

# A Very Very Short Introduction to Git

Tian Tian

January 6, 2017

# Outline

What & Why Git?

Git Commands

# Version-controlling: a comparison

Methods	Data Location	Pros	Cons
Manual Dating	L	SO EASY	What about merging?
Dropbox	L/R	Easy synchronizing	Only 1-way fallback
SVN	L/R	True VCS	Centralized
Git	L/R	Decentralized	Merging is verbose

## Basic ideas of git

- Cache snapshots of file are stored, not the difference
- Local operations, mirroring remote source
  - A central server is actually **not** necessary
- Can revert to any state

## States in git

- Not staged: changes are not recorded
- Staged (not committed): git knows your changes, but you haven't decided if you need to commit.
  - Normally it's easier to just stage and commit your changes
  - May be useful if you are doing a long fix
  - Unstage your changes does not revert the file.
- Committed: the changes are permanently recorded. You can either push your changes or continue working.
- **Anything you committed is safe!** (but the process can mess up)

## Helpful utilities for git

- Command line (**of course**)
- The github client (good for visual branching)
- Magit-mode for emacs
- vim-fugitive for vim

# Initialize a clean repository

*#Initialize a clean git repo*

```
git init
```

```
git status
```

## Results:

```
Reinitialized existing Git repository in  
/Users/tiantian/polybox/Studies_ETH/Computation-Seminar/.git/  
On branch master
```

```
Initial commit
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
#Tian-git.org#  
.#Tian-git.org  
.DS_Store  
Tian-git.dvi  
Tian-git.org  
Tian-git.org~  
Tian-git.pdf  
Tian-git.tex  
Tian-git.tex~  
_minted-Tian-git/
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

# Clone from an existing remote repository

```
# Clone from a remote repo to local dir
```

```
git clone https://github.com/lovaulonze/.matplotlib.git some
ls -al some
```

## Results

```
total 56
drwxr-xr-x  6 tiantian  staff   204  1  6 10:59 .
drwxr-xr-x 14 tiantian  staff   476  1  6 10:59 ..
drwxr-xr-x 12 tiantian  staff   408  1  6 10:59 .git
-rw-r--r--  1 tiantian  staff    61  1  6 10:59 .gitignore
-rw-r--r--  1 tiantian  staff 24528  1  6 10:59 matplotlibrc
drwxr-xr-x  4 tiantian  staff   136  1  6 10:59 stylelib
```



# Remember to use the .gitignore file

*# Git status w/o .gitignore rules*

git status

## Results

On branch master

Initial commit

Untracked files:

(use "git add <file>..." to include in what will be committed)

```
./#Tian-git.org
.DS_Store
Tian-git.dvi
Tian-git.org
Tian-git.org~
Tian-git.pdf
Tian-git.tex
Tian-git.tex~
_minted-Tian-git/
some/
```

nothing added to commit but untracked files present (use "git add" to track)

## .gitignore (II)

```
# Use .gitignore rules
```

```
cat .gitignore
```

```
echo ""
```

```
echo ""
```

```
git status
```

```
***
```

```
*~
```

```
*.tex
```

```
_minted*
```

```
some/
```

```
*.dvi
```

On branch master

Initial commit

Untracked files:

(use "git add <file>..." to include in what will be committed)

.DS\_Store

.gitignore

Tian-git.org

Tian-git.pdf

nothing added to commit but untracked files present (use "git add" to track)

## Stage the untracked files:

*# Add some file to the staged area*

*# But we leave out some*

```
git add Tian-* .gitignore
```

```
git status -s
```

```
A .gitignore
```

```
A Tian-git.org
```

```
A Tian-git.pdf
```

```
?? .DS_Store
```