**prog alapok 3 webes hibák**

**GIT**

**fogalmak**

**repository –** raktár, könyvtár. itt tárolódik a munkánk, tárol, rendszerez, hozzáférést biztosít

**git - - version** aktuális verzió

**kezdés vscode ->terminal**

**parancsok**

**cls** törli a terminál ablakát

**mkdir mappanév** / létrehozza a project mappáját. csak akkor, ha még nincs létrehozva a mappa amiben a project van /

**cd mappanév** / belép a project mappájába – munka mappa /

**ls** kilistázza a projekt mappában levő mappákat és fájlokat

**git init** létrehoz egy helyi üres repot / .git /

**git status** a reporól ad vissza infót, pl ha bekerül új fájl akkor jelzi

**git add** filehozzáadja a színpadhoz /amíg nem commitoljuk addig itt van / a file-t, ha beírjuk az első karaktert majd TAB akkor kiegészíti több fájlt is adhatok hozzá így, nem rakja még a repoba, amíg nem coomitolom

**git commit -m** elküldi a repoba a fájlt / -m -> message /

**git add \*.txt** az összes txt kiterjesztésű fájlt a színpadra küldi

**git add .**  az összes fájlt hozzáadja

**git log** kilistázza az eddigi commitokat

**git remote add origin url** távoli repohoz csatlakozás

**git push -u origin master** a távoli repoba küldi a helyi repot, ha az -u -t megadjuk akkor legközelebb elég csak a git push -t kiadni

**git config –global user.email „email@cim”** a távoli repohoz az email cím beállítása, --global helyett –local akkor a helyi gépen

**git config –global user.name „nev”** ezen a néven jelennek meg a repoid

**kijelentkezéskor:** git config --global – unset user.name „név”

git config --global –unset user.email „email”

**git pull origin master** lekéri a távoli repot

**git pull origin master - -allow-unreleated-histories** ha nem akar pullolni akkor használjuk

**git clone url** ha már van távoli repo de nincs localis

Ha már valaki PUSHolt a remote repo felé, akkor a többieknek muszály PULLolnia mielőtt PUSHolni tudna!

**git diff HEAD** megmutatja hogy az utolsó commitunk óta milyen fájlok változtak a remote repon

**git diff fájlnév** az adott fájl változásait mutatja

**git diff - -staged** a színpadon /stage/ levő fájlokról is megmutatja a változásokat

**git reset fájlnév** leveszi a színpadról a fájlt

**git checkout - -fájlnév** visszaállítja fájt a módosítás előttire, ne használjuk 😊

**git checkout XXXX** a commit első négy karaktere, visszaállítja a verzióra

**git branch branch\_neve** új branch létrehozása

**git branch** megmutatja melyik branchba vagyunk

**git checkout branch\_neve** átvált a branch\_neve branchra

**git rm „\*txt”** törli az összes .txt-t csak abból a branchból ahol vagyunk

**git merge branch\_neve**  összefésüli az adott branchet a branch\_neve branchhel

**git branch -d branch\_neve** törli a branchet

**gitk** grafikus felület

<https://www.brainextension.hu/2014/08/22/ujabb-git-parancssori-okossagok-addrmresetlog/>

<https://www.youtube.com/playlist?list=PLyriihBWoulw1KQ2aM-NMnAoVPZevfmZq>