Git Bash commands

clear - commands line’ı təmizləmək üçün

dir & ls - hal hazırda olduğun folderin daxilindəki faylları göstərmək üçün

pwd - olduğun path ’i göstərmək üçün

cd **path** - folder’ə daxil olmaq üçün

Cd .. - olduğun yerdən bir geriyə qayıtmaq üçün

Cd ../.. - olduğun yerdən 2 addım geriyə qayıtmaq üçün

mkdir **newFolderName** - folder yaratmaq üçün

mkdir **folderName**/**newFolderName** - folder daxilində yeni folder yaratmaq üçün

Rmdir folderName - boş folder ’i silmək üçün istifadə edilir

Rm -r folderName - folder daxilində işlək vəziyyətdə olan file varsa bu folderi silmir

Rm -rf folderName - folder daxilində işlək vəziyyətdə olan file olsa belə bu folderi silir

Touch **folderName**/**file.extension** - folder daxilində hər hansısa uzantılı file yaratmaq üçün

dir-Directory

Ls -list

Cd -change directory

Mkdir - make directory

Rmdir - remove directory

Git init - folder’ə .git folderini yükləyir. Folder’i initialized edir

Git add **fileName** - folder’in daxilindəki file’ları stage’ə yükləmək üçün istifadə edilir

Git add . - folder’in daxilindəki bütün file’ları stage’ə yükləmiş oluruq ancaq bu məsləhət görülmür.

Git status - folder’in daxilindəki initialized olunmuş unstage və stage mərhələsindəki file’ları göstərir. Əgər daha öncə add,commit, push edilmiş və üzərlərində hərhansısa dəyişiklik olmuşsa o file’ların da yeni versiyalarını avtomatik unstage mərhələsinə atır və orada göstərir.

Git diff - yalnız unstage mərhələsində olan hər hansısa file daha öncə add,commit, push edilmiş və üzərlərində hərhansısa dəyişiklik olmuşsa və bu file üzərində hər hansısa bir dəyişiklik olarsa bu command vasitəsi ilə bu file’lardaki dəyişiklikləri görə bilərik. Daha sonra stage mərhələsinə add etsək bu command işə yaramayacaq

Git diff –staged - yalnız stage mərhələsindəki olan hər hansısa file daha öncə add,commit, push edilmiş və üzərlərində hərhansısa dəyişiklik olmuşsa və bu file üzərində hər hansısa bir dəyişiklik olarsa bu command vasitəsi ilə bu file’lardaki dəyişiklikləri görə bilərik. Daha sonra commit edilərsə bu command işə yaramayacaq

Git reset HEAD **fileName** - bu command vasitəsi ilə stage mərhələsindəki file’ı unstage mərhələsinə qaytara bilərik

Git commit -m “**yourMessage**” - bu command vasitəsi ilə biz local repository’ə mesaj ilə birlikdə stage’i save edirik

Git log - yalnız push edilmiş və edilməmiş commit’ləri göstərmək üçün

Git log –patch - git log’dan daha detallıdır commit’dəki file’da dəyişiklik varsa onları da göstərir

Git branch -M main -

Git remote add origin url - bu command vasitəsi ilə biz local repository’dəki save olunmuş file’ları push üçün tam hazırlayır, hansı origin’ə push edəcəyimizi qeyd edir.

Git push -u origin main - bu command vasitəsi ilə biz commit elədiyimiz file’ları origin’ə push edirik.

Git branch **newBranchName** - bu command vasitəsi ilə əgər bu adda branch mövcud deyilsə localda yeni branch yaratmış oluruq. Push etdikdən sonra bu branch də file’larla birgə origin’ə push edilir

Git checkout **branchName** - əgər branch’lərimizin arasında göstərdiyimiz branch mövcuddursa bu branch’ə keçid etmiş oluruq

Git merge **branchName** - hal hazırda olduğumuz branch’lə hər hansısa branch’i merge etmək üçün istifadə edilir.

Git clone - hər hansısa public repo’nu və ya access’imiz olan private repo’nu komputerimizə yükləmək istəyiriksə bu command’dan istifadə edirik.

Git fetch - əgər bizdən başqa kimsə bizim repo’muza nəyisə push edibsə son versiyanı gətirmək üçün istifadə edilir.

Git pull - əgər bizdən başqa kimsə bizim repo’muza nəyisə push edibsə son versiyanı gətirmək və öz file’larımıza merge etmək üçün bu command’dan istifadə edilir.