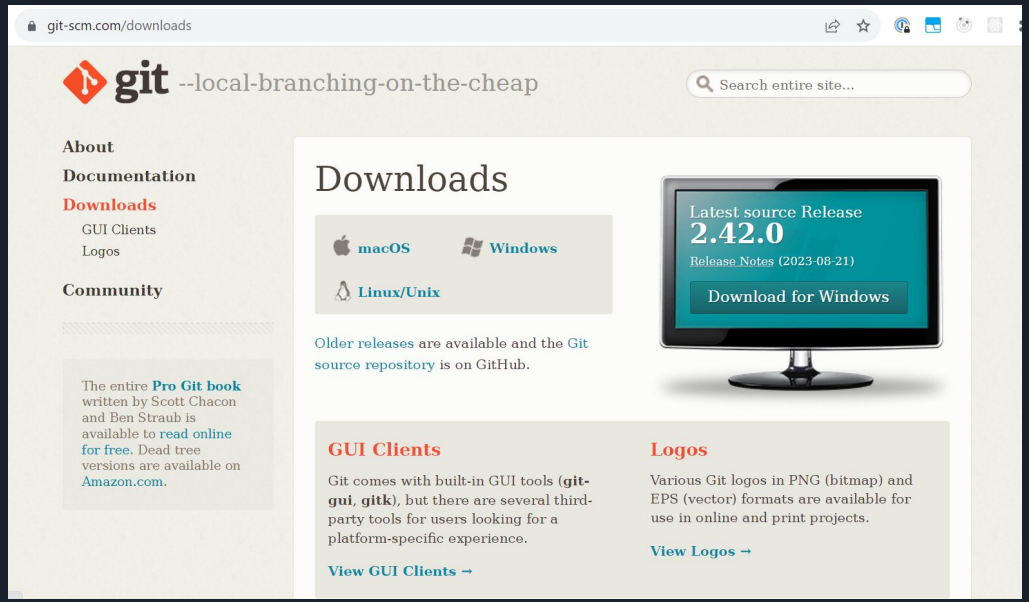
Git 101

Agenda

- Installera git
- Vad är git?
- Hur håller git reda på förändringa?
- Working directory tillstånd
- Hur fungerar commits?
- Ångra saker i git…
- Övningar och frågor
- Läxa genomgång

Installera git

https://git-scm.com/downloads



Kolla att du har git installerat

> git –version

Konfigurera git

> git config —global user.name “PatrikNygren”

> git config —global user.email “keyturnlogic@gmail.com”

Vad är git?

Git är ett distribuerat versionshanteringssystem

Git hjälper oss

- Hålla ordning på förändringar i en mapp (ett projekt)
- Kan rulla tillbaka till tidigare förändringar
- Vi kan skapa “branches” för att experimentera
- Samarbeta med andra

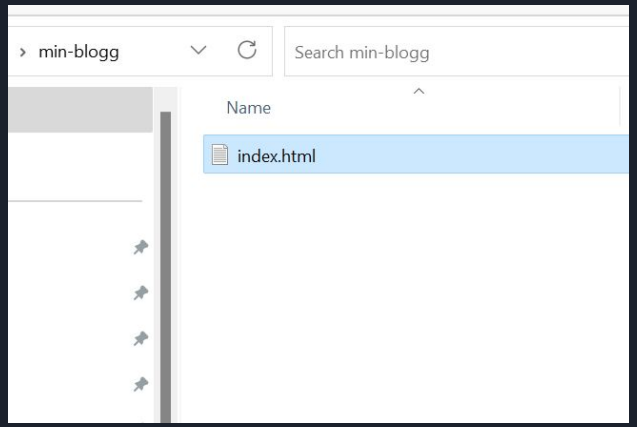
Vad är git?

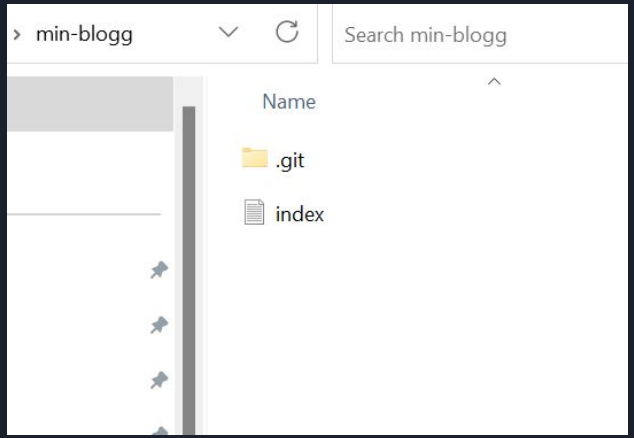
> mkdir min-blogg (skapar en mapp)

> cd min-blogg (gå in i mappen)

> code index.html (skapa en fil i mappen)

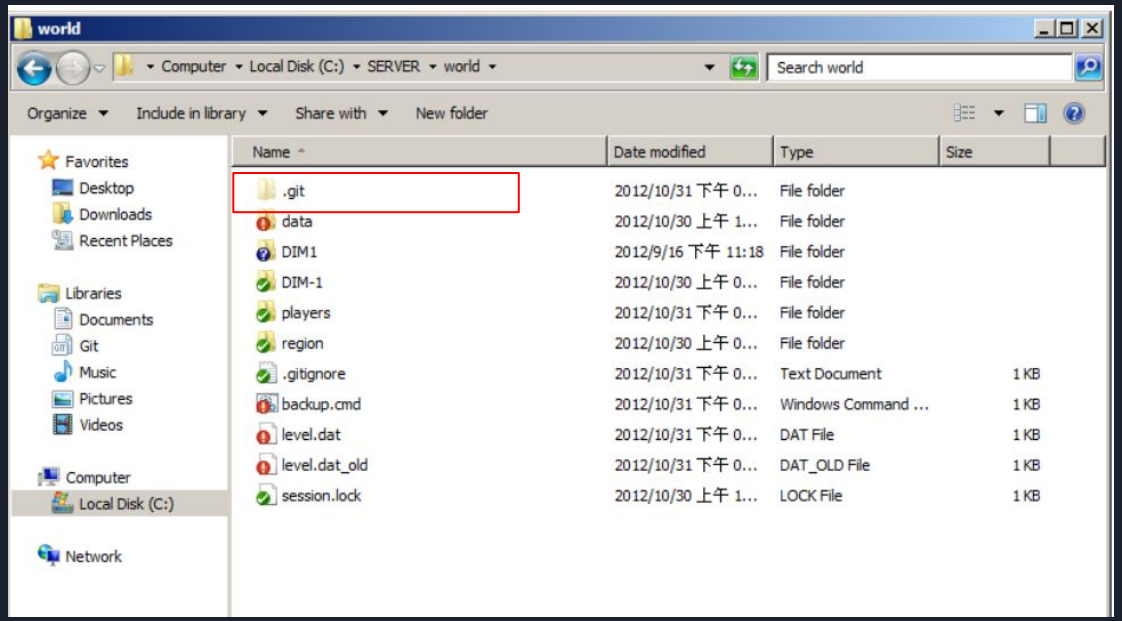
> git init





Vad är git?

När git initialiseras i en mapp skapas ett repository i mappen (databas)



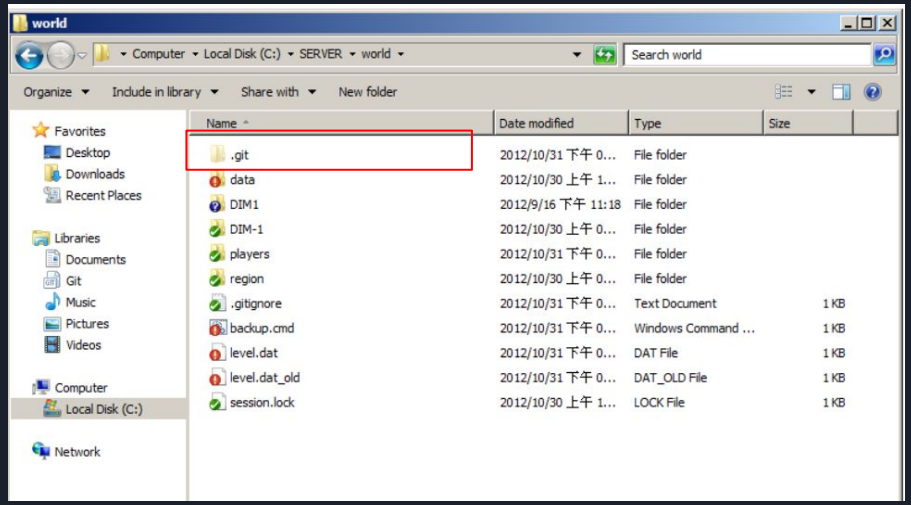
Vad är git?

Denna git mapp håller reda på förändringar

- > git init

Detta kommando initialiserar mappen

- Vi kallar .git-mappen för ett repository



Hur håller git reda på förändringar?

- Git har olika koncept för förändringsarbetet

> git status

Berättar vilka förändringar som är i staging area och inte

> git add index.html style.css

Bestämmer vilka filer med förändringar som ska läggas på “staging area”

> git commit -m “min första commit i git”

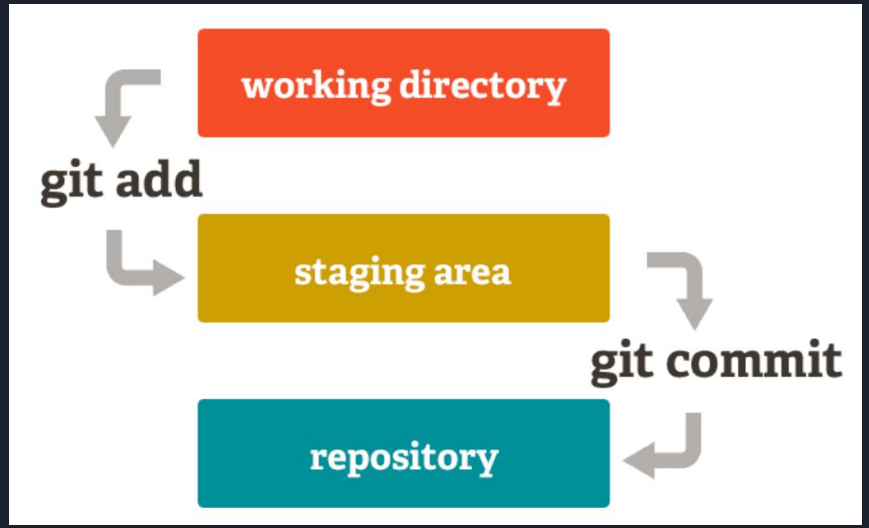
Nu sparas alla förändringar som gjorts i en commit

Hur håller git reda på förändringar?

Demo

1. Skapa en tom mapp med en fil
2. Initialisera mappen till ett repo (git init)
3. Kolla statusen (git status)
4. Skapa lite förändringar och committa… (git add, git commit)

Hur håller git reda på förändringar?



Övning

1. Skapa en mapp med en index.html fil med en div
2. Initialisera mappen till ett repo (git init)
3. Lägg till en css fil så att diven får en bakgrundsfärg
4. Kolla status i projektet (git status)
5. Committa css filen (git add, git commit)
6. Lägg till en border runt diven i css filen
7. Kolla status i projektet
8. Committa bordern

Working directory tillstånd

Working directory kan befinna sig i 3 olika tillstånd

- Förändringar som väntar på att bli “stagade”
- Förändringar är “stagade” men ännu inte committade
- Förändringar är committade och working directory är “rent”

Vi kan titta på “o-stagade/o-committade” förändringar i working directory

> git diff

Demo

Hur fungerar commits?

När vi har gjort flera commits kan vi få en överblick över vilka förändringar vi har gjort med

> git log

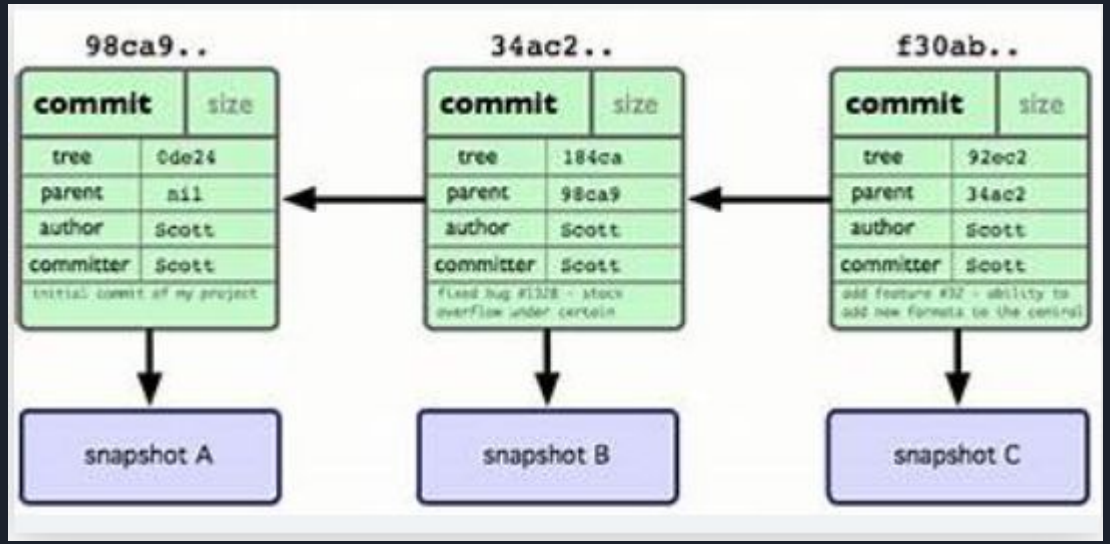
> git log –oneline

Demo

```
git log
commit 09f4acd3f8836b7f6fc44ad9e012f82faf861803 (HEAD -> master)
Author: w3schools-test
Date:   Fri Mar 26 09:35:54 2021 +0100

    Updated index.html with a new line

commit 221ec6e10aeedbfd02b85264087cd9adc18e4b26
Author: w3schools-test
Date:   Fri Mar 26 09:13:07 2021 +0100

    First release of Hello World!
```

Hur fungerar commits?

Alla commits skapar en kedja som återskapar tillståndet i working directory



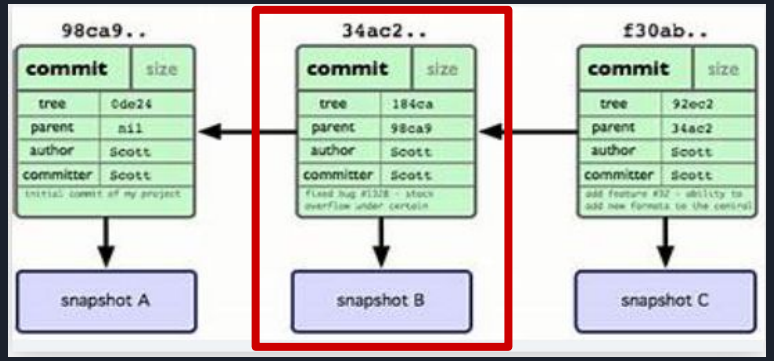
Hur fungerar commits?

Vi kan hoppa fram och tillbaka i denna kedjan!

> git checkout 34ac2… (hoppa tillbaka i tiden)

> git checkout HEAD (återställ till senaste commit)

Demo

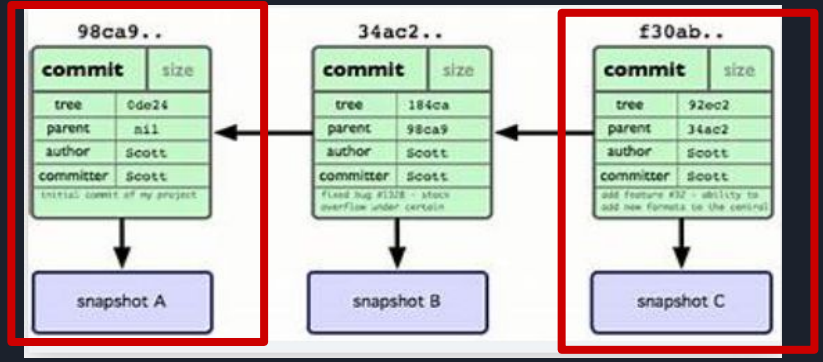


Hur fungerar commits?

Vi kan även titta på skilnaden mellan 2 olika committs med git diff

> git diff 98ca9.. f30ab..

Demo



Övning

1. Gör ytterligare en valfri förändring och kolla skillnaden mot repo (git diff)
2. Committa förändring (git add, git commit)
3. Prova att hoppa tillbaka till en tidigare commit (git log, git checkout)
4. Återställ till senaste commit (git checkout HEAD)
5. Kika på skillnaden mellan tidigare commit och senaste (git diff)

Frågor

- Vad är git? Använd gärna egna ord, din egna uppfattning….
- Vad gör git init?
- Vad är ett repository?
- Vad är en commit?

Frågor

- Ponera att du har ändrat i en fil, hur committar du?
- Vad är working directory?
- Vad är staging area?
- Vilka 3 tillstånd kan ett working directory ha?

Frågor

- Vad gör git log?
- Hur kan vi titta på skillnaden mellan olika commits?
- Hur hoppar vi fram och tillbaka i historiken?

Få hjälp....

> git <command> -help

Ger hjälp med ett commando för att se vilka flaggor som finns

> git help —all

Visar alla subcommndon och översikt av dessa

Ångra saker i git...

Ibland behöver vi ta bort saker i git, tex en fil

> rm uneccessary.html

Tar bort fil från working dir men om vi stagat filen så ligger den fortfarande på staging area… Hmm….

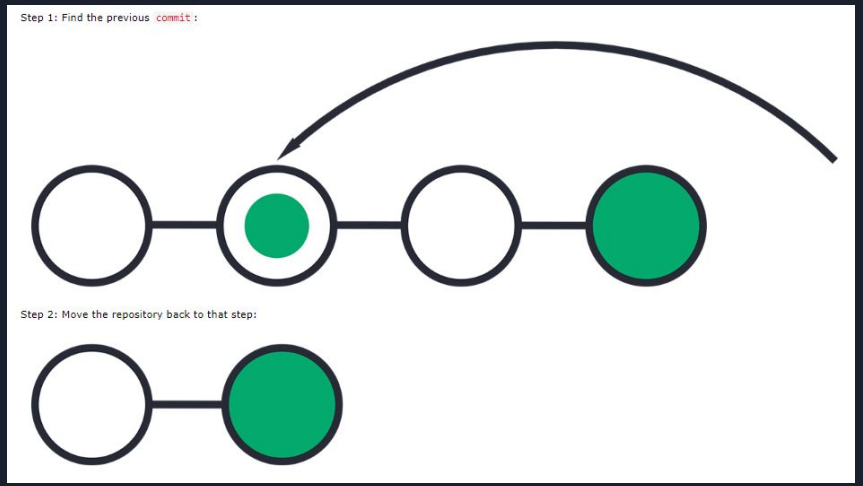
> git rm uneccessary.html (tar bort från både work dir och staging area)

Ångra saker i git...

Om vi vill inte bara vill gå tillbaka och kika i tidigare historik utan ta bort commits

> git reset 78ba34...

Tar bort alla commits efter denna commit



Övning Sammanfattning

1. Gå in i ditt projekt (din blogg) och initialisera ett git repo
2. Kolla status (git status)
3. Committa allt du hittills har gjort (git add –all, git commit)
4. Gör en liten förändring
5. Kolla status (git status)
6. Committa din förändring (git add, git commit)
7. Titta på historiken (git log)
8. Hoppa fram och tillbaka mellan din första och senaste commit (git checkout)
9. Skapa en onödig fil och staga den (git add)
10. Ta bort filen från working directory och staging area (git rm)
11. Ta bort allt du gjort från och med din första commit (git reset)

Läxa

Coderefinery

- https://coderefinery.github.io/git-intro/basics/# (~1 h)
  - Teori
  - Övningar
- Gör om dagens övningar om de inte sitter…

Rekommenderad youtube presentation (tar upp ungefär samma material)

- https://www.youtube.com/watch?v=8JJ101D3knE
- https://www.youtube.com/watch?v=tRZGeaHPoaw (annan med github)