Aalto University
School of Science

Git / Github intro

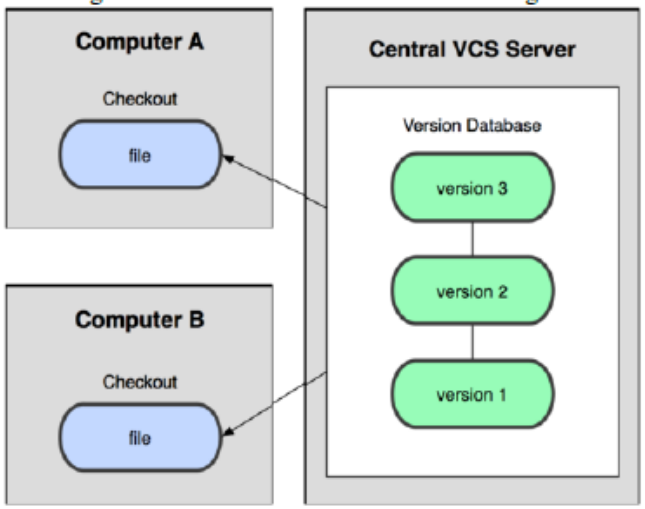
Kimmo Karhu
SCI
Department of Computer Science and Engineering

Motivation

- Do you work on several computers?
- Do you save files with dates in the name? i.e. you have a need to store different versions of the files
- If yes, you should consider using version control
- Version control is useful for any type of files
  - Source code
  - Design files (PSD, Autocad, etc)
  - Text files, articlesVer

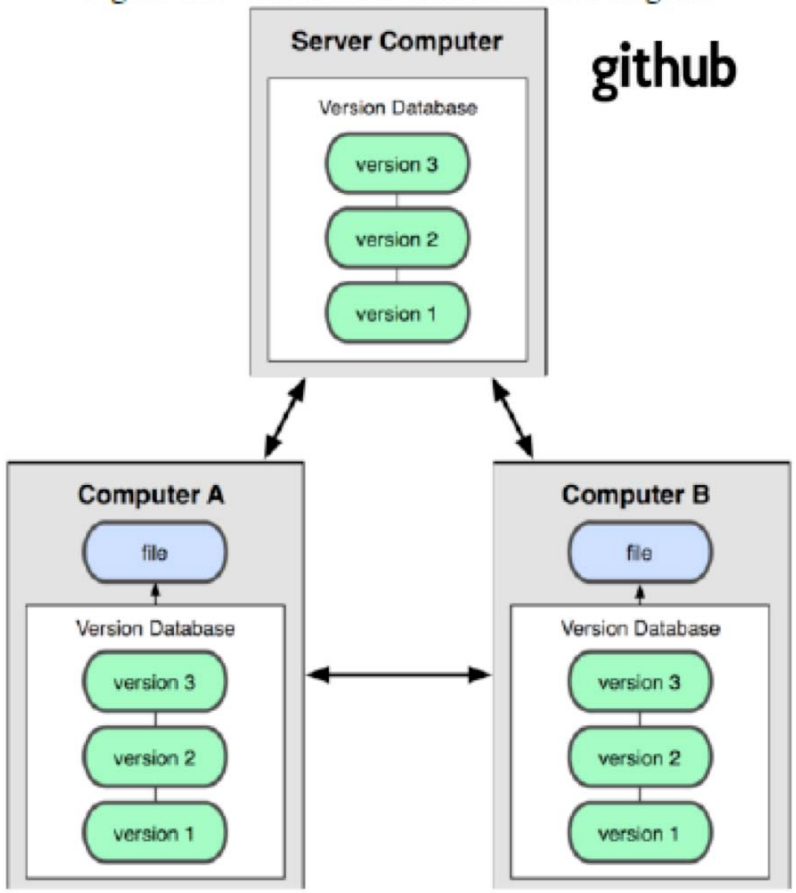
Traditional version control

Figure 1.2: Centralized version control diagram



Distributed version control such as GIT

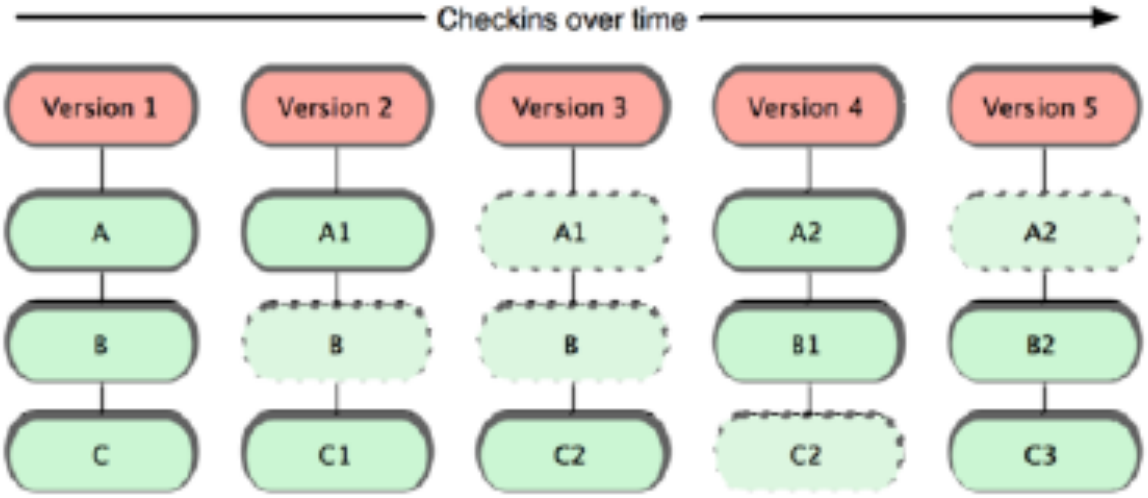
Figure 1.3: Distributed version control diagram



http://git-scm.com/book

How git stores the different versions of files

Figure 1.5: Git stores data as snapshots of the project over time.

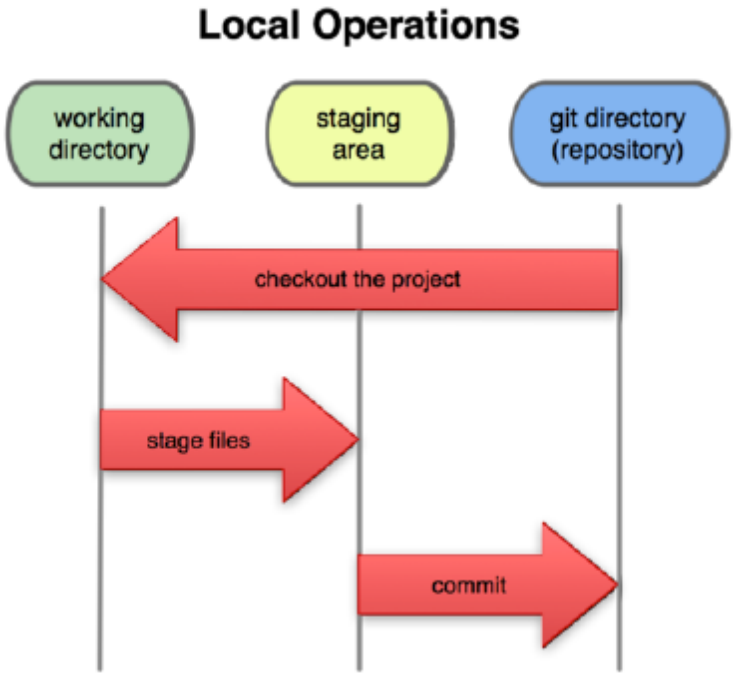


http://git-scm.com/book

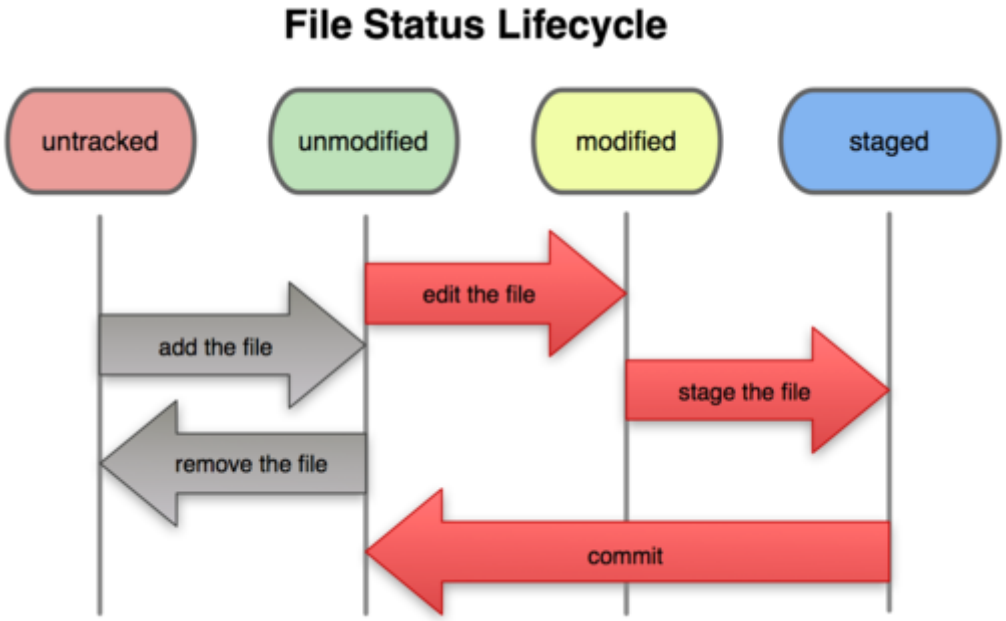
Git basic workflow

1. You modify files in your working directory.
2. You stage the files, adding snapshots of them to your staging area.
3. You do a commit, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory.

Figure 1.6: Working directory, staging area, and git directory

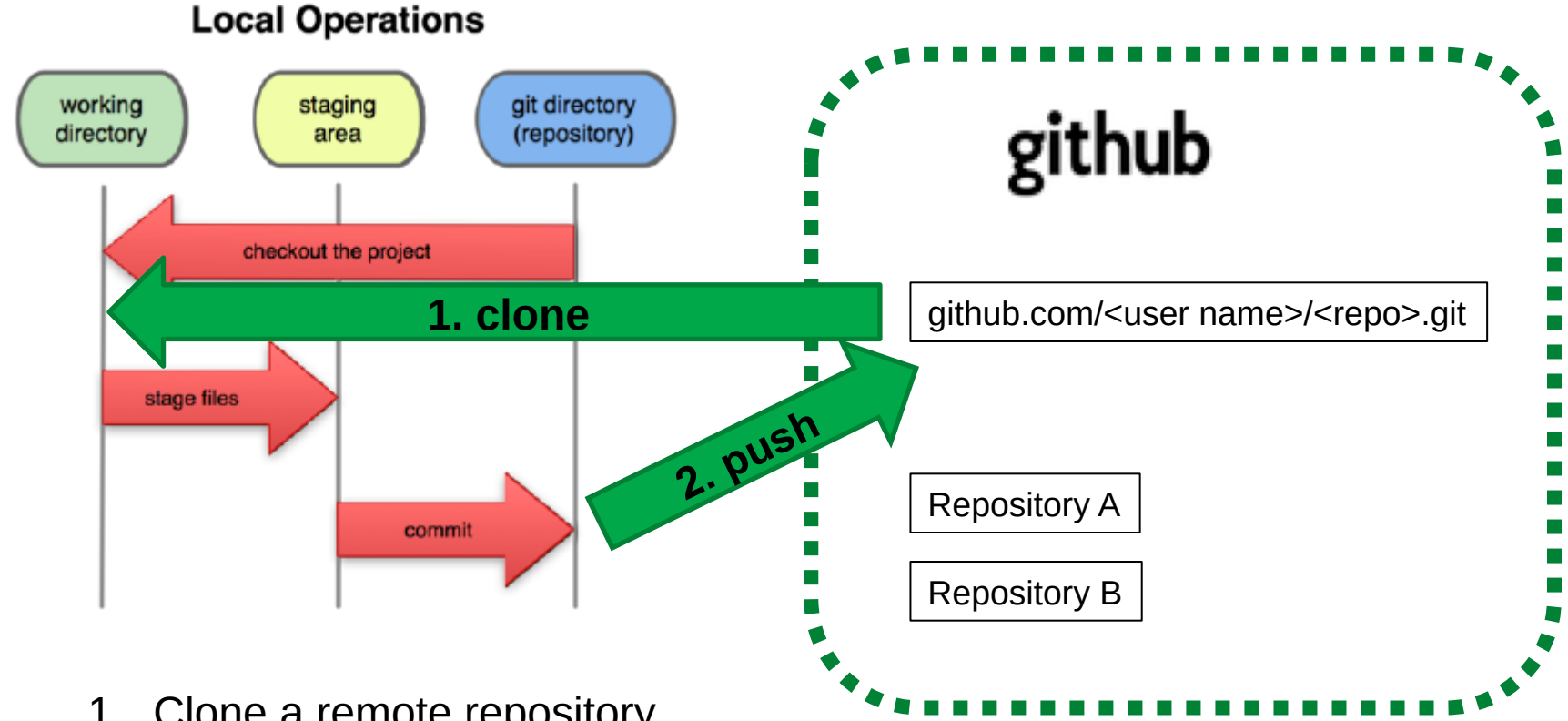


- Basically staging area is used to mark files that are to be committed next
- This allows you to keep changes that you don't want to yet commit in your working directory


File Status Lifecycle
untracked
unmodified
modified
staged
add the file
edit the file
stage the file
remove the file
commit

Working with remote repositories

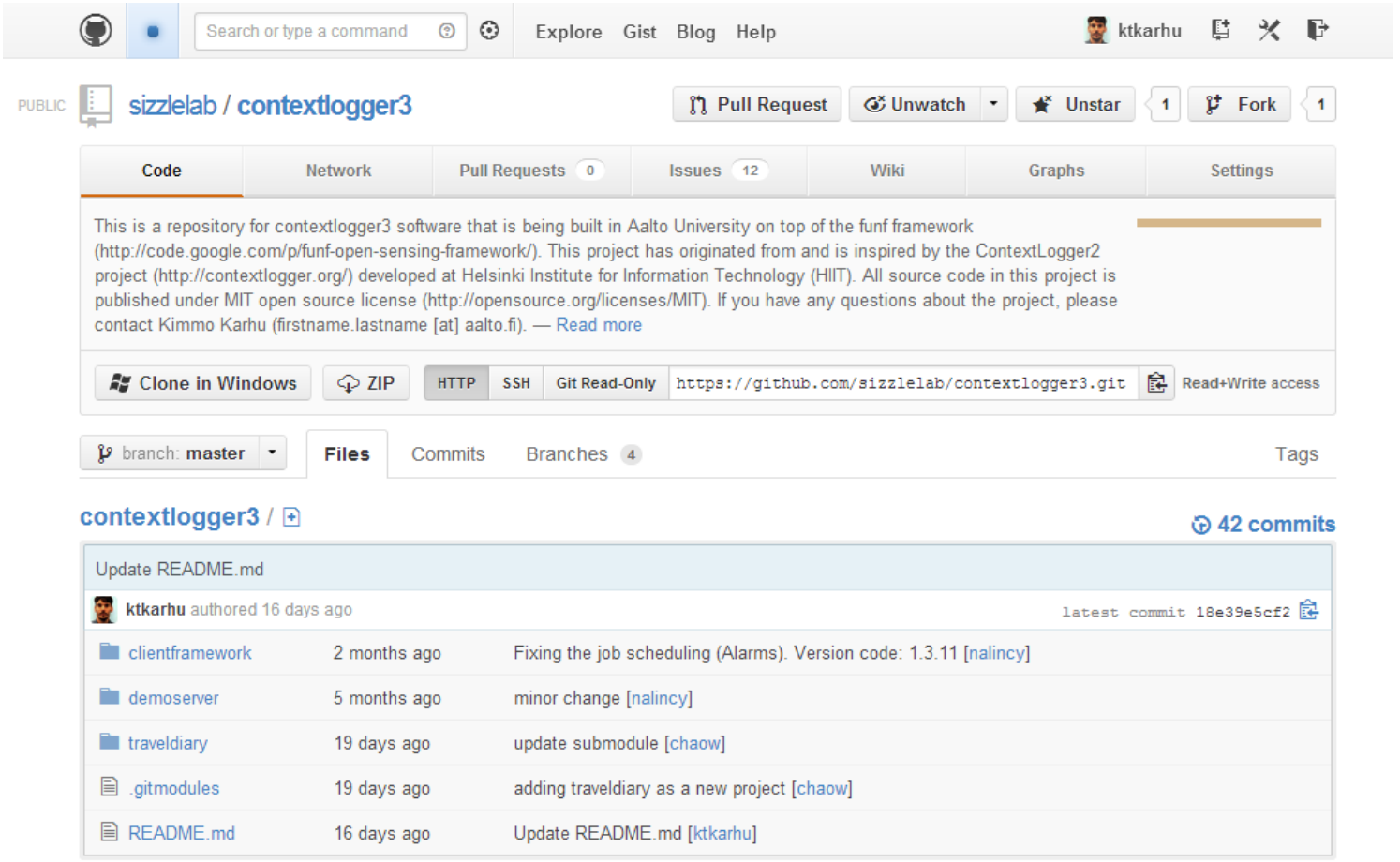
Figure 1.6: Working directory, staging area, and git directory



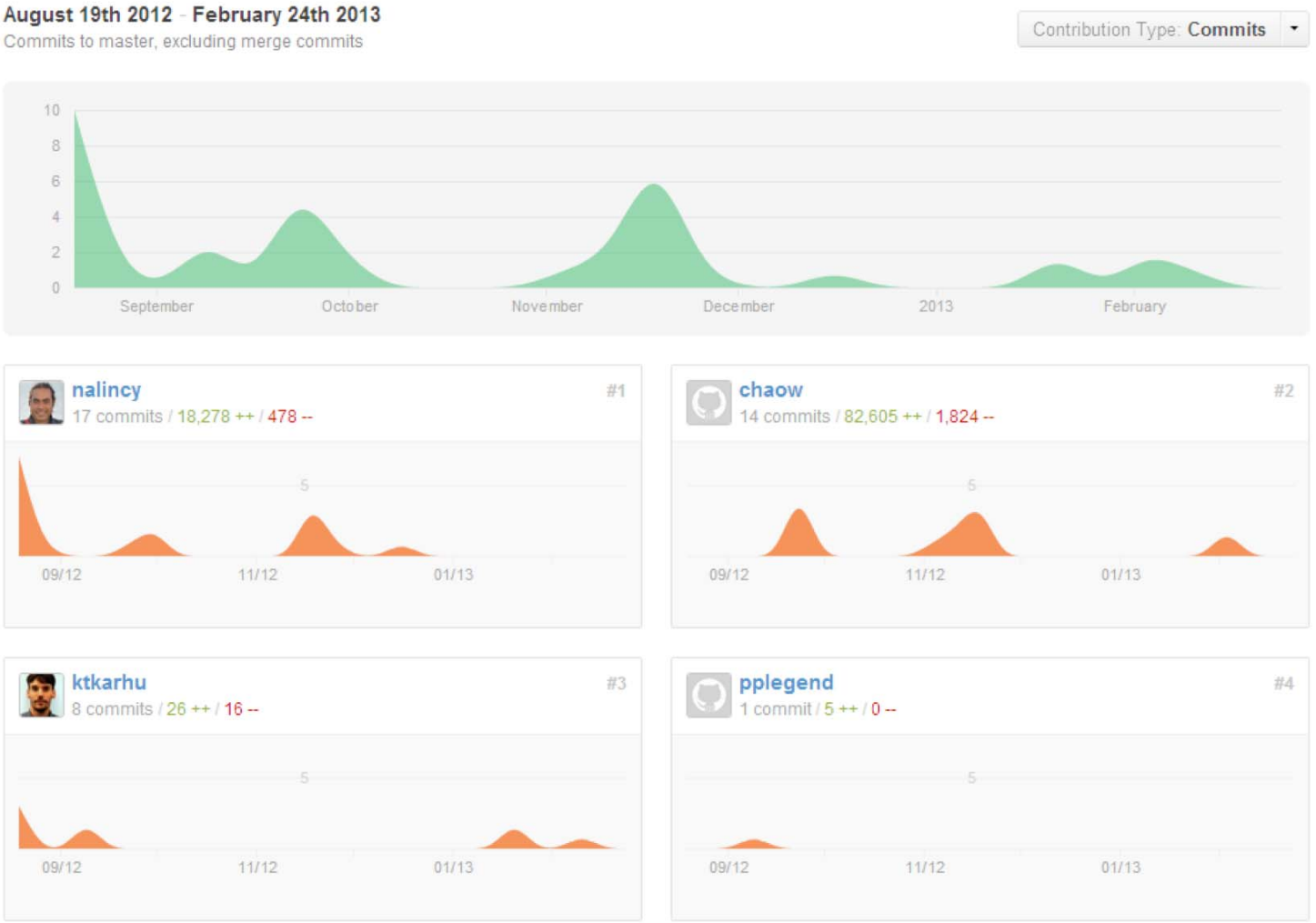
1. Clone a remote repository
2. Push your own changes

http://git-scm.com/book

Github


Search or type a command
Explore Gist Blog Help
ktkarhu
PUBLIC
sizzlelab / contextlogger3
Pull Request
Unwatch
Unstar
1
Fork
1
Code
Network
Pull Requests 0
Issues 12
Wiki
Graphs
Settings
This is a repository for contextlogger3 software that is being built in Aalto University on top of the funf framework (http://code.google.com/p/funf-open-sensing-framework/). This project has originated from and is inspired by the ContextLogger2 project (http://contextlogger.org/) developed at Helsinki Institute for Information Technology (HIIT). All source code in this project is published under MIT open source license (http://opensource.org/licenses/MIT). If you have any questions about the project, please contact Kimmo Karhu (firstname.lastname [at] aalto.fi). — Read more
Clone in Windows
ZIP
HTTP
SSH
Git Read-Only
https://github.com/sizzlelab/contextlogger3.git
Read+Write access
branch: master
Files
Commits
Branches 4
Tags
contextlogger3 /
42 commits
Update README.md
ktkarhu authored 16 days ago
latest commit 18e39e5cf2
clientframework 2 months ago Fixing the job scheduling (Alarms). Version code: 1.3.11 [nalincy]
demoserver 5 months ago minor change [nalincy]
traveldiary 19 days ago update submodule [chaow]
.gitmodules 19 days ago adding traveldiary as a new project [chaow]
README.md 16 days ago Update README.md [ktkarhu]

Follow the progress and contributions



1. **Install git**
   - Mac: http://code.google.com/p/git-osx-installer
   - Windows: http://code.google.com/p/msysgit
2. **Create a github account and add a new repository into your account**
3. **Open terminal (Utilities -> terminal)**
4. **Enter the following commands in the terminal replacing the text inside (and including) angle brackets (<,>) with your own data**

```
$ git config --global user.name "<Your Name>"
$ git config --global user.email <name@domain.com>
```
- *In terminal, you can use up arrow to go through previous commands and use them as a basis for a new command*

```
$ git config --global core.editor <path & executable of your favourite editor>
```
- *If you don't set up your own editor, git will use vi to input comments, very difficult to use*

- create a directory (using file manager) somewhere on your hard disk to store your repository

```
$ cd </path/your dir>
```
- *cd = Change directory*
- *In command prompt you can use tab to auto-fill file and directory names, enter first few characters and press tab*

```
$ git clone https://github.com/<your username>/<your repo>.git
```
- *You can also copy & paste this URL from your github repo's site*

```
$ cd <your repo>
```
- Create a text file into your repo's directory

```
$ git status
```
- *Use status command to check the status of your files (see slide 7)*

```
$ git add <file>
```
- *Use git add to collect new files and changes into your staging area, ready to be committed*
- *You can use * wild card in the file name or you can also use tab to auto-fill, highly recommended!*

```
$ git status
$ git commit
```
- *Use git commit to commit your changes (= a new version) to your local repository*
- *Your editor should pop-up. Enter your comment on top*
- *You can use git –m "<your comment>" to skip the editor pop-up*

```
$ git status
$ git push
```
- *Push your file and modifications from your local computer/repo to your github repository*

- Modify your file

```
$ git diff
```
- *You should see removed and added new lines for your change*

```
$ git add <file>
$ git commit
$ git log
```
- *You should see your all your commits and comments that you have entered*

```
$ git push
```

Got excited about Git/Github? Read these!

- Pro Git book: http://git-scm.com/book
  - Openly available
  - Especially chapters 1-3 are highly recommend, easy read
- Git Reference: http://git-scm.com/docs
- Github help: https://help.github.com/

- The GitHub Generation: Why We're All in Open Source Now
  - http://www.wired.com/opinion/2013/03/github/