Exercise Worksheet

From the Course:
Understanding Docker Run, Dockerfile, Docker-Compose for Beginners

Running docker containers with a shared host file system (a.k.a. docker volumes)

Let's share some files from our host to our docker container. This is done with the "-v" flag and then "host-directory:client-directory" and is directly mapped into the container.

```
docker run --rm -v ${PWD}:/myvol ubuntu /bin/bash -c "ls -lha >
/myvol/myfile.txt"
```

- This command runs "ls -lha" and pipes the output to /myvol/myfile.txt
- /myvol is "mounted" to the current working directory on the host
- On the host you can observe a new file called "myfile.txt" with the output of "ls -lha"
- The container is then ended and removed because of "--rm"

From the docker hub I get an image that contains the rar tool. It's minimal, but it demonstrates what can be done with separated environments:

https://hub.docker.com/r/klutchell/rar/dockerfile

```
docker run --rm klutchell/rar
```

- This will download the image klutchell/rar
- Execute it
- In the Dockerfile you can see the "entrypoint" is already the "rar" executable
    - This means, upon execution of "docker run" it starts the rar process.
    - All you must do is work with it like you work with "rar" itself
- "a" command will add to an archive.
- If we mount the right directory, we can simply compress files from the command line:

```
docker run --rm -v ${PWD}:/files klutchell/rar a /files/myrar.rar
/files/myfile.txt
```

- It creates a new compressed rar file called "myrar.rar" inside the hosts working directory
- It's fully compliant to rar files

```
docker run --rm -v ${PWD}:/files -w /files klutchell/rar a myrar.rar
myfile.txt
```

- The "-w /files" flag sets the working directory to /files.
- It's like "cd /files" inside the container and we can skip the full path to the files we want to compress