**Git Syntax and Notes**

Downloads: <https://git-scm.comdownloads> - [www.gitblit.com](http://www.gitblit.com) Last Update: 14 March 2023

Launch: ***Git Bash*** *from the Git folder within Window’s* ***Start***

**Basic Commands**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

**cd < . . . . . >** Change directory to < /path/… directoryName >

**clear** Clears the contents on the screen

**commit -m “<**message**>”** Example of command with an argument. Commit early/often- between stages

**git config –global --list** Lists the variables in Git-configuration (Works in any directory)

**git config—global user.name “< . . . >”**  Sets the user name to < user name >

**git config --global user.email “< . . . >”** Sets the user e-mail to < user e-mail >

**git config --global init.defaultBranch “< . . . >”**

Sets / change the default branch to < . . . > -> “main” if placed in the <. . .>

**git init** Establishes the current focused directory as a Repository directory

**git –version** Displays the current installed version of git

**ls** Displays visible files

**ls -a or** < **ls -la >** Displays visible files & hidden files: Stacked vertical < -a > , horizontal < -la >

**mkdir** Makes / Creates a new directory

**pwd** Displays working directory. The < ~ > character signifies base directory.

**cd < . . . . . >** Change directory to < /path/… directoryName > Backup directory **cd ..**

**Utility Commands**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

**Q** Exits the command

**git init** Establishes the present focused directory as a Repository directory

**git status** Checking the state of files in the working directory. Displays state of the working directory and differences between the working and stage area directories.

**git add “< . . . >”** Adding an untracked file to version control **staging area**. Creates **index** file in **.git**

**git add .** Adds all the files in the current directory to the stage area.

**git rm –cached “< . . . >”** Unstages the specified **“< . . . >”** file.

**git commit -m “<**message**>”** Save a file or a Saved a version of the file in the project including the message

**git log** Displays history of the commits of the repository in reverse chronological order.

Shows “ **(HEAD -> main)** ” branch.

**git log –all** Displays all the commits not just the ones back to the parent branches

**git branch** Displays current focused branch

**git branch** <branchName >Creates a branch with name < …. >

**git switch** <branchName > Switch to the branch <branchName >

**git merge** <branchName > Merges source branch <branchName > into current branch (where Head points)

**Git Working Areas:** •**Repository •Working Directory •Staging Area •Commit History**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

**•Repository:**

Directory containing the hidden directory **.git** This Repository directory is created when the project directory (current) is initialized using **.git init** ( .git directory contains directories hooks, info, objects, refs and files config, description, Head )

**•Working Directory**

The Working Directory is the **project directory contents** (not including the **.git** directory). The workbench of the project.

This is the directory location where all modifications are made: directories and files are added, edited and deleted.

**•Staging Area**

Staging area is like a rough draft space. A preparation area for contents to be included in the next save or commit.

The **index** file is created int the ‘**.git’** directory when a file is added to the staging area. The ‘Staging’ area is within the Repository (.git) area.

**•Commit**

A commit is basically one snapshot version of a project: 1 - Add changed file to the staging area. 2 – Commit the file with a message. . Every commit has a **hash** which is a 40 hexadecimal character which acts like a name for the commit.

**•Commit History**

Location of where the commits exist. Commit history represented by the **objects** directory in the ‘**.git**’ directory. Every time a commit is made, it is saved in the **Commit History**.

**~~Git Text Editors: Visual Studio Code, Sublime Text, Atom~~**

**•Untracked / Tracked File**

A file not yet under version control. It has never been added to the staging area nor has it been ever committed. A Tracked file is one that is presently under version copwdntrol. Such as the file being added to the Staging area..