OPEN AUSTIN
Git Cheatsheet

> go to the project's repo
> read through the README
> use this cheatsheet to help you navigate git!

GETTING STARTED

`$ git clone <repo>`
clone the <repo> onto your local machine from a remote location like Github. accessible via HTTP or SSH

`$ git checkout -b <branch name>`
create your own branch before yous tart coding. look at the README / contribution guidelines for info on how to name your branch.

`$ git log --pretty`
view a nicely formated and detailed commit history to see what contributors have been working on.

OOPS!

`$ git reset --hard`
reset your staging area and current directory to the lasest commit and *overwrite all current changes* (careful with this one!)

`$ git commit --amend`
correct a mistake in your last commit message

LET'S CODE

`$ git status`
view your current staged, unstaged, and untracked changes and see where you are

`$ git add .`
stage all of your changes. try 'git add -p' to add your changes as you review them in small hunks of code

`$ git commit -m`
commit those staged changes and write a commit message

`$ git diff`
view any unstaged changes in your current directory that differ from your last commit

`$ git checkout master`
checkout (switch to) the master branch

`$ git stash`
set aside your current changes and get back to your latest commit by stashing your changes. you can get those changes back with 'git stash pop'

BRANCHES

`$ git branch`
view all the branches on your local machine. don't let them get unwieldy!

`$ git merge <branch name>`
merge the <branch name> into the current branch that you're on.

`$ git branch -d <branch name>`
all done with a branch? use this to delete it from your local machine

FINISH UP!

`$ git push origin <branch name>`
once you're all done, push your branch up to origin. now you can go to github and make your pull request

Nicely done! Tweet us about your pull request **@OpenAustin.**

Thanks to @CodeNewbies for the design & concept. Check out their resources at www.codenewbie.org.

OPEN AUSTIN
Git Cheatsheet

> go to the project's repo
> read through the README
> use this cheatsheet to help you navigate git!

GETTING STARTED

`$ git clone <repo>`
clone the <repo> onto your local machine from a remote location like Github. accessible via HTTP or SSH

`$ git checkout -b <branch name>`
create your own branch before yous tart coding. look at the README / contribution guidelines for info on how to name your branch.

`$ git log --pretty`
view a nicely formated and detailed commit history to see what contributors have been working on.

OOPS!

`$ git reset --hard`
reset your staging area and current directory to the lasest commit and *overwrite all current changes* (careful with this one!)

`$ git commit --amend`
correct a mistake in your last commit message

LET'S CODE

`$ git status`
view your current staged, unstaged, and untracked changes and see where you are

`$ git add .`
stage all of your changes. try 'git add -p' to add your changes as you review them in small hunks of code

`$ git commit -m`
commit those staged changes and write a commit message

`$ git diff`
view any unstaged changes in your current directory that differ from your last commit

`$ git checkout master`
checkout (switch to) the master branch

`$ git stash`
set aside your current changes and get back to your latest commit by stashing your changes. you can get those changes back with 'git stash pop'

BRANCHES

`$ git branch`
view all the branches on your local machine. don't let them get unwieldy!

`$ git merge <branch name>`
merge the <branch name> into the current branch that you're on.

`$ git branch -d <branch name>`
all done with a branch? use this to delete it from your local machine

FINISH UP!

`$ git push origin <branch name>`
once you're all done, push your branch up to origin. now you can go to github and make your pull request

Nicely done! Tweet us about your pull request **@OpenAustin.**

Thanks to @CodeNewbies for the design & concept. Check out their resources at www.codenewbie.org.