

GIT VERSIONSHANTERING - GRUND

ÖVNING I CSS

I denna övning ska vi hantera våra filer med versionshanteringssystemet GIT. Vi kommer nu att installera git, och initiera ett git repository på din dator, som du sedan kommer att synkronisera med ditt repository på Github. Om du använder Microsoft Visual Code kan du hantera din Git genom tillägget (extension) GitLens.

Observera att git kan hanteras direkt via terminalfönster, eller via så kallade gränssnitt där man klickar och fyller i information. Att kunna grundläggande git kommandon i konsol, terminal är en stor fördel. Verktyg som exempelvis GitLens eller SmartGit används för att effektivisera sitt arbetsflöde och få en bättre överblick över sitt projekt.

Den bifogade lathunden är bra att ha i denna övning.

DOKUMENTATION & REFERENS

- ✓ <https://git-scm.com/downloads> <-- Programvara för att hantera Git på din dator
- ✓ <https://www.syntevo.com/> <-- Populärt verktyg som används för Git på dator.
- ✓ <https://github.com/> <-- En av de populäraste tjänsterna för Git.
- ✓ <https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud>
- ✓ <https://help.github.com/>
- ✓ <https://git-scm.com/book/en/v2/Getting-Started-Git-Basics>

EN GIT REPOSITORY

Börja med att ladda ned och installera Git programvara för ditt operativsystem från Git-Scm (se länk ovan). Öppna sedan Github (se länk ovan) i din webbläsare och registrera ett konto med din epost-adress.

I din dator har du en mapp där din webbplats finns (index.html, style.css, bilder med mera). Se till att du har en klar struktur på dina filer och mappar. Om installationen av Git gick bra, bör du nu kunna köra Git-kommandon i terminalfönster.

Det finns flera sätt att köra terminal i din mapp för webbplatsen, ett sätt är att använda MS Visual Code så som demonstreras i föreläsningen, eller genom att öppna *Power Shell* i Windows (I utforskaren kan du klicka på Arkiv-Öppna Power Shell – Kör som administratör). Om du använder Power Shell / Mac Terminal är det viktigt att du *står* i den mapp där din webbplats finns.

För att initiera ett repository för Git i mappen för din webbplats, kör du kommandot:

```
git init .
```

Häriifrån kan du välja att hantera ditt *repository* via MS Visual Codes GitLens, SmartGit, eller via terminalfönstret.

För att se status på ditt *repository* kör du följande kommando:

git status

Git kommer då att rapportera vilka filer som finns, och vilka som ej ännu lagts till för bevakning i git. Vi ska nu lägga till din webbplats i gits bevakningsfunktion (tracking). För att lägga till alla filer i ditt projekt kör du följande kommando:

git add .

Git lägger då till alla dina filer för bevakning av ändringar i dem. Kontrollera genom att köra *git status* igen.

Nu ska vi göra en första *commit* som är sparad tidpunkt i tidslinjen för ditt projekt, vi skapar här en så kallad *snapshot*. För att göra en *commit* kör du följande kommando:

git commit -m "Beskrivning av åtgärd du gjort i ditt projekt, ex. lagt till en navigeringsmeny"

Du har nu skapat en slags "återställnings-punkt" i ditt projekt som du kan använda för att återställa dina filer så som de var vid detta tillfälle. I nästa steg ska vi synkronisera dina ändringar med ditt centrala repository på Github.

GITHUB

Observera att du behöver ha sett föreläsningen *Verifiera dig med SSH-nycklar på GitHub*, den hittar du på Play-tjänsten i samma spellista som föreläsningen till denna övning.

När du registrerat ett konto på Github, behöver du logga in och skapa ett repository. Du skapar ett genom att klicka på *New repository* och ange ett namn samt en beskrivning för ditt repository. Välj här att använda MIT-license, välj att inte skapa en *.gitignore*.

När du skapat ditt repository är det dags att synkronisera detta med ditt lokala repository. Öppna ditt repository på Github och klicka på den gröna knappen med texten "*clone or download*" och kopiera den URL som visas för SSH.

DITT LOKALA REPOSITORY

Prova nu att lägga till ditt eget repository på Github som en *remote* (kommando: *git remote add -f origin <url>*) i ditt lokala repository.

Om du lyckades med att skapa SSH-nycklar och verifiera dig hos Github, ska du nu kunna synkronisera ditt lokala repository med ditt repository på Github.

För att synkronisera, behöver du först köra kommandot:

git pull origin master

Om du får ett felmeddelande av något slag, kan du även testa att köra:

git fetch origin master

Du bör nu ha synkat ned data från ditt repository på Github. I nästa steg ska vi skicka data.

SYNKA TILL GITHUB

För att du ska kunna synkronisera filer och data med Github kan du antingen använda de förenklade verktygen i SmartGit, eller i Microsofts Visual Code för git. Denna övning använder endast terminal-kommandon. Vi ska nu synkronisera dina *repositories*.

Extra hjälp: <https://www.atlassian.com/git/tutorials/syncing/git-push>

För att skicka och synkronisera data, kör du följande kommando:

```
git push origin master
```

Om allting fungerar korrekt och du konfigurerat dina SSH-nycklar, bör dina filer nu synkroniserats med ditt repository på Github.

Kom ihåg att du måste lägga till alla ändringar du gör i filer innan du kör *'git push'* till din centrala repository. De tre vanligaste kommandona när man använder git är oftast:

- `git add .` <-- *Lägger till alla filer och ändringar, du kan även ersätta punkten med ett filnamn.*
- `git commit -m "Beskrivning av vad du gjort i detta snapshot"` <-- *För att göra ett snapshot i ditt projekt.*
- `git push origin master` <-- *Synkroniserar alla ändringar till din Master-branch med din Remote som heter Origin (som innehåller url till ditt repository på Github).*

Om allting är korrekt konfigurerat, så har du nu synkroniserat dina filer till Github, och du ska kunna se dem i ditt repository på Github, tillsammans med ändringshistorik.