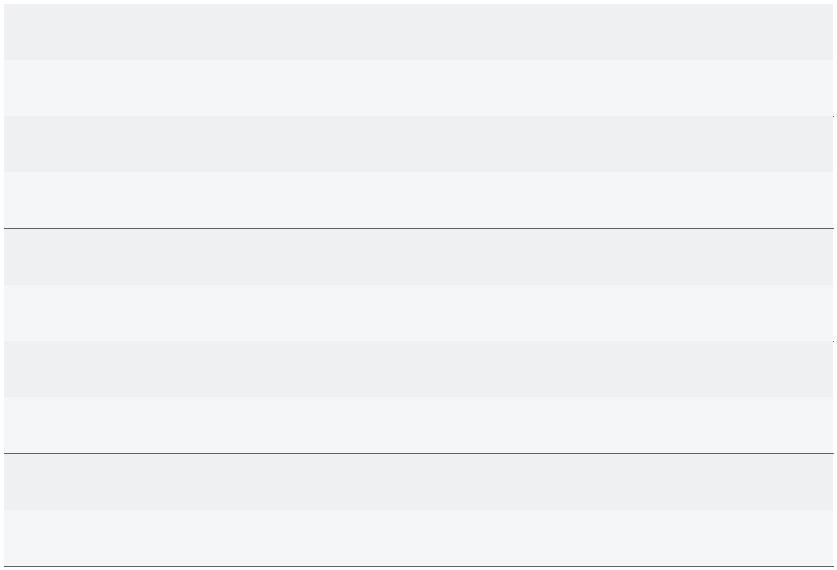
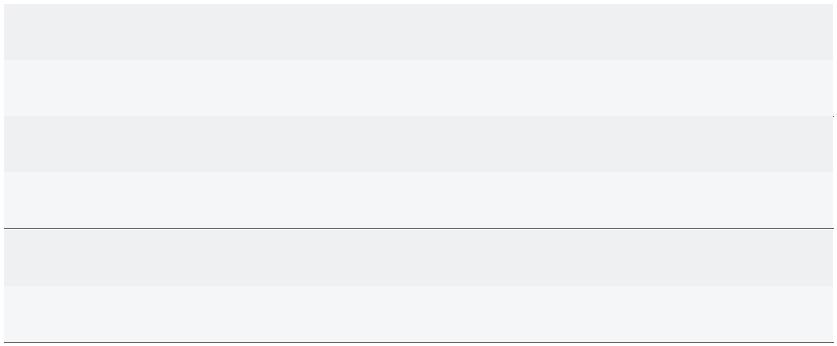
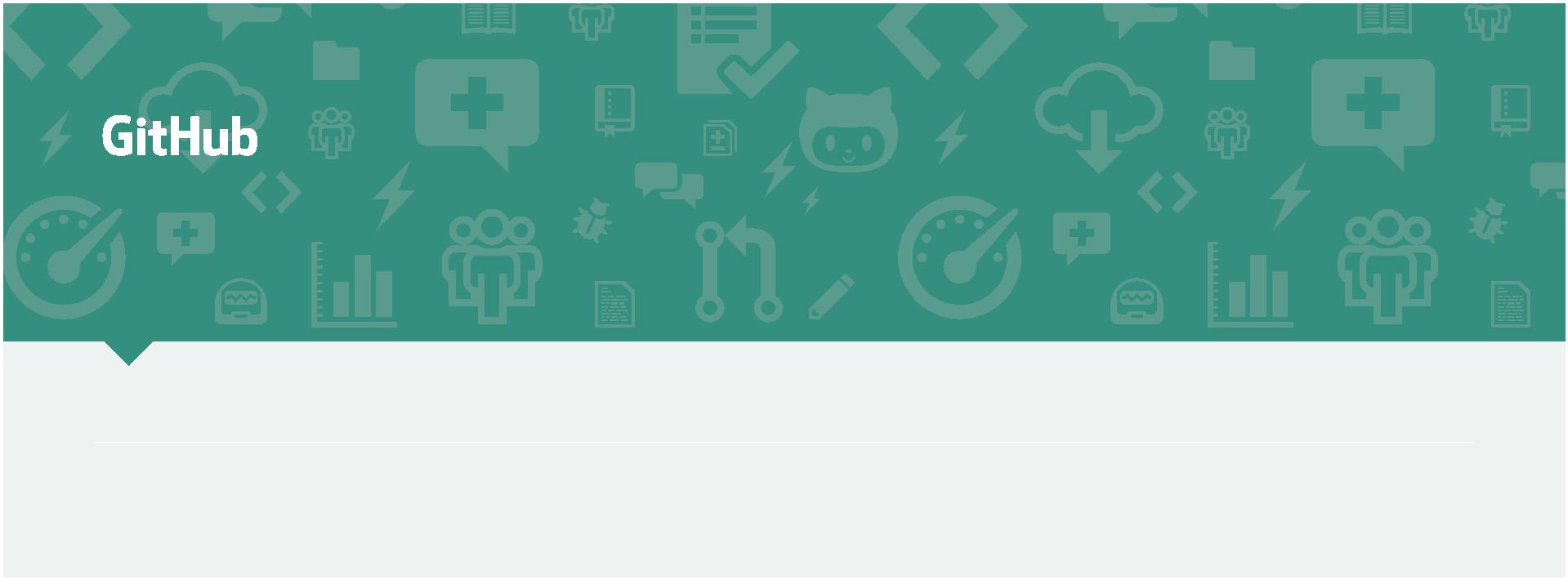
GIT CHEAT SHEET ssl cloud



Git is the free and open source distributed version control system that's responsible for everything GitHubrelated that happens locally on your computer. This cheat sheet features the most important and commonlyused Git commands for easy reference.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INSTALLATION & GUIS  With platform speciﬁc installers for Git, GitHub also provides the  ease of staying up-to-date with the latest releases of the command  line tool while providing a graphical user interface for day-to-day  interaction, review, and repository synchronization. |  | STAGE & SNAPSHOT  Working with snapshots and the Git staging area  git status  show modiﬁed ﬁles in working directory, staged for your next commit | | | | | | |
| GitHub for Windows  htps://windows.github.com | | |  | git add [file] |
| GitHub for Mac  htps://mac.github.com | | |  | add a ﬁle as it looks now to your next commit (stage)  git reset [file] | | |
| For Linux and Solaris platforms, the latest release is available on  the oﬃcial Git web site. | | |  | unstage a ﬁle while retaining the changes in working directory  git diff | | | |
| Git for All Platforms  htp://git-scm.com | | |  | diﬀ of what is changed but not staged  git diff --staged | |

diﬀ of what is staged but not yet commited

|  |  |  |
| --- | --- | --- |
| SETUP  Conﬁguring user information used across all local repositories |  | git commit -m “[descriptive message]”  commit your staged content as a new commit snapshot |

git config --global user.name “[firstname lastname]”

|  |  |  |  |
| --- | --- | --- | --- |
|  | set a name that is identiﬁable for credit when review version history  git config --global user.email “[valid-email]”  set an email address that will be associated with each history marker |  | BRANCH & MERGE  Isolating work in branches, changing context, and integrating changes |

git branch

git config --global color.ui auto

list your branches. a \* will appear next to the currently active branchset automatic command line coloring for Git for easy reviewing

git branch [branch-name]

create a new branch at the current commit

|  |  |  |
| --- | --- | --- |
| SETUP & INIT  Conﬁguring user information, initializing and cloning repositories |  | git checkout  switch to another branch and check it out into your working directory |

git init git merge [branch]

initialize an existing directory as a Git repository merge the speciﬁed branch’s history into the current one

git clone [url] git log

retrieve an entire repository from a hosted location via URL show all commits in the current branch’s history

|  |  |  |
| --- | --- | --- |
| INSPECT & COMPARE  Examining logs, diﬀs and object information |  | SHARE & UPDATE  Retrieving updates from another repository and updating local repos |

git log git remote add [alias] [url]

show the commit history for the currently active branch add a git URL as an alias

git log branchB..branchA git fetch [alias]

show the commits on branchA that are not on branchB fetch down all the branches from that Git remote

git log --follow [file] git merge [alias]/[branch]

show the commits that changed ﬁle, even across renames merge a remote branch into your current branch to bring it up to date

git diff branchB...branchA git push [alias] [branch]

show the diﬀ of what is in branchA that is not in branchB Transmit local branch commits to the remote repository branch

git show [SHA] git pull

show any object in Git in human-readable format fetch and merge any commits from the tracking remote branch

|  |  |  |
| --- | --- | --- |
| TRACKING PATH CHANGES  Versioning ﬁle removes and path changes |  | REWRITE HISTORY  Rewriting branches, updating commits and clearing history |

git rebase [branch]git rm [file]

apply any commits of current branch ahead of speciﬁed onedelete the ﬁle from project and stage the removal for commit

git reset --hard [commit]

git mv [existing-path] [new-path]

clear staging area, rewrite working tree from speciﬁed commitchange an existing ﬁle path and stage the move

git log --stat -M

show all commit logs with indication of any paths that moved TEMPORARY COMMITS

Temporarily store modiﬁed, tracked ﬁles in order to change branches

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IGNORING PATTERNS  Preventing unintentional staging or commiting of ﬁles | | |  | git stash  Save modiﬁed and staged changes |
|  | logs/  \*.notes  pattern\*/ | |  | git stash list  list stack-order of stashed ﬁle changes | | |
|  | | Save a ﬁle with desired paterns as .gitignore with either direct string  matches or wildcard globs. |  | git stash pop  write working from top of stash stack | |

git config --global core.excludesfile [file] git stash drop

system wide ignore patern for all local repositories discard the changes from top of stash stack

ssl cloud

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | [sslcloudlearning@gmail.com](mailto:sslcloudlearning@gmail.com) |