

Bonusuppgift: Bra användning av Git (4 poäng)

Uppgiften bedöms individuellt. Uppgiften behöver påbörjas så fort arbetet med projektet påbörjas. Git ska användas löpande under hela projektet. Börjar du använda Git fyra veckor in i projektet går det inte att få dessa bonuspoäng. När koden ska lämnas in ska de medlemmar som få denna uppgift bedömd listas i samma mejl som kodinlämningen.

Tanken med bonusuppgiften är att uppvisa en god förståelse för hur Git används: både hur det tekniskt fungerar och vad ett bra användande av Git innebär. Följande ska uppfyllas för att uppgiften ska bli godkänd:

- Aktivt användande av Git så fort arbetet med projektet påbörjas.
- Arbete ska utföras på branches som sedan merge:as till master. (Din branch kan antingen vara personlig eller skapad efter feature som arbetas på)
- Tydligt namn på *varje* branch du skapar. Namnet ska indikera vad branchen innehåller.
- Dina commit-meddelanden ska vara tydliga. En utomstående ska förstå vad du tillfört med din commit.
- Arbetet med Git ska utföras genom kommandon i terminalen. (Arbetet med Git får absolut inte utföras i det grafiska gränssnittet på Gitlabs hemsida.)

Tips

Här är några tips för att jobba effektivt med Git.

- **Gör små isolerade commits:** Varje commit bör vara liten och innehålla lösningen på ett specifikt problem. Om flera orelaterade ändringar har gjorts bör dessa läggas som separata commits. Små commits gör det lättare för andra i gruppen att förstå ändringarna du har gjort och det blir lättare att lösa eventuella merge-konflikter som uppstår. Det är också lättare att skriva ett tydligt commit-meddelande om alla ändringar relaterar till samma problem.
- **Gör commits ofta:** Undvik att göra en stor commit vid slutet av dagen. Gör istället en commit varje gång du avslutat en uppgift eller löst ett problem. Genom att göra commits ofta kommer varje commit bli mindre och innehålla mer relaterade ändringar. Du slipper också separera flera ändringar i samma fil till flera commits¹.
- **Inkludera delar av en fil i en commit:** En commit behöver inte innehålla alla ändringar i alla filer. Det går bra att välja vissa filer eller vissa delar av filer i en commit¹. Detta gör det lätt att plocka ut de relaterade delarna av koden och skapa commits även om du glömt checka in ditt arbete på ett tag.
- **Jobba effektivt med branches:** Jobba alltid i en branch som inte är main. Se till att koden i main alltid är komplett och merga in ändringar till main från din branch först när de är fungerande och testade. Om du jobbar i en personlig branch, se till att din branch är up-to-date med det som finns i main. Det är tråkigt att jobba med kod som det senare visar sig är utdaterad.

¹git add -i, Se <https://git-scm.com/book/en/v2/Git-Tools-Interactive-Staging>