

# GitHub

## Step By Step

### Förkunskaper - Terminal

Du skall kunna navigera mellan olika kataloger inom terminalen.

#### Krav:

- **GitHub** konto
- Verifierad **Email**
- **Git** installerat - <https://git-scm.com/downloads>

### Första gången du gör detta?

Detta steg görs bara **EN GÅNG** i livslängden av ditt 'Token'.

#### TUTORIAL:

<https://betterprogramming.pub/dealing-with-github-password-authentication-deprecation-1b59ced90065>

#### Instruktioner

#1 Öppna Github

#2 Gå in till Settings -> Developer Settings -> Personal Access Tokens -> Generate new Token

#3 Döp sedan ditt 'Token' till vad som helst.

#4 Ge behörigheter.

#5 Generera sedan ditt 'Token'.

#6 Kopiera Ditt Token och göm den någonstans säkert

#7 Senare när du blir efterfrågad om lösenord, så klistrar du in ditt '**TOKEN**'

(*'Token' kommer försvinna och ej visas upp efter du stänger ner GitHub*)

## Terminal Kommandon

\$ tecken skall **INTE** inkluderas inom terminalen

Navigera mellan mappar:

**Mac** == \$ cd MappNamn

**windows** == \$ dir MappNamn

Skriv cd/dir .. istället för namnet, för att hoppa ut ur mappen

Visa upp filer / folders == \$ ls / dir

Visa upp 'path' == \$ pwd / cd

Visa senaste Git == \$ git --version

Hur går det på Git? == \$ git status

Du kan navigera till path == \$ cd desktop/path

## Dessa steg utförs för varje nytt projekt du skapar

#1 Öppna Terminalen

#2 Navigera in till ditt projekt (roten)

#3 Mata in:

\$ git init

(Genererar git projekt lokalt på datorn)

#4 Mata in:

\$ git add .

(Förbereder alla filer och kataloger där vi står)

\$ git commit -m "Initial Commit"

(Meddelande)

#5 Om du inte redan har gjort det, skapa ett 'Repository' på GitHub.com

#6 Hitta **URL'n** på 'repository' och kopiera den

#8 Mata in:

\$ git remote add origin **RepositoryUrlHär**

\$ git push --set-upstream origin master (main istället för master ibland)

—Alternativt—

\$ git push -u origin master

## Nu då?:

Efter att du gjort stegen ovanför, så fort du ändrat eller lagt till någon ny fil eller kodrader, så behöver du framöver bara göra **följande tre steg**:

## STEP BY STEP:

#1 Öppna terminalen, navigera in till projektet

#2 Lägg till nya filer

\$ git add .

#3 Kommentera på förändringarna

\$ git commit -m "New Paragraphs"

#4 Lägg upp förändringarna

\$ git push

## Branch Commands

Grenar är oerhört viktiga när man vill arbeta med flertal gruppmedlemmar.

<code>\$ git branch</code>	Lista alla grenar
<code>\$ git branch feature/login</code>	Skapar en ny <b>gren</b> som heter 'feature/login'
<code>\$ git branch -d feature/login</code>	'Safe delete', tar bort grenen 'feature/login'
<code>\$ git branch -D feature/login</code>	Tvingar borttagning av gren
<code>\$ git branch -m newName</code>	Ändrar namnet på den befintliga grenen
<code>\$ git checkout master/main</code>	Navigerar till <b>master</b> eller <b>main</b>

## Merging & Conflicts

När man sedan är klar med sin feature, vill man skicka vidare förändringarna till 'master/main'. Då ska man navigera till Master/main, göra en 'pull', och sedan skriva **git merge**!

<code>\$ git merge feature/login</code>	Sammanfogar grenen 'feature/login' till master
---	--

Har man stött på konflikter i koden, löser man det enklast genom att ta bort kord man ej vill ha med, och ersätta det med den nya. Detta kommer automatiskt visas upp inom IDÉ'n. Är man klar med konflikter i koden gör man sedan en 'git add, commit, push' som vanligt!

## ***FORKING***

### **Step 1: Create a GitHub Repository**

- Go to GitHub and create a new repository.
- Set its name, description, and visibility (public or private).
- Click "Create repository."

### **Step 2: Share Your Repository**

- Share the repository link with your students.

### **Step 3: Students Fork the Repository**

- Instruct students to click "Fork" on your repository's page.
- This makes a copy in their GitHub account.

### **Step 4: Clone the Forked Repository**

- Students open their forked repo and click "Code."
- They copy the URL or use "Open with GitHub Desktop" to clone it locally.

### **Step 5: Make Changes**

- Students edit code in their local copy using their code editor.

### **Step 6: Commit and Push Changes**

- In the terminal, students run:`bash`

```
git add .  
git commit -m "Describe changes"  
git push origin main
```

(Replace `main` with the branch name if different.)

### **Step 7: Create a Pull Request**

- On their forked repo's page, students click "Pull Requests."
- They hit "New Pull Request" and describe the changes.
- Click "Create Pull Request."

### **Step 8: Review and Merge**

- As the owner, you'll get a notification about the pull request.
- Review the changes, add comments if needed, and click "Merge" to accept them