

How to git out in case of fire

In case of fire



1. `git commit`



2. `git push`



3. `leave building`

Git Tips

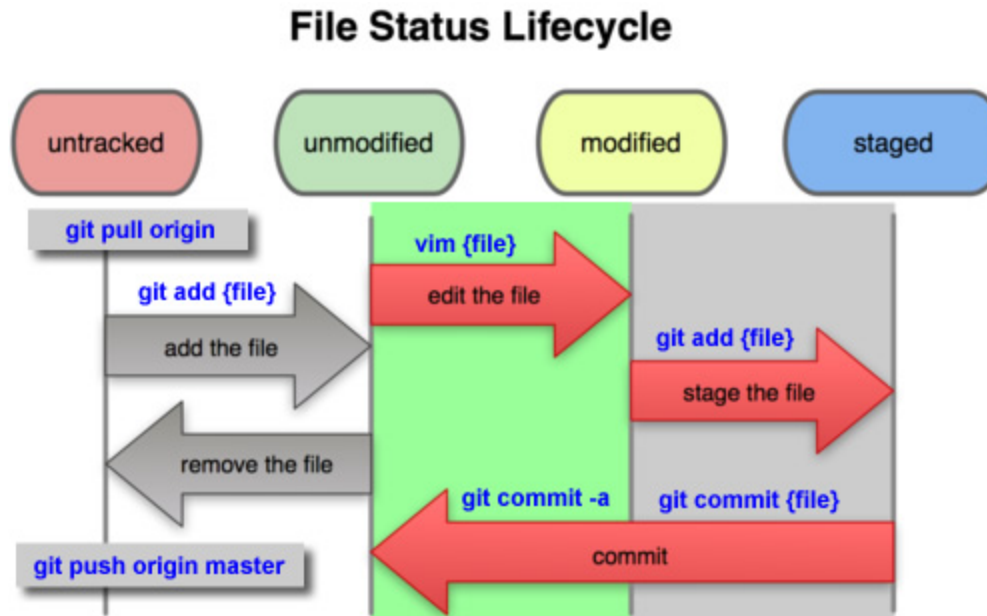
1. Rename repo when clone

```
git clone git://web.com/repo.git my-repo
```

2. use hyphens when name a git repo

```
git clone git://github.com/username/my-cool-project.git
```

Git File Lifecycle



Git File Lifecycle Demo

used commands:

git status, git init, git add, git rm, git commit, git log

```
~ mkdir git-example
~ cd git-example
~/git-example ls
~/git-example git init
Initialized empty Git repository in /home/yunfeng/git-example/.git/
~/git-example (master ✓) vim README
~/git-example (master ✗)* git status
On branch master

Initial commit

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

        README

nothing added to commit but untracked files present (use "git add" to track)
~/git-example (master ✗)* git add README
~/git-example (master ✗)+ git status
```



Git protocol

1. local protocol

```
git clone /data2/my-repo.git  
git clone file:///data2/my-repo.git  
git remote add local-project /data2/my-repo.git
```

2. SSH

```
git clone git@github.com:username/my-repo.git
```

3. Git protocol

```
git clone git://github.com/username/my-repo.git
```

4. HTTP protocol

```
git clone https://github.com/username/my-repo.git
```

git bitsect

find, by binary search, change that introduced a bug

```
git bitsect start  
git bitsect bad  
git bitsect reset  
git bitsect start bad-point good-point  
git bitsect run test-error.sh
```

git blame

show what revision and author last modified each line of a file

```
git blame my-file.txt  
git blame -L 10,20 my-file.txt  
git blame -C my-file.txt
```

git submodule

initialize, update, or inspect submodules

```
git submodule add git://github.com/user/repo.git my-sub-repo
git clone git://github.com/username/repo-with-submodule.git
git submodule init
git submodule update
git clone --recurvise git://some.com/user/repo.git
```


Improve your git skill

```
git mv
git rm -f
git commit -a
git diff --cached
git diff --staged
git log -p
git log -2
git log --stat
git log --pretty=oneline|short|full|fuller
git log --pretty=format:"%h - %an, %ar : %s"
git log --graph
git log --since=2.weeks|2016-10-01|"two years ago"
git log --author="Yangqing Jia"|"jiayq84@gmail.com"
git log --grep="bug"
git log --author="Yangqing Jia" --grep="bug" --all-match
git log -- README
git log -- src/caffe/layers/
git reset HEAD my-file
git checkout -- my-file
git remote -v
git remote show origin
git remote add tom-origin git@github.com:/tom/caffe
```

Improve your git skill

```
git fetch tom-origin
git remote rename tom-origin tom-caffe
git remote rm tom-caffe
git tag -l "v0.9*"
git tag -a v0.001 -m "my first milestone"
git show v0.001
git tag -v v0.001
git tag -a v0.002 -m "missed milestone" d195e60
git push origin v0.002
git push origin --tags
git config --global alias.co checkout
git config --global --unset alias.co
git branch -v
git branch --merge
git branch --no-merged
git branch -D unmerged-branch
git checkout -b new-brach origin/new-branch
git merge origin/another-branch
git rebase --onto master b2 b1
git show d195e60
git log --abbrev-commit
git rev-parse origin/master
```

Improve your git skill

```
git reflog
git show HEAD@{2}
git show master@{2015-10-12}
git show HEAD~~~
git show HEAD~3
git show HEAD^3
git show HEAD^^^
git log master..curr-branch
git log origin/master..HEAD
git log origin/master..
git log refA refB ^refC
git log refA refB --not refC
git add -i
git cherry-pick d195e60
```

quiz: how to rewrite commit message?

```
git commit --amend
```

quiz: How to change past commit to include a missed file?

```
git add missed-file  
git commit --amend --no-edit
```

quiz: how to fix a bug with when you are developing a new feature?

```
# assuming you are on new-feature branch
git add new-feature-file
git commit -m "save new feature file"
git checkout master
git checkout -b bugfix
#fix bug here...
git add fixed-file
git commit -m "fix bug#23: wrong idx"
git checkout master
git merge bugfix
git branch -d bugfix #del bug branch since all works merged
git checkout new-feature # work on new-feature branch
```

quiz: How to delete remote branch?

```
git push origin :branch-name
```

quiz: How to save only one part of change of a file?

```
1. git add -i
```

or

```
2. git add -p|--patch
```

Then enter a interactive interface, choose your code to commit

Learn git by playing games

1. [githug](#)
2. [git-game](#)
3. [git-game-v2](#)