**Hjälp för Git/Github**

**Nytt projekt**

* Skapa ett repository på gitHub.
  + Ge det ett namn.
  + Ge det en beskrivning.
  + Låt det vara public
  + Kryssa i för readme-fil
  + Klicka sedan på knappen för att skapa ditt repository
* Starta gitbash på din lokala dator.
* Skriv **pwd** för att se var du är i fil-hiearkin.
* Byt till mappen **projects (cd projects)**
* Klona ner ditt repository genom att skriva:  
    
  **git clone + adressen**   
    
  Adressen är den nu kopierar från gitHub.  
    
  Nu skapas ett lokalt repository som är en kopia av ditt repository på gitHub.
* Kolla att din nya mapp skapats:  
  Skriv: **ls**
* För att byta till den mapp ni skapade (klonade) så skriver ni:  
  **cd** <namn på mapp>  
  t.ex. cd HTML
* När ni är i er nyskapade mapp kan ni kolla igen att vilka filer där finns.  
  **ls**Den enda fil ni ser är readme.

**Arbetsgång när ni jobbar med projekt**

* Skriv **git status**
  + Röda filer kan inte ”ses” av git och kan inte commitas.
  + För att lägga till filer i ”staging area”:
    - En fil: git add filnamn.
    - Alla icke tillagda filer. Git add . (git add punkt)
  + Skriv alltid git status igen för att se att filerna är gröna.
* När ni lagt till de filer ni vill commita är det dags att göra commit:
  + git commit -m ”beskriv ändring ni gjort”
* Nu har vi skapat en ”ögonblicksbild” på din lokala dator. Nu ska vi skicka detta till gitHub.
* För att först vara säkra på att vårt lokala repository är ”up to date” ska vi först ”dra ner” de filer vi har på gitHub.
  + Skriv: **git pull origin master**
* Sedan ska vi skicka upp våra ändringar till gitHub:
  + Skriv: **git push origin master**

**Git- kommandon:**

* **git** **status** – Kollar statusen på ditt repo. Vilka filer som finns där, vilka som har commitats och vilken gren (branch) som du arbetar på just nu.
* **git log** – Se alla commits
* **git** **add** - lägger *inte* nya filer i ditt repo. Istället gör det Git uppmärksam på att det har kommit nya filer i ditt repo.
  + När en fil inte har gits uppmärksamhet kommer den inte med i en commit. Filen räknas då som *untracked.*
  + Git add gör att filen kommer med när du gör commit.
* **git push** – ”pushar” lokala ändringar till gitHub.
* **git pull** – hämtar de senaste filerna från gitHub
* **git reset HEAD <filnamn>** - tar bort filer som lagts till med add från ”staging area” så de åter är ”untracked” (röda). Ifall man ångrar sig och inte vill ha med det i sin commit.

**Icke git-kommandon**

* **pwd** – se var du är i filhierarkin.
* **mkdir** – skapa ny mapp.
* **cd <mappnamn>**– byt mapp.
* **cd** .. – gå upp ett steg.
* **ls** – lista filer/mappar
* **clear** – rensa fönstret