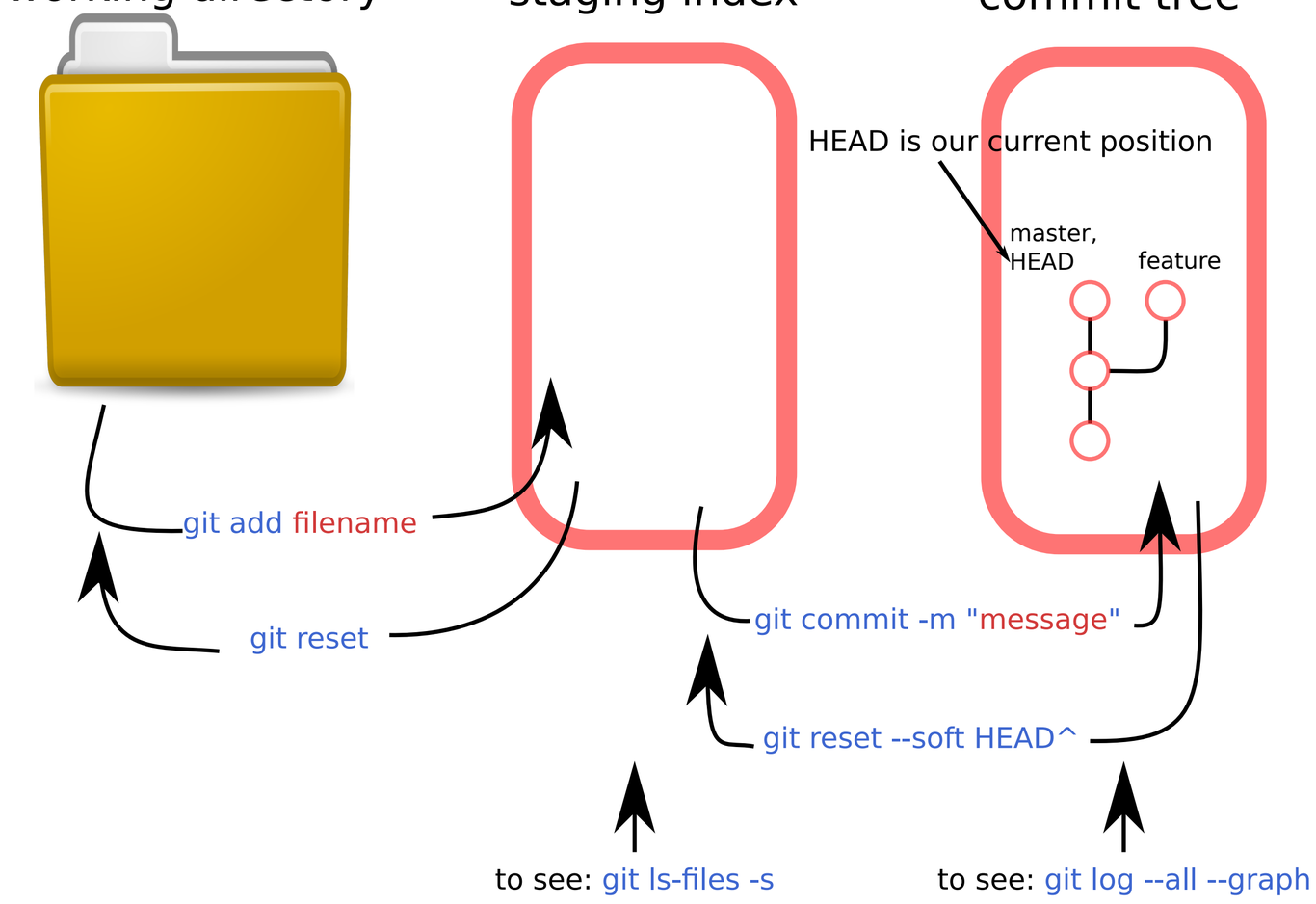
git workshop Cheat sheet

Initial config commands

git init executed inside a repository initiates an empty git repository

git config --global user.name your_name needs to be set the first time you use git to identify you

git config --global user.email your_mail needs to be set the first time you use git to identify you



to see the difference between the state of the three trees: git status
to see the difference since your last commit: git diff

Remote repository (for exemple on github)

git remote -v to see the remote names existing and their url

git push remote_name branch_name

git clone url to copy a remote repository on your computer

git fetch remote_name branch_name

(git pull = git fetch + git merge)

git remote add remote_name url adds a new remote pointing to a specific url

Branching commands

git branch list the branches and indicate which one you are in

git branch branch_name creates a new branch

git branch -d branch_name deletes a branch

git checkout branch_name switch to the specified branch

git merge branch_name merge the specified branch into our current one

"CherryPick" commands

git checkout commit-name -- /path/to/file picks the state of the file in this commit and set it in your working directory

Going back in time

git reset commit-name will change HEAD to commit-name, unstaging every changes since then

git reset --hard commit-name will change HEAD to commit-name, unstaging every changes since then and delete all local changes (DANGER ZONE)

git revert commit-name will create a commit that reverts the changes in the named commit.
This is the safest way to go to a previous commit

git checkout commit-name -b new_branch_name will create a branch starting at the commit and place you there. This is super safe as well

git stash will save your local changes since the last commit in a trash repo.

git stash pop will release the changes that were saved with git stash