* Git summarize

HEAD 指向的版本就是当前版本，因此，Git允许我们在版本的历史之间穿梭，使用命令 git reset --hard commit\_id 。

穿梭前，用 git log 可以查看提交历史，以便确定要回退到哪个版本。

要重返未来，用 git reflog 查看命令历史，以便确定要回到未来的哪个版

本。

场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命

令 git checkout -- file 。

场景2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修

改，分两步，第一步用命令 git reset HEAD file ，就回到了场景1，第二步按

场景1操作。

场景3：已经提交了不合适的修改到版本库时，想要撤销本次提交，参考版本回退一节，不过前提是没有推送到远程库。

命令 git rm 用于删除一个文件。如果一个文件已经被提交到版本库，那么你永远

不用担心误删，但是要小心，你只能恢复文件到最新版本，你会丢失最近一次提交

后你修改的内容。

Scenario 1:

PC1 has a project which framework development ready to submit to share with team. The PC1 repository has both master and dev branch and the all changes at dev branch have been merged to master before commit to remote. Task and steps:

1. Commit local repository to remote team share repository as master.

2. Generate a dev branch for development work which all team members pull to local and push the changes to it.

3. Everyone pull the dev branch and link to it to submit changes.

Git procedure:

1. create remote repository at hub or server.

2. push local repository to remote:

**$ git push origin master**

3. check remote status with detail:

**$ git remote -v**

**origin git@github.com:JimmyGitHub16/NodeLearning.gorigin git@github.com:JimmyGitHub16/NodeLearning.g**

4. push dev branch from pc1 to remote as a sharing branch for all team members’ development

**$ git push origin dev**

Check status:

**$ git remote -v**

5. pc2 and others clone remote repository to local first, then pull dev branch to local to develop new feature:

a. **$ git clone git@github.com:michaelliao/learngit.git**

//pc 2 checkout and create dev branch to local for development

**b. $ git checkout -b dev origin/dev**

// pc 2 or others submit local changes to remote:1. Pull latest remote dev branch to local and merge it if necessary(resolve conflicts) 2. // Push local to remote

// pull latest remote dev branch to local

6. Pc2 and others commit local changes to remote dev branch. Steps: 1. Pull latest remote dev branch to local, if any link issue, uses **git branch --set-upstream-to=origin/<branch> dev** to resolve it, 2. solve conflicts before pull to remote. 3. Commit changes to local dev branch before push to remote. 4. Push to remote

**$ git pull**

// if no tracking information as below, link it by highlight command

From github.com:JimmyGitHub16/NodeLearning

4feaf01..908c0dd dev -> origin/dev

6c4c578..917a71d master -> origin/master

There is no tracking information for the current branch.

Please specify which branch you want to merge with.

See git-pull(1) for details.

git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

git branch --set-upstream-to=origin/<branch> dev

// if any conflict, link local dev branch to remote

**$ git branch --set-upstream-to=origin/dev**

// merge conflict by edit conflict files, then add and submit to local repository // before push to remote

**$ git add fileName**

**$ git commit -m "merged"**

// push local dev branch to remote

**$ git push origin dev**

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn

$ pws

bash: pws: command not found

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn

$ pwd

/c/gitlearn

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn

$ git init

Initialized empty Git repository in C:/gitlearn/.git/

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ ls

readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git add readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git commit -m "wrote a file

> "

[master (root-commit) c217f33] wrote a file

1 file changed, 2 insertions(+)

create mode 100644 readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git diff

**diff --git a/readme.txt b/readme.txt**

**index 6090292..566c4d3 100644**

**--- a/readme.txt**

**+++ b/readme.txt**

@@ -1,2 +1,3 @@

get git all features.

that all.

+Modify this file.

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

Changes not staged for commit:

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

(use "git checkout -- <file>..." to discard changes in working directory)

modified: readme.txt

no changes added to commit (use "git add" and/or "git commit -a")

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git add readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

modified: readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git commit -m "commit changes"

[master c9f89f5] commit changes

1 file changed, 1 insertion(+)

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

nothing to commit, working tree clean

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git log

commit c9f89f5592e137e94c551e62a60935e3971aa495

Author: Jimmy <JimmyWangmy2016@gmail.com>

Date: Fri Feb 9 14:26:46 2018 +0800

commit changes

commit c217f33ddc4e4982bab6537bb15049a255ed463c

Author: Jimmy <JimmyWangmy2016@gmail.com>

Date: Fri Feb 9 14:23:56 2018 +0800

wrote a file

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git reset --hard HEAD^

HEAD is now at c217f33 wrote a file

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git reflog

c217f33 HEAD@{0}: reset: moving to HEAD^

c9f89f5 HEAD@{1}: commit: commit changes

c217f33 HEAD@{2}: commit (initial): wrote a file

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git reset --hard C2f89

fatal: ambiguous argument 'C2f89': unknown revision or path not in the working tree.

Use '--' to separate paths from revisions, like this:

'git <command> [<revision>...] -- [<file>...]'

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git reset --hard C9f89

HEAD is now at c9f89f5 commit changes

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ ls

readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

nothing to commit, working tree clean

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

Changes not staged for commit:

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

(use "git checkout -- <file>..." to discard changes in working directory)

modified: readme.txt

Untracked files:

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

lic.txt

no changes added to commit (use "git add" and/or "git commit -a")

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git add readme.txt lic.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: lic.txt

modified: readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git commit -m "submit 2 together"

[master 9d303b0] submit 2 together

2 files changed, 1 insertion(+)

create mode 100644 lic.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git add readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

modified: readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git commit -m "git track change test"

[master ec61d35] git track change test

1 file changed, 2 insertions(+), 1 deletion(-)

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

Changes not staged for commit:

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

(use "git checkout -- <file>..." to discard changes in working directory)

modified: readme.txt

no changes added to commit (use "git add" and/or "git commit -a")

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git diff

**diff --git a/readme.txt b/readme.txt**

**index 16cdde9..6652bcd 100644**

**--- a/readme.txt**

**+++ b/readme.txt**

@@ -2,4 +2,5 @@ get git all features.

that all.

Modify this file.

Ok.

-check modify.

\ No newline at end of file

+check modify.

+modify second time

\ No newline at end of file

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git diff HEAD -- readme.txt

**diff --git a/readme.txt b/readme.txt**

**index 16cdde9..6652bcd 100644**

**--- a/readme.txt**

**+++ b/readme.txt**

@@ -2,4 +2,5 @@ get git all features.

that all.

Modify this file.

Ok.

-check modify.

\ No newline at end of file

+check modify.

+modify second time

\ No newline at end of file

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git add readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git commit -m "that all"

[master 989d8cb] that all

1 file changed, 2 insertions(+), 1 deletion(-)

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

nothing to commit, working tree clean

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

Changes not staged for commit:

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

(use "git checkout -- <file>..." to discard changes in working directory)

modified: readme.txt

no changes added to commit (use "git add" and/or "git commit -a")

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git diff

**diff --git a/readme.txt b/readme.txt**

**index 6652bcd..5795c3e 100644**

**--- a/readme.txt**

**+++ b/readme.txt**

@@ -3,4 +3,5 @@ that all.

Modify this file.

Ok.

check modify.

-modify second time

\ No newline at end of file

+modify second time

+all the lazy status time you spend will cause the difficult of future with you

\ No newline at end of file

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git check out --readme

git: 'check' is not a git command. See 'git --help'.

Did you mean one of these?

checkout

check-attr

cherry

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git check out --readme.txt

git: 'check' is not a git command. See 'git --help'.

Did you mean one of these?

checkout

check-attr

cherry

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git checkout --readme.txt

error: unknown option `readme.txt'

usage: git checkout [<options>] <branch>

or: git checkout [<options>] [<branch>] -- <file>...

-q, --quiet suppress progress reporting

-b <branch> create and checkout a new branch

-B <branch> create/reset and checkout a branch

-l create reflog for new branch

--detach detach HEAD at named commit

-t, --track set upstream info for new branch

--orphan <new-branch>

new unparented branch

-2, --ours checkout our version for unmerged files

-3, --theirs checkout their version for unmerged files

-f, --force force checkout (throw away local modifications)

-m, --merge perform a 3-way merge with the new branch

--overwrite-ignore update ignored files (default)

--conflict <style> conflict style (merge or diff3)

-p, --patch select hunks interactively

--ignore-skip-worktree-bits

do not limit pathspecs to sparse entries only

--ignore-other-worktrees

do not check if another worktree is holding the given ref

--progress force progress reporting

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ ls

lic.txt readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git checkout -- readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git add readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ cat readme.txt

get git all features.

that all.

Modify this file.

Ok.

check modify.

modify second time

gag

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

modified: readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git reset HEAD readme.txt

Unstaged changes after reset:

M readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

Changes not staged for commit:

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

(use "git checkout -- <file>..." to discard changes in working directory)

modified: readme.txt

no changes added to commit (use "git add" and/or "git commit -a")

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git checkout -- readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git add lic.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git commit -m "add lic.txt

> "

[master cc96693] add lic.txt

1 file changed, 1 insertion(+)

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ rm lic.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

deleted: lic.txt

no changes added to commit (use "git add" and/or "git commit -a")

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git checkout -- lic.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ ls

lic.txt readme.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ rm lic.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

deleted: lic.txt

no changes added to commit (use "git add" and/or "git commit -a")

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git rm lic.txt

rm 'lic.txt'

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git commit -m "remove lic.txt"

[master 4f3a392] remove lic.txt

1 file changed, 1 deletion(-)

delete mode 100644 lic.txt

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$ git status

On branch master

nothing to commit, working tree clean

NapIOS@IOSDEV\_02 MINGW64 /c/gitlearn (master)

$