

Git Commands - Sheet1

git	init	
git	status	
git	add	file_name
git	commit	.
git	commit -m "messagei282"	.
git	log	
git	add	.
git	add	*.html
git	branch	branch_name
git	checkout	branch_name
git	merge	my_branch
git	commit -a	.
git	mergetool	
git	stash	
git	stash apply	
git	remote	
git	clone	https://github.com/username/repositoryname.git
git	remote -v	
git	fetch	origin
git	pull	origin
git	push	origin master
git	remote add	name https://github.com/username/repositoryname.git
git	config --global user.name	"user_name"
git	config --global user.email	"email@test.com"
	touch	.gitignore

Initializes Git / creates ".git" hidden folder
Provides Status of condition
Stages "file_name" before Committing / Tracks the current version of the file
brings you to a vim screen to ad a commit message
Commits with the message within the inverted commas(messagei282)
Shows commit log
stages current directory
stages current folder's html files only(selective staging)
Makes and names the new branch as "branch_name"
Moves from current branch to branch_name
merges "my_branch" to current branch
skips staging
requires a mergetool to be installed- helps merge
saves working directory to return to later on
retrieves and restores changes from stash
lists remote repositories
clones git repository of username"username" with repository name "repositoryname"
shows URLs for origin's git repositories
Retrieves changes made on origin since previous clone and commit.the command does not merge both copies.
will fetch and merge into current branch
uploads current origin directory to the remote git repository on branch master
stages the current repository remotely and names the version of the stages file to "name"
configures git completely on the system to the user "user_name"
configures git throughout on the system to user email "email@test.com"
ignores content placed within .gitignore while committing. Touch is used since windows requires a file name for a file wi