Git-logg, Toni Halmetoja

Grupp 12

Allmänt arbetsflöde

Först klonade jag ner vårt repo med *git clone* och addressen.

Sedan skapade jag en branch med *git branch [namn]*, och bytte till den med *git checkout*, något som jag upprepade för varje gång som jag gjorde ändringar. Alltså skapade jag en branch och gjorde ändringar i denna istället för i main.

Efter att jag gjort några ändringar i min branch, med namn Toni, så commitade jag den med *git commit -a -m ”meddelande”*. -a för ”all”, och ”-m” för att kunna skriva meddelandet i commandline istället för via webbsidan.

Sedan pushade jag upp min branch med *git push --set-upstream origin [namn]* för att skapa en remote branch.

Därefter körde jag en *merge* med main. Inga konflikter uppstod.

Jag märkte nästa dag att andra gruppmedlemmar hade gjort ändringar, så jag uppdaterade mitt repo med *git pull origin [main].* Just *pull* upprepade jag varje gång innan jag skapade en ny branch.

Jag började göra ändringar i main som en klant, och körde en *git switch -c [namn]* när jag märkte det, för att flytta ändringarna till en ny branch innan de committades.

Jag gjorde en *pull request* via webbsidan, och tog bort min ursprungliga branch

---

Denna process upprepades under arbetstiden tills en konflikt uppstod. Då valde jag de ändringar jag ville behålla i VSC, lade till filen i fråga, gjorde en *commit* av ändringarna, och körde sedan en *merge*.

Övriga kommandon som användes

Git clone för att skapa ett lokalt repo från remoten.

Git status för att se status på repot.

Git checkout [namn] för att byta branch.

Git add [filnamn] för att lägga till specifika filer.

Git add . för att lägga till samtliga filer.

Git branch för att se alla tillgängliga brancher.

Git branch [namn] för att byta aktiv branch.

Git commit -m för att committa med meddelande.

Git merge för att slå ihop ändringar i filerna.

Git switch för att byta branch men behålla ändringar.