

git introduktion

Del 3 av 4

<https://github.com/Rnqst>

git-kursen schema

- 9:00 – 9:45
- Rast
- 10:00 – 10:45
- Rast
- 11:00 – 12:00

Vad gör ett versionshanteringssystem?



Termer

- Master, main, trunk, stam, rot
- Gren, branch, fork....
- Revision, version

Bilden på föregående sida från:

<https://git-scm.com/about>

Miss inte boken

- Läs boken Pro Git som du antingen kan beställa som en tryckt bok eller ladda hem
- <https://git-scm.com/book/en/v2>
- Den finns översatt till 30-talet språk
- Dock är inte alla översättningar klara

Git från kommandoraden

Den som behöver styra git mer precist använder ofta git från kommandoraden.

Om du vill kan du installera från:

<https://git-scm.com/downloads>

Detta är inte nödvändigt för kursen här.

Var är vi nu?

- Alla har konton på github
- Alla har GitHub Desktop
- De som vill har installerat git för terminalfönster
- Alla vet var de kan läsa boken Pro Git
- Alla har börjat använda git
- Några har grenat av stammen
- Förhoppningsvis har somliga även grenat tillbaks till stammen igen

Självstudierna

Hur gick övningarna?

Frågorna?

Övningar från i tisdags

Övning 1

- Vi jobbar i desktop-tutorial
- Skapa en fil, **MinFil.txt**, skriv in minst fem rader text.
- Skriv in en förklarande kommentar
- Kör **commit**

Övningar från i tisdags

Övning 2

- Vi jobbar i desktop-tutorial
- Ändra rad tre i filen **MinFil.txt**.
- Skriv in en förklarande kommentar
- Kör **commit**
- Ladda upp filerna med **push**
- Hur ser det ut i din webbläsare på **github.com**?

Övningar från i tisdags

Övning 3

- Vi jobbar i desktop-tutorial
- Skapa en ny gren, döpt den till **laboration**
- Vad händer då? Finns filerna kvar?
- Lägg till en ny fil
- Lägg till två nya rader i mitten av filen **MinFil.txt**
- Kör commit och push, vettiga kommentarer...
- Vad har hänt på github nu?

Övningar från i tisdags

Fördjupningsövningar

Fördjupning 1

- Vi jobbar i desktop-tutorial, i grenen **laboration**
- Du vill kopiera ändringarna i din gren till stammen (**main**).
- Hur gör du det? Finns det flera sätt?

Läs **Pro Git**.

Övningar från i tisdags

Fördjupning 2

- Det här är för dem som vill lära sig git-kommandon, har du svårt att hänga med i hur du använder GitHub Desktop, vänta med denna uppgiften till nästa lektion.
- Hur gör du för att lägga upp ett nytt projekt i git?
- Tänk dig att du redan har en katalog med några filer och utgå från den.
- Vilka kommandon behöver du köra för att allt skall hamna i git lokalt på din maskin? Vad behöver du ytterligare göra för att allt även skall hamna på github?

Läs **Pro Git**.

Självstudier från i tisdags

Självstudier

Hemarbete 1

- Vad menas med att en fil är ”**ospårad**” eller ”**untracked**”?
- Vad gör **git clean**?
- Vad är en ”**prepareringsyta**” eller ”**stage area**”?
- Hur kan du ställa in så att git inte bryr sig om vissa filer?
- Läs kapitel 5 om ”Distributed Git” i boken Pro Git.

Självstudier från i tisdags

Hemarbete 2

- 1. Jobba i desktop-tutorial med GitHub Desktop. Gör en gren, branch, lägg till ett par filer där. Hur för du då tillbaka dina nya filer till stammen, main?
- 2. Prova att göra en gren, gå till main och editera och commita en fil där som även finns i stammen. Vad händer nu när du för tillbaka den nya grenen till stammen?
- 3. Gör som under två, men innan du för tillbaka grenen till stammen så går du till stammen och editerar stammens version av den filen du editerade i grenen, men ändra den på ett annat sätt än du gjorde i grenen. Gå sedan till grenen och försök föra tillbaks den till stammen, vad händer då?

Lägga till filer

Ospårade filer, untracked

Normalt när du lägger till en fil i en katalog så hamnar filen där.

Vill du ha in den i git kör du:

```
git add <filnamn>
```

GitHub Desktop kör "git add" automatiskt när den ser en ny fil i en katalog som hör till ett projekt.

Hur skapar vi ett nytt repo?

I GitHub Desktop kan du antingen köra **CTRL+N** eller använda menyerna: **File > New Repository**

Med kommandon kör du:

```
git init
```

Då skapas de filer som git behövs för att hålla rätt på filerna i den katalog du står i och underliggande filträd.

Knyta ihop påsen, göra klart

`git commit`

När du arbetat färdigt med filerna så körs commit för att lägga till dina uppdateringar (från prepareringsytan, staging area, till lokalt repo).

`git push`

Laddar sedan upp ditt repo till en server.

GitHub Desktop, hur gör du där?

Övningar

Vi jobbar i desktop-tutorial

- Skapa en fil, **MinFil.txt**, skriv in minst fem rader text.
- Skriv in en förklarande kommentar
- Kör **commit**

Övningar

Vi jobbar i desktop-tutorial

- Ändra rad tre i filen **MinFil.txt**.
- Skriv in en förklarande kommentar
- Kör **commit**
- Ladda upp filerna med **push**
- Hur ser det ut i din webbläsare på **github.com**?

Övningar

Vi jobbar i desktop-tutorial

- Skapa en ny gren, döpt den till **laboration**
- Vad händer då? Finns filerna kvar?
- Lägg till en ny fil
- Lägg till två nya rader i mitten av filen **MinFil.txt**
- Kör commit och push, vettiga kommentarer...
- Vad har hänt på github nu?

Fördjupningsövning

Vi jobbar i desktop-tutorial, i grenen **laboration**

Du vill kopiera ändringarna i din gren till stammen (**main**).

Hur gör du det? Finns det flera sätt?

Läs **Pro Git**.

Självstudier

Vad menas med att en fil är ”**ospårad**” eller ”**untracked**”?

Vad gör **git clean**?

Vad är en ”**prepareringsyta**” eller ”**stage area**”?

Hur kan du ställa in så att git inte bryr sig om vissa filer?

Ta bort filer

```
git rm <filnamn>
```

I linux heter kommandot för att ta bort en fil `rm` (förkortning av remove).

Hur gör vi i GitHub Desktop?

Flytta filer

```
git mv <filnamn> <ny plats>
```

I linux heter kommandot för att flytta en fil `mv` (förkortning av move).

Hur gör vi i GitHub Desktop?

Ångra dina ändringar

Om du ändrat filen min.fil men kommer på att det var dumt kan du köra:

```
git checkout - min.fil
```

Dessa förändringar sker här i din beredningsyta, stage.

Hur gör vi i GitHub Desktop?

git kommandon

- Skapa ett "repo" med **git init**
- Lägg till filer med **git add <fil>**
- Hämta information om ditt arbete med
- **git status**
- **git log**

git kommandon

- Markera att du arbetat färdigt med **git commit**
- Hämta kod med **git clone**
- `git remote add origin https://github.com/<användare>/<project>`

git kommandon

- Ladda upp ditt arbete med **git push**
- Tanka hem dina kollegor med **git pull**

Stageing area

- De filer som väntar på att behandlas av git finns i "stageing area"
- Ändrar du i en fil kan du vara tvungen att lägga till den med "git add"
- https://www.youtube.com/watch?v=SWYqp7iY_Tc

Sammanfattning

- Vad är git?
- Revisionshantering vs Versionshantering?
- Dokumenthantering?

Sammanfattning

- Behövs en speciell git-server?
- När du checkat ut koden med "`git clone...`", vad finns i din kopia då?
- Bara senaste versionen?

Övningar och självstudier