

Introduktion till git och github (för statistiker)

Måns Magnusson

2016

Overview

Versionshantering

Git

Github, GitLab, BitBucket ...

Grunderna i git

Syfte

- ▶ Förstå modern versionshantering

Syfte

- ▶ Förstå modern versionshantering
- ▶ Komma igång med git och github

Syfte

- ▶ Förstå modern versionshantering
- ▶ Komma igång med git och github
- ▶ Skapa grund för att:
 - ▶ Använda git i “skarpt läge”

Syfte

- ▶ Förstå modern versionshantering
- ▶ Komma igång med git och github
- ▶ Skapa grund för att:
 - ▶ Använda git i “skarpt läge”
 - ▶ Egen fördjupning vid behov

Vad är versionshantering?

[What is version control?]

Varför versionhantera?

- ▶ Samarbete i samma projekt

Varför versionhantera?

- ▶ Samarbete i samma projekt
- ▶ Spara olika versioner

Varför versionhantera?

- ▶ Samarbete i samma projekt
- ▶ Spara olika versioner
- ▶ Återställa tidigare versioner

Varför versionhantera?

- ▶ Samarbete i samma projekt
- ▶ Spara olika versioner
- ▶ Återställa tidigare versioner
- ▶ Spårbarhet (kod och data)

Varför versionhantera?

- ▶ Samarbete i samma projekt
- ▶ Spara olika versioner
- ▶ Återställa tidigare versioner
- ▶ Spårbarhet (kod och data)
- ▶ Kodsäkerhet - best practice vid vetenskaplig programmering

Best practice for scientific computing

Best practice for scientific computing (Wilson et al., 2014):

Best practice for scientific computing

Best practice for scientific computing (Wilson et al., 2014):

- ▶ use a version control system (3b)

Best practice for scientific computing

Best practice for scientific computing (Wilson et al., 2014):

- ▶ use a version control system (3b)
- ▶ put everything that has been created manually in version control (3c)

Best practice for scientific computing

Best practice for scientific computing (Wilson et al., 2014):

- ▶ use a version control system (3b)
- ▶ put everything that has been created manually in version control (3c)
- ▶ use pre-merge code reviews (8a)

Best practice for scientific computing

Best practice for scientific computing (Wilson et al., 2014):

- ▶ use a version control system (3b)
- ▶ put everything that has been created manually in version control (3c)
- ▶ use pre-merge code reviews (8a)
- ▶ use an issue tracking tool (8c)

Vad är git?

- ▶ Versionshanteringssystem

Vad är git?

- ▶ Versionshanteringssystem
- ▶ Distribuerat och lokalt

Vad är git?

- ▶ Versionshanteringssystem
- ▶ Distribuerat och lokalt
- ▶ Kommandoradbaserat men...

Vad är git?

- ▶ Versionshanteringssystem
- ▶ Distribuerat och lokalt
- ▶ Kommandoradbaserat men...
- ▶ SourceTree (eller R-Studio) gör livet lättare

Vad är git?

- ▶ Versionshanteringssystem
- ▶ Distribuerat och lokalt
- ▶ Kommandoradbaserat men...
- ▶ SourceTree (eller R-Studio) gör livet lättare
- ▶ Skapades 2005 för Linux

Vad är git?

- ▶ Versionshanteringssystem
- ▶ Distribuerat och lokalt
- ▶ Kommandoradbaserat men...
- ▶ SourceTree (eller R-Studio) gör livet lättare
- ▶ Skapades 2005 för Linux
- ▶ Passar stora och små projekt

Vad är git?

- ▶ Versionshanteringssystem
- ▶ Distribuerat och lokalt
- ▶ Kommandoradbaserat men...
- ▶ SourceTree (eller R-Studio) gör livet lättare
- ▶ Skapades 2005 för Linux
- ▶ Passar stora och små projekt
- ▶ Standard för dataanalys och statistisk programmering (?)

Varför git?

- ▶ (Relativt) enkelt att börja använda

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas isolerat av enskilda medarbetare

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas isolerat av enskilda medarbetare
- ▶ Öppen källkod

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas isolerat av enskilda medarbetare
- ▶ Öppen källkod
- ▶ Distribuerat

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas isolerat av enskilda medarbetare
- ▶ Öppen källkod
- ▶ Distribuerat
- ▶ Snabbt

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas isolerat av enskilda medarbetare
- ▶ Öppen källkod
- ▶ Distribuerat
- ▶ Snabbt
- ▶ Säkert

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas isolerat av enskilda medarbetare
- ▶ Öppen källkod
- ▶ Distribuerat
- ▶ Snabbt
- ▶ Säkert
- ▶ Standard i R-världen - hantering av R-paket på github

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas isolerat av enskilda medarbetare
- ▶ Öppen källkod
- ▶ Distribuerat
- ▶ Snabbt
- ▶ Säkert
- ▶ Standard i R-världen - hantering av R-paket på github
- ▶ Bas i modern (statistisk) mjukvaruutveckling

Vad är git remote?

- ▶ Central lagring av versionshanterad kod, data m.m.

Vad är git remote?

- ▶ Central lagring av versionshanterad kod, data m.m.
- ▶ github, GitLab, BitBucket...

Vad är git remote?

- ▶ Central lagring av versionshanterad kod, data m.m.
- ▶ github, GitLab, BitBucket...
- ▶ Elementära hemsidor för projekt

Vad är git remote?

- ▶ Central lagring av versionshanterad kod, data m.m.
- ▶ github, GitLab, BitBucket...
- ▶ Elementära hemsidor för projekt
- ▶ Issue trackers / Wikis / Diskussionsmöjligheter

Varför git remote?

- ▶ Underlättar samarbeten (kraftigt)

Varför git remote?

- ▶ Underlättar samarbeten (kraftigt)
- ▶ Distribuera kode mellan datorer

Varför git remote?

- ▶ Underlättar samarbeten (kraftigt)
- ▶ Distribuera koden mellan datorer
- ▶ Säkerhet och backup

Varför git remote?

- ▶ Underlättar samarbeten (kraftigt)
- ▶ Distribuera kode mellan datorer
- ▶ Säkerhet och backup
- ▶ Administrera projekt

Varför git remote?

- ▶ Underlättar samarbeten (kraftigt)
- ▶ Distribuera koda mellan datorer
- ▶ Säkerhet och backup
- ▶ Administrera projekt
- ▶ The burn it all down workflow

The burn it all down workflow

The amount of git skills to fix a borked up git repo is order of magnitudes larger than to bork it - Jennifer Bryan

Section 4

Grunderna i git

Acknowledgments

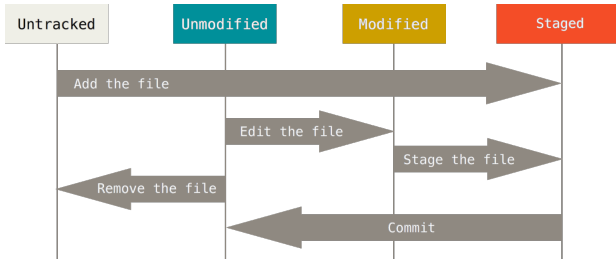
Bilder är tagna från:

- ▶ The git book [\[här\]](#)
- ▶ Atlassian git tutorial [\[här\]](#)

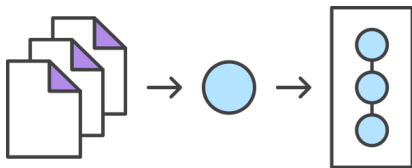
Repository



Arbetsflödet i git

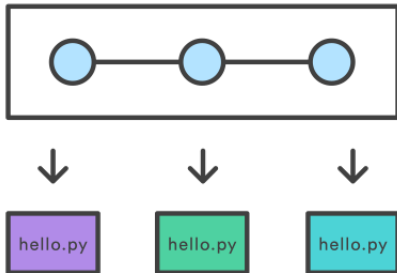


Commit



Commit

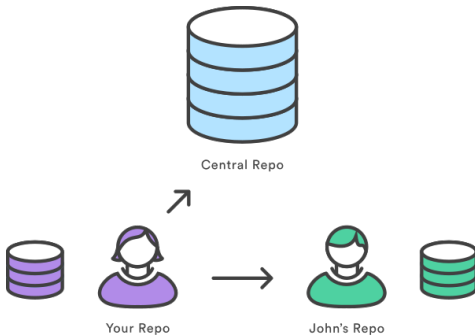
Recording Snapshots (Git)



Commit message

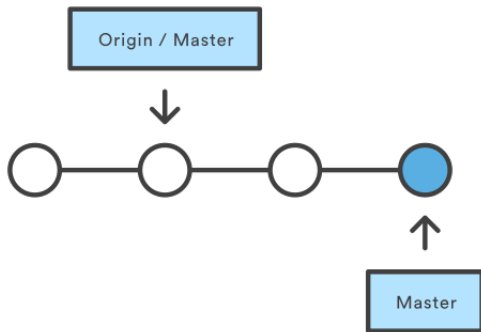
If applied, this commit will [your commit message here]

Remote



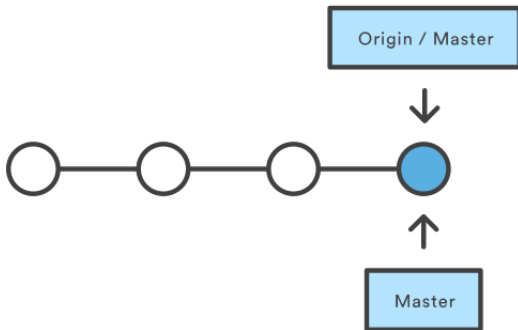
Local till remote - Push

Before Pushing

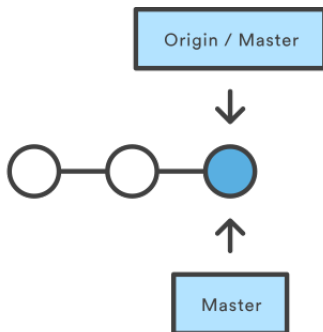


Local till remote - Push

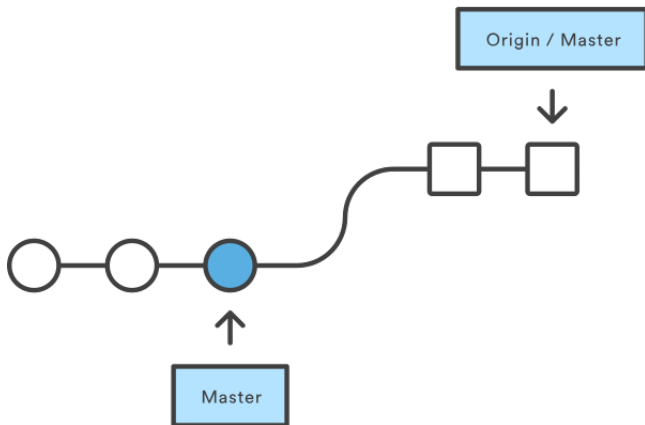
After Pushing



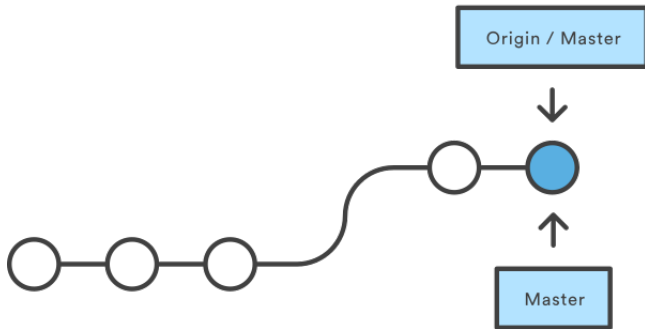
Remote till local - Pull



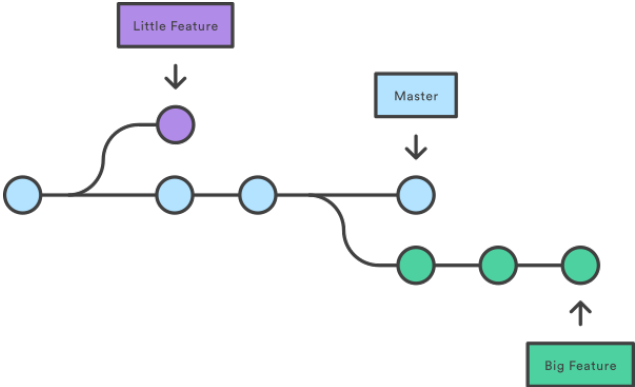
Remote till local - Pull



Remote till local - Pull

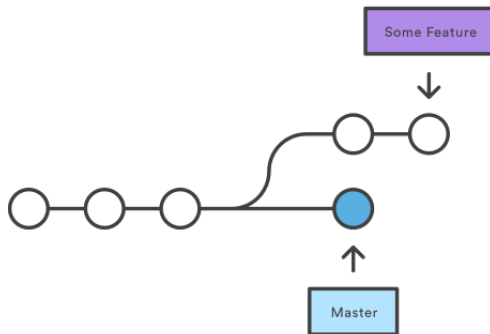


Branch och merge



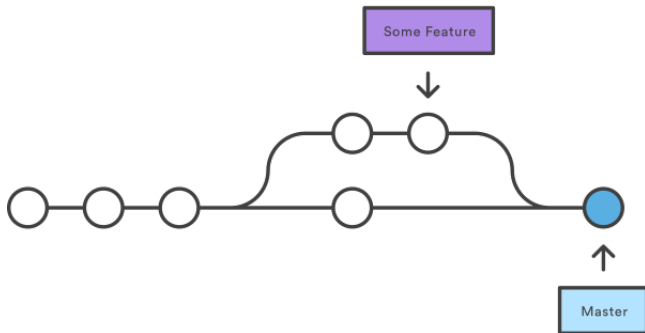
Branch och merge

Before Merging



Branch and merge

After a 3-way Merge



References

Wilson, G., Aruliah, D., Brown, C. T., Hong, N. P. C., Davis, M., Guy, R. T., Haddock, S. H., Huff, K. D., Mitchell, I. M., Plumbley, M. D., et al., 2014. Best practices for scientific computing. PLoS Biol 12 (1), e1001745.