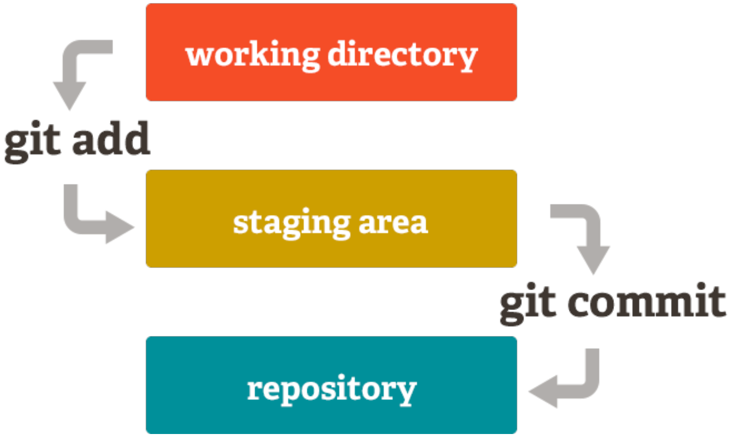
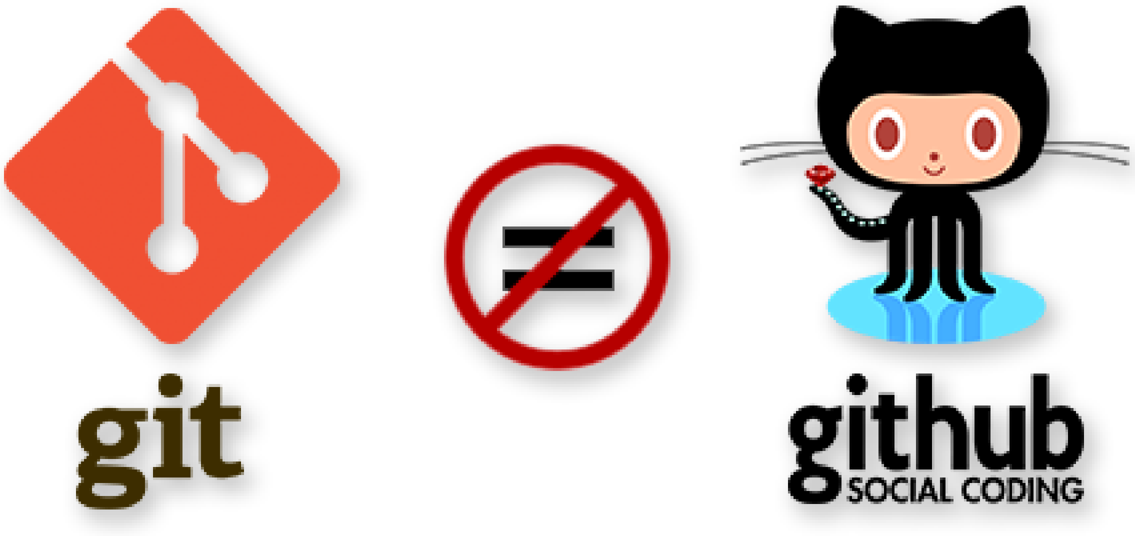
git

What is git (1/2)?

- Open Distributed Version Control System
  - Open – as a software, as a community (example: git book!)
  - Distributed – work locally, contribute globally
  - Version Control – helps you, without getting in the way



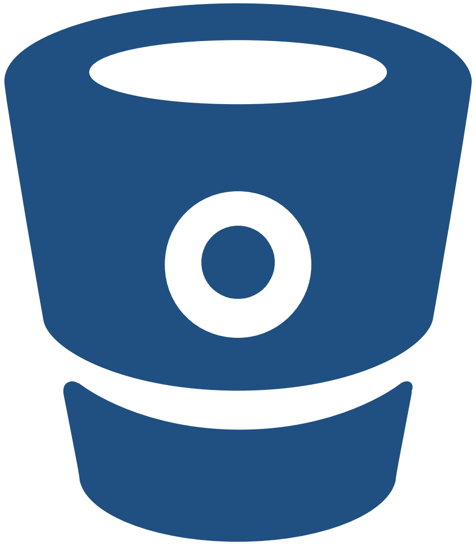
What is git (2/2)?



How do I get started with git?

- Get yourself the “official” cheat sheet
  - https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf
- Download and install git
  - https://git-scm.com/downloads
- Try it out :-)
- Read “the” git book
  - https://git-scm.com/book/en/v2
- Start collaborating online ... ;-)

Do you know the different online flavors?



Github

- Sweetest logo ;-)
- Used by FCCVienna
- Free repos can be seen by everybody
- You pay to get a “closed” repo
- https://github.com

Bitbucket

- Number 3 in category “logo”
- Free repos are closed to the world
- https://bitbucket.org/

Gitlab

- Coolest logo
- Perfect for free self hosting
- https://gitlab.com

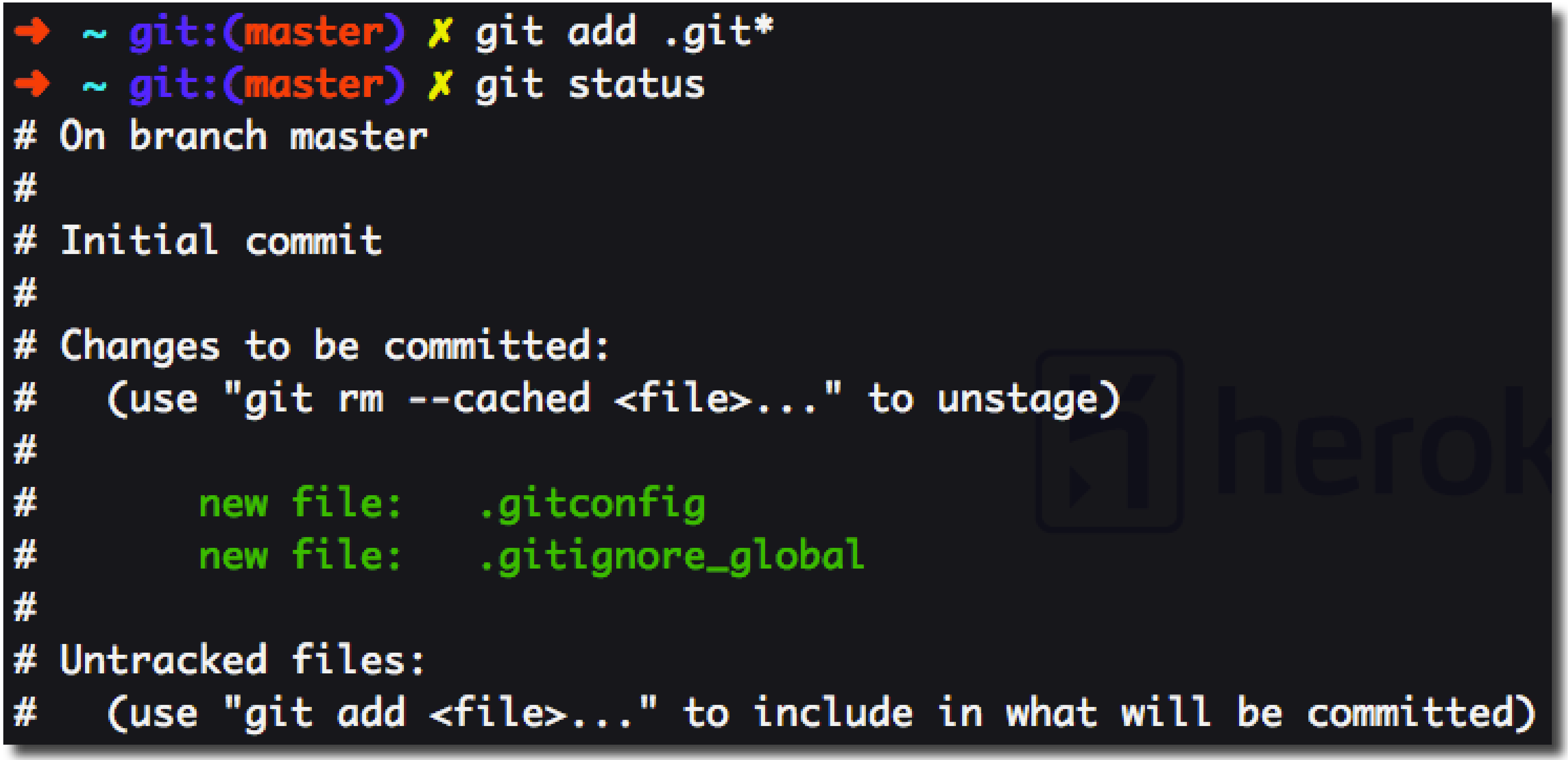
How to use a fork?

- Forks help you collaborate on github
- Forking – the “quick and dirty” explanation:
  - Fork the project you want to contrib to
  - Clone your fork locally
  - Add your awesome code
  - Sync everything with your fork on github
  - Next step: Pull request (with the original project)
- Great guide on github
  - https://help.github.com/articles/fork-a-repo/

How to “pull a request”?

- You have your forked code with your awesome new feature, but what now?
- Pull request – the “quick and dirty” explanation:
  - Go to your fork on github
  - Click on the pull request button
  - Leave a comment
  - Create the request and wait for the original owner to respond
- Great guide on github
  - https://help.github.com/articles/about-pull-requests/

Live Demo?!



```
➜  ~ git:(master) ✗ git add .git*
➜  ~ git:(master) ✗ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#	new file:   .gitconfig
#	new file:   .gitignore_global
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
```

Where to go from here?

- Create a wonderful .gitignore
- Personalize your installation with git config
  - Line endings (Win/Mac/Linux), pretty logs, and so much more
- Get a graphical interface (gitg, github, …)
- What to do when sh*t hits the fan?
  - What went (wrong)? git diff, log, status
  - How to fix it? git rebase, revert, reset, merge
  - The best git command ever: git blame ;-)
- Talk to others (online: gitter, irc, forums, stackoverflow, ...)