \
  -v $PWD:/srv -e API_NAME=yourname-2 node:alpine node /srv
$ docker run -d --name node-api-3 -p 3003:3000 \
  -v $PWD:/srv -e API_NAME=yourname-3 node:alpine node /srv
```

9.a Create a Dockerfile

```
$ cd ~/Learn-Docker/node-api
$ $EDITOR Dockerfile
```

9.b Dockerfile

```
FROM node:alpine
EXPOSE 3000
ENV API_NAME default
WORKDIR /srv

COPY package*.json /srv/
COPY index.js /srv/

RUN npm install

CMD ["node", "."]
```

9.c Build an Image from Dockerfile

```
$ docker build -t yourname/node-api:0.0.1 .
$ docker run -d --name node-api-5 -p 3005:3000 \
  -e API_NAME=yourname-5 yourname/node-api:0.0.1
```