**GIT COMMANDS**

**Useful Resources:** git-scm.com, docs.github.com

**Tools:** Windows PowerShell x86

VS Code extension: **Git graph**

**git --version** / **git -v**

**git status**

**ls -la** (for hidden folders)

**git init** (initialize git)

**Get-ChildItem -Force** (view subfolders)

**cd..** (get out of the current folder)

**git push**

**git commit**

**pwd** (present working directory)

**New-Item -ItemType File -Name "newfile.txt"** (Add new text file)

**git add** (Add file to staging area)

**git add .** (Add all files to staging area)

**git commit -m "Add commands.md"** (Save files to Git)

**“files with spaces or long”** – use quotation marks

**NOTES:**

**clone** (Repository hosted somewhere like GitHub)

**add** (edit files and add changes to GIT)

**commit** (save files/changes to GIT)

**git log**

**git log –oneline**

(All git logs in one line)

**q** (Escape **log file**)

**git config**

(How it gets to know about your details)

**git config --global user.name "Christopher Mawela"**

**git config --global user.email “**christopher.mawela@gmail.com”

**git config --global core.editor "code --wait"**

**.gitignore**

(don’t track files, e.g. node\_modules, files holding passwords and API keys)

**new-item .gitignore**

(Add empty gitignore file, ignores files holding sensitive info e.g. .env Variables files for holding API keys, etc**)**

**git commit**

**git log –oneline**

Google: gitignore generator (Choose any, they work the same. Copy and paste)

**C:\Users\Ndivhudza>** cat . gitconfig (Show Git configuration)

**cat .gitconfig**

**Unique ID (is called Hash)**

**Press CTRL + , => Settings “Commonly Used” remove \*\*/.git**

**FOLDER TWO**

**git status**

**git init** (Initialise repository)

**git rm -cached index.html** (Removes files from staging area)

**git branch** (shows current git branch)

**new-item index.html (**Add file, and makes changes after git status CMD)

**git add index.html**

**git commit -m "Modified index file"**

**git checkout** "nav-bar"(Switch to another branch, commit before switch)

**git merge** footer (Merge a branch to current branch)

**git switch** main (same as above)

**git add** footer.html (Add to staging area)

**git diff --staged** (Diff on same file on staging area)

**git diff 91f09f6 4451e16**

**git diff 91f09f6..4451e16** (Same above)

**git diff branchone..branchtwo**

**git switch -c bugfix** (Creates and switches to new branch)

**git stash** (Can switch branches when changes are not staged and committed)

**git stash pop**

(when you move out of the branch via stash and switch back again your changes disappear, this CMD will bring back hose changes)

**git stash apply** stash@{0}

**git stash --list**

**git checkout <hash>** (Will go back to where the project was before this commit)

**git reflog** (rarely used)

**git checkout HEAD~2** (moves 2 commits back)

**GIT REBASE**

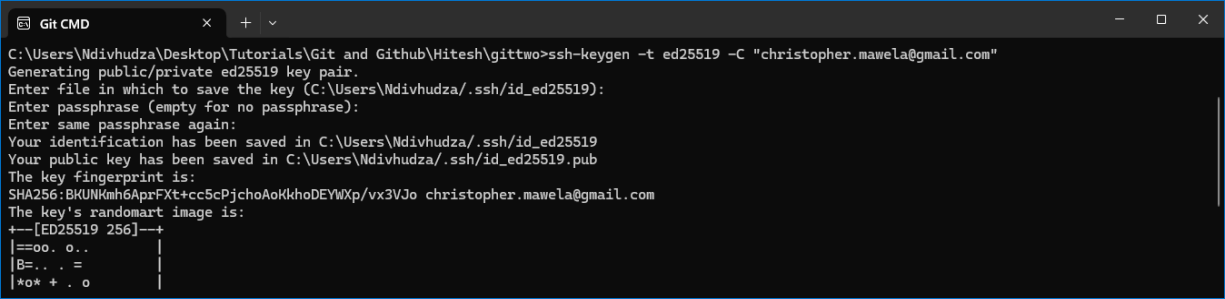
**git commit -am "Updated main website"** (Git add and commit)

**git rebase main** (To be careful used, only work on the branches not main / master branch)

**git rebase –-continue** (Continue after resolving conflicts when **rebase** paused)

**PUSH TO GITHUB**

Enter file in which to save the key (C:\Users\Ndivhudza/.ssh/id\_ed25519):





**git remote -v** (See if connected to remote Repo)

**git push -u origin master** (Allows git to push code to GitHub)

**git clone https://github.com/hiteshpw1/open-source.git** (Copy from GitHub to your desktop)

**git checkout –b navbar** (Switch to new navbar branch)

**git remote remove origin** (Removes remote Repo)

**git remote add origin https://github.com/hiteshchoudhary/open-source-11.git** (Add Repo origin and URL)

**KEVIN STRATVERT – Git and GitHub for Beginners Tutorial**

**Tool:** Git Bash

**git config –list** (Check your git config)

**$ git --version** (Check if Git is installed)

**GIT CONFIGURATION**

**$ git config --global user.name "Christopher Mawela"**

**$ git config --global user.email “christopher.mawela@gmail.com”**

**$ git config --global init.default branch main (Start git default to main branch)**

**$ git config -h (**help on config command)

**$ git help config** (Opens an offline manual on Git)

**$ git add index.html** (Add index to be tracked)

**$ touch .gitignore** (Add new file)

**ADD FILES TO STAGING AREA**

**$ git add** . (Add all files/ whole directory to staging)

**$ git add –all** (Add all files)

**$ git add -A** (Add all files)

**$ git restore --staged index.htm** (Removes from staging area to working)

**ADD CHANGES TO REPOSITORY**

**$ git commit -m "Adds all files to repository"**

**$ git diff** (Shows changes)

**$ git commit -a -m "Updated text"** (Skipped staging went to commit)

**$ git rm "secret recipe.htm"** (Secret recipe.htm deleted)

**$ git restore "secret recipe.htm"** (restore deleted file)

**$ git mv "KCC Logo.png" "Primary Logo.png"** (rename to Primary logo.png)

**$ git commit -m "Renamed KCC Logo to Primary logo" –amend** (Edit commit message)

**$ git log -P** (What changed in all of these commits)

**$ git help log** (git log help offline)

**$ git reset c193894** (goes back to previous commits)

**$ git rebase -i –root** (modifies the way things appear)

**ADD BRANCHES**

**$ git branch FixTemp** (new branch FixTemp)

Or

**$ git checkout -b FixTemp** (Creates and switches to new branch FixTemp)

**$ git switch FixTemp** (Switches to branch FixTemp)

**$ git commit -a -m "Updated temp"** (Add to staging and commit**)**

**$ git merge -m "Fix temp" FixTemp** (merge to main)

**$ git branch -d FixTemp** (delete FixTemp branch)

**$ git switch -c UpdateText** (create and switches to new branch Update text)

**GITHUB**

**$ git remote add origin** <https://github.com/nmawela35/KevinCookieCompany.git> (origin - on GitHub)

**$ git branch -m main master** (rename main to master branch)

**$ git branch -M main (**target branch main)

**$ git push -u origin main** (push main branch to GitHub)

**$ git push –all** (pushes all branches to GitHub)

**$ git fetch** (download all history from remote)

**$ git merge** (merge with what We have on local machine)

**$git pull** (combine fetch + merge)

**TRAVERSY MEDIA – Git and GitHub Crash Course For Beginners**

**$ git init**

**$ git config --global user.name "Christopher Mawela"**

**$ git config --global user.email christopher.mawela@gmail.com**

**$ git add \*.html (adds all HTML files to staging area)**

**$ git rm --cached index.html** (remove index file from staging area)

**$ git commit -m "Modifed index and App.js"**

**$ touch .gitignore**

**BRANCHES**

**$ git branch -m main master** (rename main to master branch)

**$ git checkout -b login** (creates and switches to new branch login)

**$ git merge login** (Merge login to Master)

**GITHUB**

**$ git remote** (List remote Repos)

**Add remote Repo** (Copy from GitHub)

**Git push -u origin master** (pushed master branch to GitHub)

**Add README.md**

**$ git clone** (copy the entire directory from GitHub)

**$ git pull**  (pull everything including changes)

**Git Tutorial for Beginners – Programming With Mosh**

**$git**