Git Cheatsheet

Glossary

Keywords	Description
`git`	Open-source distributed version-control system, used to store code in repositories
`staging`	Proposed files/directories that you'd like to commit
`commit`	Saving all staged files/directories to your local repository
`branch`	An independent line of development, so you can develop features isolated from each other. Master branch is the default.
`clone`	Local version of a repository, including all commits and branches
`remote`	Common repository on eg. Github that all team members to keep that changes in sync with
`fork`	Copy of a repository owned by a different user
`pull request`	A method of submitting contributions to a repository
`HEAD`	Represents your current working directory

Configuration

Key/Command	Description
`git config --global user.name [name]`	Set author name to be used for all commits (your account name)
`git config --global user.email [email]`	Set author email to be used for all commits (your account email)
`git config color.ui true`	Enables helpful colorization of command line output

Core Commands

Key/Command	Description
`git init [directory]`	Creates new local repository
`git clone [repo]`	Creates local copy of remote repository
`git add [directory]`	Stages specific [directory]
`git add [file]`	Stages specific [file]
`git add -A`	Stages all changed files
`git commit -m "[message]"`	Commit everything that is staged and write a short commit comment
`git commit`	Opens your default editor (vim) to write a commit message. The ideal commit message has a title (<50 chars) followed by a detailed explanation (optional)
`git status`	Shows status of changes as untracked, modified or staged

Synchronization of Changes

Key/Command	Description
`git fetch`	Downloads all history from the remote branches
`git merge`	Merges remote branch into current local branch
`git pull`	Downloads all history from the remote branch and merges into the current local branch
`git push`	Pushes all the commits from the current local branch to its remote equivalent

Tip: `git pull` is the combination of `git fetch` and `git merge`

Undo Changes

Key/Command	Description
`git checkout -- [file]`	Replace file with contents from HEAD
`git revert [commit]`	Create new commit that undoes changes made in [commit], then apply it to the current branch
`git reset [file]`	Remove [file] from staging area
`git reset --hard HEAD`	Removes all local changes in working directory
`git reset --hard [commit]`	Reset your HEAD pointer to previous commit and discard all changes since then

Branches

Key/Command	Description
`git branch [branch]`	Create a new branch
`git checkout [branch]`	Switch to that branch
`git checkout -b [branch]`	Create and checkout new branch
`git merge [branch]`	Merge [branch] into current branch
`git branch -d [branch]`	Deletes the [branch]
`git push origin [branch]`	Push [branch] to remote
`git branch`	Lists local branches