

Programming

GitHub / CLI

Vad vi ska göra idag

1. CLI
2. GitHub

CLI – Command Line Interface

Command Line Interface – Vad är det?

- Text baserad miljö som gör det möjligt att skapa, flytta, kopiera och radera filer och mappar genom att använda olika kommandon
- Är samma kommandon som i Mac och GitBash.

```
drush_set_option('session_data', $session_data);
    .+?:
    .+?:
// Determine destination.
$default_dest = FALSE;
$home = drush_home();
if ($home) {
    $default_dest = drush_normalize_path($home . '/.drush/pantheon.aliases.drushrc.php');
}
$destination = drush_option_get('destination', $default_dest);
if ($destination == FALSE) {
    drush_set_option('destination', $default_dest);
    return;
}
// Ensure that the destination can be written to.
$fh = fopen($destination, 'w');
if ($fh) {
    return drush_set_option('destination', $destination);
} else {
    return drush_set_option('destination', $default_dest);
}
drush_set_option('destination', $destination);
```

Hur använder man den?

- cd – (change directory) flyttar till en mapp
- pwd – (print working directory) visar i CLI vart du är just nu
- ls – (list) visar innehåll i en mapp
- mkdir – (make directory) skapar en mapp
- touch – skapar en tom ny fil med en angiven namn
- rm – (remove) raderar en mapp eller fil
- rmdir - (remove directory) raderar en mapp
- mv – (move) flyttar en mapp eller fil
- cp – (copy) kopierar en mapp eller fil

CLI Övning A - Nu kör vi!

1. Se vart vi är: `pwd`
2. Skapa en mapp som heter Funky: `mkdir Funky`
3. Gå in i mappen Funky: `cd Funky`
4. Skapa en fil som heter `index.html` och en fil som heter `style.css`: `touch index.html style.css`
5. Se vilka filer vi har: `ls`
6. Radera filen `index.html`: `rm index.html`
7. Skapa en ny fil: `main.css`
8. Gå upp en mapp: `cd ..`
9. Radera mappen Funky: `rmdir Funky` (aha inte tom?)
 1. Du kan radera en mapp med filer i med : `rm -rf Funky`

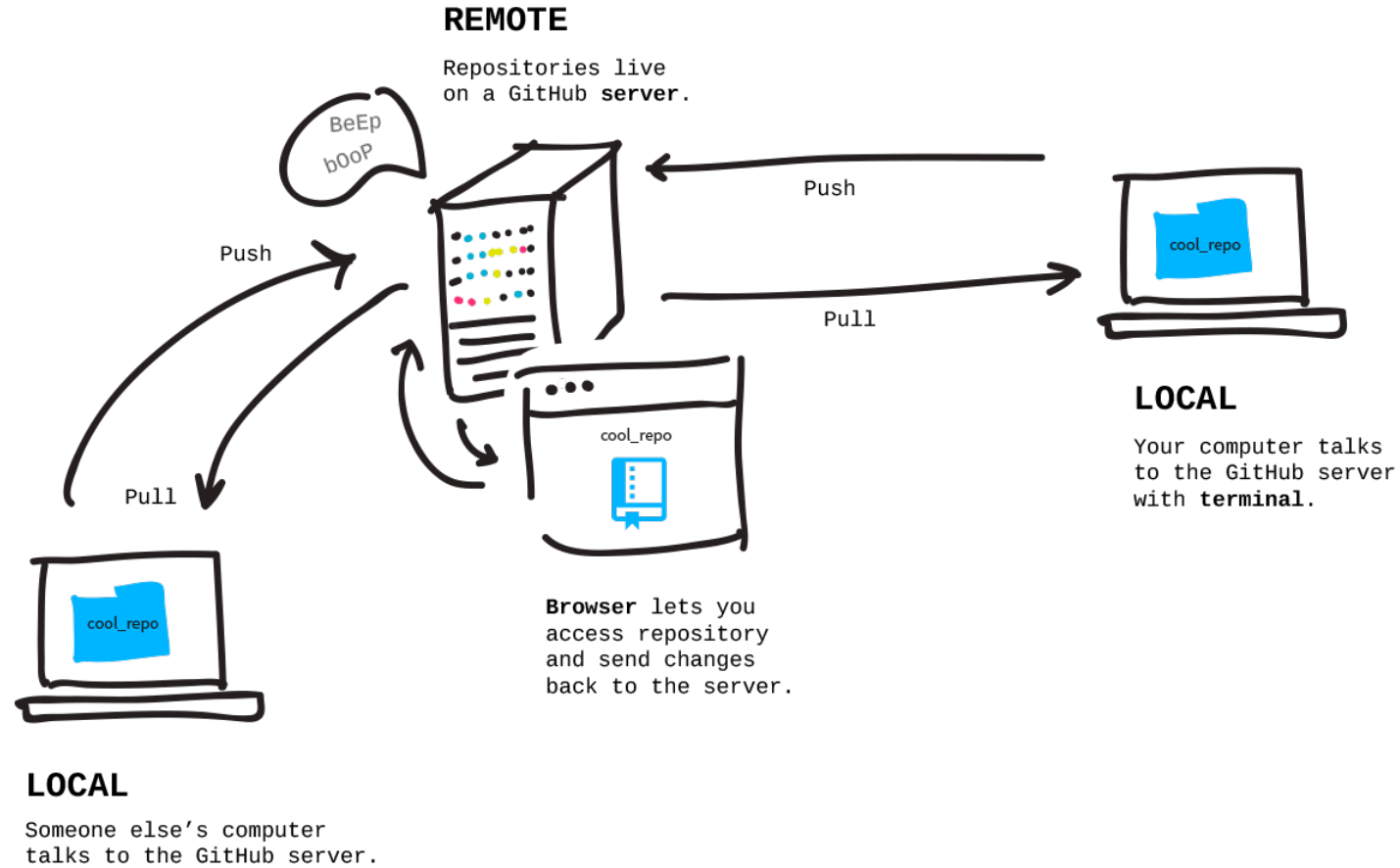
CLI Övning B – Nu kör vi igen!

1. Skapa en mapp med valfri namn och skapa tre filer
 1. index.html
 2. style.css
 3. app.js
2. Duplicera index.html till max.html: `cp index.html max.html`
3. Skapa en mapp i mappen
4. Gå in i den nya mappen
5. Skapa en fil, flytta.txt och flytta den till mappen som ni skapade i steg 1: `mv flytta.txt mapp-i-mappen`

GitHub



Vad är det?



Vad vi ska göra...

- Skapa konto på GitHub
- Konfigurera *git* lokalt
- Arbetsflöde
- Lära oss några *git* kommandon
- *Push* (ladda upp) filer till en *Repository* (mapp av din projekt på GitHub

Skapa Konto

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside millions of other developers.

Username

Pick a username

Email

you@example.com

Password

Create a password

Use at least one letter, one numeral, and seven characters.

Sign up for GitHub

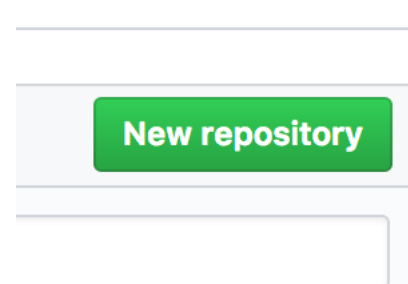
By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails.

Konfigurera *Git* Lokalt

```
git config --global user.name "Sune Sunesson"
```

```
git config --global user.email minfräckamail@gmail.com
```

Skapa *Repository* på GitHub

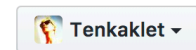


Ge den ett namn

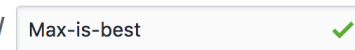
Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Repository name



Great repository names are short and memorable. Need inspiration? How about **probable-broccoli**.

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▼

Add a license: **None** ▼



Create repository

Nu CLI...

1. `git init` – skapar en repository i din datorn
2. `git add .` – Adderar det ändringar du har gjort i din lokala repository
3. `git commit -m "Förklaring"` – beskriver det ändringar du har gjort
4. `git remote add origin {origin}` (bara första gången) – kopplar den lokala repository med den på GitHub
5. `git push origin {branch}` – uploads local repository to GitHub on branch {branch}

Resurser

- <https://www.git-tower.com/blog/git-cheat-sheet/>
- <http://rogerdudler.github.io/git-guide/>
- <http://13pixlar.se/artiklar/guide-till-git-versionshantering-nar-den-ar-som-bast/>