

git introduktion

Del 4 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?



Hur kommer vi vidare?

Hur arbetar vi med versionsnummer i git?

- git sätter egna versionsnummer
- git tag ...
- git describe ...
- Skapa en ny gren med passande namn

Bilden på föregående sida från:

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

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
- Vi har grenat av stammen
- Många har även grenat tillbaks till stammen igen

Självstudierna

Hur gick övningarna?

Frågorna?

Självstudierna

- Igår gjorde de flesta en gren i sina övningsprojekt. Om du inte gjort en gren ännu, gör det nu, om du har svårt att hitta på ett namn, kalla den för laboration.
- Du kan välja gren genom att trycka på mittenknappen i det svarta knappfältet med tre knappar längst upp i GitHub Desktop fönstret.
- Skapa en fil i main, kör commit på den och se om den finns i din laborationsgren. Likadant i din labbgren, skapa filer där och se om de dyker upp i main.
- Se hur du skall göra för att kopiera de nya filerna från main till laboration, se även hur du kopierar dem åt andra hållet.

Självstudierna

- Editera en fil i main på ett sätt och editera samma fil i laboration, vad händer med dina förändringar när du kopierar mellan grenarna? Vad händer om dina ändringar är på samma rad?
- Grena av laborationsgrenen till en tredje gren, kalla den test. Ändra i någon fil där som finns i både main och laboration, pröva att kopiera från test till laboration. Händer någonting i main då?
- Ändra på en annan fil i test och kopiera från test till main.

Självstudierna

- Uppdatera laboration från main.
- Slå ihop test med laboration.
- Hur gör du för att slå ihop din laborationsgren med main?

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**?

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

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