Help Commands

Git Help Commands

Help commands here

Lecture Command Listing

git help

git help config

Command Reference

Getting general help:

git help

Help Command Syntax -- getting help about a specific Git command:

git help command

Configuration Commands

Git Configuration Commands

What's the current directory (present working directory)?

Pwd OR cd

Git Config (Global/User-level) Syntax

git config --global setting value

Configure User and Email

General Syntax:

git config --global user.name "Your Name"

git config --global user.email "you@someplace.com"

Example using course author's information:

git config --global user.name "Jason Taylor"

git config --global user.email "jason@jasongtaylor.com"

Listing All Global Configuration Settings

git config --global --list

Seeing Git's User-based Config file

cat ~/.gitconfig

Starting Commands

Git Starting Commands

Lecture Command Listing - Fresh Start

Pwd OR cd

cd projects/

git init git-demo

Lecture Command Listing - Start with Existing Project

pwd OR cd

cd projects/

cd website/

ls OR dir

git init

Command Reference

Present Workding Directory

pwd OR cd

Change Directory

cd folder-name

First Commit Commands

Git First Commit Commands

Lecture Command Listing

pwd OR cd

ls OR dir

mate README.md

ls OR dir

git status

git add README.md

git status

git commit -m "Initial commit"

clear

git status

Command Reference

List

ls OR dir

Lists files and folders in current directory. Without parameters, will list non-hidden folders and files.

Git Status

git status

Shows which files have been modified in the working directory vs Git's staging area.

Git Add

git add file-name

Adds the new or newly modified *file-name* to Git's staging area (index).

Git Commit

git commit -m "A really good commit message"

Commits all files currently in Git's staging area. The -m parameter allows for a commit message directly from the command line.

Clear!

clear

Clears all previous commands from the terminal screen -- just a bit of clean up.

Text Mate

mate file-name

All command line demos are preformed on the MacOS. Creating and editing files is done with TextMate 2 (free) using the **mate** command from Terminal. Passing a *file-name* to the **mate** command will create or open that file. Windows users can use the **notepad *file-name*** command instead.

Working Locally Commands

Git Working Locally Commands

Lecture Command Listing - Working Locally, Part One

pwd OR cd

git status

mate README.md

git status

git add README.md

git status

git commit -m "Adding some ipsum"

clear

git status

mate README.md

git status

git commit -am "Adding more ipsum"

git status

Lecture Command Listing - Working Locally, Part Two

pwd OR cd

git status

clear

mate index.html

git status

git add index.html

git status

mate README.md

git status

clear

git status

git add README.md

git status

git commit -m "A few changes for the website"

clear

mate README.md

mate index.html

git status

git add .

git status

git commit -m "A few more changes for website"

clear

mate README.md

git status

git add README.md

git status

git reset HEAD README.md

clear

git status

mate README.md

git checkout -- README.md

mate README.md

git status

Command Reference

Express Commit for Tracked files

git commit -am "Awesome commit message"

Use the *-a* parameter with the **git commit** command to directly commit newly modified tracked files. Warning: Only do this for small changes. Tracked files are files that have been previously added to Git (committed or staged).

Adding All Changed Files

git add .

The period parameter for the git add command will recursively add all new and newly modified files.

Unstage File

git reset HEAD file-name

Following the above command will "unstage" the specified file from Git's staging area (aka index).

Backout Working Directory Changes

git checkout -- file-name

Following the above command will back out any changes made to the specified file and replace it with the version last committed in Git

History and File Management Commands

Git History / File Management Commands

Lecture Command Listing -- History

git log

git help log

git log --oneline --graph --decorate --color

Lecture Command Listing -- Removing Files

pwd OR cd

git status

mate debug.log

ls OR dir

git status

git add .

git status

git commit -m "adding log file that really does not belong here"

clear

git status

git rm debug.log

ls OR dir

git status

git commit -m "removing log file"

clear

mate info.log

ls OR dir

git add info.log

git commit -m "adding info log"

git status

clear

ls OR dir

rm info.log

ls OR dir

git status

git add .

git add -u

clear

git status

git commit -m "Removing info.log"

Lecture Command Listing -- Moving Files

ls OR dir

mkdir web

ls OR dir

git mv index.html web

cd web/

ll

pwd OR cd

cd ..

ls OR dir

git status

git commit -m "Moving index.html file to web folder"

clear

Lecture Command Listing -- Ignoring Files

mate application.log

ls OR dir

git status

mate .iitignore

git status

ls -a OR pwd

git add .gitignore

clear

git status

git commit -m "adding ignore file"

Command Reference

Seeing Repository History

git log

git log --oneline --graph --decorate --color

Git's **log** command displays the repository's history in reverse chronological order. The no-params version displays the standard view.

Git log options from above: --oneline Compacts log data on to one line, abbreviating the SHA1 hash --graph Adds asterisk marks and pipes next to each commit to show the branching graph lines --decorate Adds the markers for branch names and tags next to corresponding commits --color Adds some color to the output -- nice to have, depending on the operating system

Removing a file using Git

git rm file-name

Removing a file using Terminal

rm file-name

This removes the file outside Git's knowledge

Updating Git's Index (staging area)

git add -u

The *-u* parameter will recursively update Git's staging area regarding deleted/moved files outside of Git.

Making a directory (folder)

mkdir folder-name

The **mkdir** command is a nearly universal command for creating a directory/folder.

Making a directory (folder)

git mv source destination

The **git mv** command will move the *source* (file or folder) to the *destination* with Git.

SSH Authentication Commands

SSH Authentication Commands

Lecture Command Listing

cd ~

cd .ssh

mkdir .ssh

cd .ssh

pwd OR cd

ssh-keygen -t rsa -C "jason@jasongtaylor.com"

mate id\_rsa.pub

ssh -T git@github.com

Command Reference

Generating an SSH Key

ssh-keygen -t rsa -C "your.name@your-company.com"

Use your actual email address in the example above.

Verify SSH authentication

ssh -T git@github.com

Above command uses **ssh** to connect to GitHub over the SSH protocol.

Git Remote Commands

Git Remote Commands

Lecture Command Listing

git status

Add new Remote named origin and set url

git remote add origin git@github.com:scm-ninja/git-demo.git

git remote -v

git push -u origin master

git push origin master

ls OR dir

cd web/

mate index.html

clear

git commit -am "Updating index page for GH"

git status

git pull origin master

git push origin master

Command Reference

Creating a remote repository reference

git remote add remote-name remote-repository-location

Using **git remote add** command allows us to associate a remote repository. Normally, you want to paste in the full URL for the remote repository given to you by your Git host (GitHub). By convention, the first or primary remote repository is named *origin*.

List Git's Remotes

git remote -v

The **git remote** command lists the names of all the remote repositories and the -v parameter (verbose) will display the full URL of the remote repository for each remote name listed

Send Changes to Remote

git push -u remote-name branch-name

git push remote-name branch-name

The **git push** sends all your local changes (commits) on branch *branch-name* to the remote named *remote-name*. The **-u** parameter is needed the first time you push a branch to the remote.

Receive Changes from Remote

git pull remote-name branch-name

The **git pull** receives all your remote changes (commits) from the remote named *remote-name* and on branch *branch-name*.

Additional Resources

Git Resources

Below are some recommended resources to continue your Git education.

Books

[Pro Git](http://git-scm.com/book) - Free Book on Git, worth every penny.

|  |  |  |
| --- | --- | --- |
| **Windows command** | **Unix command** | **Notes** |
| arp | [arp](http://www.nxmnpg.com/arp) |  |
| assign | [ln](http://www.nxmnpg.com/ln) | Create a file link |
| assign | [ln](http://www.nxmnpg.com/ln) -s | On Unix, a directory may not have multiple links, so instead a symbolic link must be created with ln -s. |
| assoc | [file](http://www.nxmnpg.com/file) |  |
| at | [at](http://www.nxmnpg.com/at) [batch](http://www.nxmnpg.com/batch) [cron](http://www.nxmnpg.com/cron) |  |
| attrib | [chown](http://www.nxmnpg.com/chown) [chmod](http://www.nxmnpg.com/chmod) | Sets ownership on files and directories |
| cd | [cd](http://www.nxmnpg.com/cd) | On Windows, cd alone prints the current directory, but on Unix cd alone returns the user to his home directory. |
| cd | [pwd](http://www.nxmnpg.com/pwd) | On Windows, **cd** alone prints the current directory. |
| chkdsk | [fsck](http://www.nxmnpg.com/fsck) | Checks filesystem and repairs filesystem corruption on hard drives. |
| cls | [clear](http://www.nxmnpg.com/clear) | Clear the terminal screen |
| copy | [cp](http://www.nxmnpg.com/cp) |  |
| date time | [date](http://www.nxmnpg.com/date) | Date on Unix prints the current date and time. Date and time on Windows print the date and time respectively, and prompt for a new date or time. |
| del | [rm](http://www.nxmnpg.com/rm) |  |
| deltree | [rm](http://www.nxmnpg.com/rm) -r | Recursively deletes entire directory tree |
| dir | [ls](http://www.nxmnpg.com/ls) | “dir” also works on some versions of Unix. |
| doskey /h F7 key | [history](http://www.nxmnpg.com/history) | The Unix history is part of the Bash shell. |
| edit | vi emacs etc. | edit brings up a simple text editor in Windows. On Unix, the environment variable EDITOR should be set to the user’s preferred editor. |
| exit | [exit](http://www.nxmnpg.com/exit) Control-D | On Unix, pressing the control key and D simultaneously logs the user out of the shell. |
| explorer | nautilus etc. | The command explorer brings up the file browser on Windows. |
| fc | [diff](http://www.nxmnpg.com/diff) |  |
| find | [grep](http://www.nxmnpg.com/grep) |  |
| ftp | [ftp](http://www.nxmnpg.com/ftp) |  |
| help | [man](http://www.nxmnpg.com/man) | “help” by itself prints all the commands |
| hostname | [hostname](http://www.nxmnpg.com/hostname) |  |
| ipconfig /all | [ifconfig](http://www.nxmnpg.com/ifconfig) -a | The /all option lets you get the MAC address of the Windows PC |
| mem | [top](http://www.nxmnpg.com/top) | Shows system status |
| mkdir | [mkdir](http://www.nxmnpg.com/mkdir) |  |
| more | [more](http://www.nxmnpg.com/more) [less](http://www.nxmnpg.com/less) |  |
| move | [mv](http://www.nxmnpg.com/mv) |  |
| net session | [w](http://www.nxmnpg.com/w) [who](http://www.nxmnpg.com/who) |  |
| net statistics | [uptime](http://www.nxmnpg.com/uptime) |  |
| nslookup | [nslookup](http://www.nxmnpg.com/nslookup) |  |
| ping | [ping](http://www.nxmnpg.com/ping) |  |
| print | lpr | Send a file to a printer. |
| reboot shutdown -r | [shutdown](http://www.nxmnpg.com/shutdown) -r |  |
| regedit | edit /etc/\* | The Unix equivalent of the Windows registry are the files under /etc and /usr/local/etc. These are edited with a text editor rather than with a special-purpose editing program. |
| rmdir | [rmdir](http://www.nxmnpg.com/rmdir) |  |
| rmdir /s | [rm](http://www.nxmnpg.com/rm) -r | Windows has a y/n prompt. To get the prompt with Unix, use rm -i. The i means “interactive”. |
| set | [env](http://www.nxmnpg.com/env) | Set on Windows prints a list of all environment variables. For individual environment variables, set <variable> is the same as echo $<variable> on Unix. |
| set Path | echo $PATH | Print the value of the environment variable using set in Windows. |
| shutdown | [shutdown](http://www.nxmnpg.com/shutdown) | Without an option, the Windows version produces a help message |
| shutdown -s | [shutdown](http://www.nxmnpg.com/shutdown) -h | Also need -f option to Windows if logged in remotely |
| sort | [sort](http://www.nxmnpg.com/sort) |  |
| start | & | On Unix, to start a job in the background, use command &. On Windows, the equivalent is start command. See [How to run a Windows command as a background job like Unix ?](http://www.tomshardware.com/forum/34598-45-windows-command-background-unix). |
| systeminfo | [uname](http://www.nxmnpg.com/uname) -a |  |
| tasklist | [ps](http://www.nxmnpg.com/ps) | “tasklist” is not available on some versions of Windows. See also [this article on getting a list of processes in Windows using Perl](http://www.lemoda.net/perl/win-ps-list/win-ps-list.html) |
| title | ? | In Unix, changing the title of the terminal window is possible but complicated. Search for “change title xterm”. |
| tracert | [traceroute](http://www.nxmnpg.com/traceroute) |  |
| tree | [find](http://www.nxmnpg.com/find) [ls](http://www.nxmnpg.com/ls) -R | On Windows, use tree | find “string” |
| type | [cat](http://www.nxmnpg.com/cat) |  |
| ver | [uname](http://www.nxmnpg.com/uname) -a |  |
| xcopy | [cp](http://www.nxmnpg.com/cp) -R | Recursively copy a directory tree |