simple docker commands

- `docker info`
  docker related system information
- `docker version`
  show installed version
- `docker images`
  show downloaded images
- `docker search <pattern>`
  search Hub for *pattern* matching image

docker options

- `docker run [OPTIONS]`
  run docker image

with: `[OPTIONS]` matching

`-v, --volume` *vh:vc*
: connect volume to container

`-p, --port` *ph:pc*
: forward ports to host machine

`-e, --env` *list*
: set container environment

`-d, --detach`
: run container in background

`-i, --interactive`
: interactive STDIN, (mostly used with `-t`)

`-t, --tty`
: run tty, (mostly used with `-i`)

`-u, --user` *user*
: connect with user

`-w, --workdir string`
: start container with workdir

manage images & container

- `docker stats`
  container statistics (cpu/mem etc.)
- `docker top <container>`
  display running processes of a container
- `docker ps`
  show all active containers
- `docker ps -a`
  show additional inactive containers
- `docker commit <container>`
  save container / create an image
- `docker rename <name/ID> <new_name>`
  rename image
- `docker history <image>`
  Command/Dockerfile History des Images
- `docker stop <container>`
  stop container
- `docker rm <container>`
  active deletion of a container
- `docker rmi <image>`
  active deletion of an image
- Ctrl P followed by Ctrl Q
  detach from active terminal
- `docker exec <container> command`
  send `command` into container
- `docker attach <container>`
  attach to containers terminal

system commands

- `docker system df`
  show used storage by docker
- `docker system prune`
  delete all stopped docker containers, networks etc.

Dockerfiles

Example Listing:

```
FROM ubuntu:18.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

- `FROM`
  create Layer from Image (e.g. Ubuntu:18.04)
- `COPY`
  copy data from current directory
- `RUN`
  builds your app with `make`
- `CMD`
  specifies what command to run within the container

docker build

- `docker build -t <imagename> - < Dockerfile`
  build image from *Dockerfile*

Examples

https://www.rocker-project.org/

start R without Rstudio:

```
~$ docker run -it --rm
-v /bph/puredata1/work/folder:/data
rocker/r-base
```

start R with Rstudio:

```
~$ docker run --rm -p 8787:8787
-v /bph/puredata1/work/folder:/data -e
PASSWORD=password rocker/rstudio
```

Forwarding GUI elements

Windows:

- Prerequisite: Installation of VcxSrv

```
PS: set -name DISPLAY -value <IP>:0.0
-e DISPLAY=$DISPLAY
```

Linux:

- Forward socket and Display to host

```
-e DISPLAY=$DISPLAY
-v /tmp/.X11-unix:/tmp/.X11-unix
```

Cheatsheet

2018 sven.kreienbrock@bph.ruhr-uni-bochum.de