

# Git in der Schule verwenden

Dies ist eine Auflistung der wichtigsten Befehle, die nötig sind um, an den Schulcomputern, auf die Github Repository unseres P-Seminars zuzugreifen sowie Änderungen hoch/herunterzuladen.

## 1. Das Terminal

Das Terminal ist die Kommandozeile auf einem Ubuntu System. Um Git zu bedienen muss das Terminal in dem für Github benutzten Ordner, auf dem Schulrechner, geöffnet sein.

*Ordner Öffnen> Rechtsklick im Ordner> Terminal hier öffnen*

oder allgemein

*Startmenü> Terminal*

## 2. Einrichtung eines Proxys

Damit über die Github Software die online gespeicherten Dateien heruntergeladen werden können muss diese mit dem Internet verbunden werden. Um einen Zugang zu schaffen muss folgender Befehl im Terminal eingegeben werden:

*export https\_proxy="https://BENUTZERNAME:DEINPASSWORT@10.16.1.254:8080"*

## 3. Konfigurieren von Git

Nun muss Git konfiguriert werden, damit es benutzbar wird.

*git config --global user.name „DEIN NAME“*

*git config --global user.email „EMAILADRESSE“*

Anschließend sind folgende Befehle hilfreich um Git auf der Konsole übersichtlicher zu machen:

*git config --global colour.ui „auto“*

*git config --global pack.threads „0“*

## 4. Git auf dem Computer einrichten

Die gewünschten Dateien, welche sich noch auf einem Server befinden müssen nun einem Ordner auf dem Computer zugewiesen werden. Dazu muss die Konsole im dafür vorgesehenen Ordner geöffnet sein (siehe 1.) Mit dem folgenden Befehl wird der Ordner zugewiesen:

*git clone „ADRESSEDERDATEI“*

Sollte schon ein solcher Ordner mit heruntergeladenen Dateien bestehen schadet es nicht ihn beim Starten von Git zu initialisieren:

*git init*

## 5. Commits erstellen & hoch-/herunterladen

Soll geänderter Quellcode oder allgemein aktualisierte/neue Dateien hochgeladen werden, so fasst man alle Änderungen in so genannte Commits zusammen. Diese Commits erhalten zudem eine Beschreibung welche andere Projektteilnehmer über die Änderungen aufklären soll.

Solltet ihr neue Dateien zum Projekt hinzugefügt haben müssen diese erst einmal zum Commit hinzugefügt werden.

<code>git add „DATEINAME“</code>	fügt eine einzelne Datei hinzu
<code>git add „ORDNERNAME/“</code>	fügt einen ganzen Ordner hinzu

Nun können alle hinzugefügten Dateien zu einem Commit zusammengefasst werden.

`git commit -a`

Anschließend wird man aufgefordert den Commit zu beschreiben und zu speichern. Es kann auch sofort eine Änderungsbeschreibung eingefügt werden.

`Git commit -a -m „BESCHREIBUNG ALLER ÄNDERUNGEN“`

Auch nachträglich können vergessene Dateien zu einem Commit hinzugefügt werden.

`git add „VERGESSENE DATEI“`  
`git commit --amend`

Soll eine Datei oder ein Ordner aus dem Commit entfernt werden hilft dieser Befehl:

`git rm „DATEINAME“ -f`

Sobald der Commit fertig ist kann er hochgeladen werden.

`git push „ADRESSE DES PROJEKTS“ „NAME DES BRANCHS“` (siehe 7.)

Um die neuste Version des Projekts von den Github Servern heruntergeladen werden muss dieser Befehl benutzt werden:

`git pull`

## 6. Lokale Änderungen zurücksetzen

Möglicherweise wurde der Quellcode durch eine lokale Änderung so verändert, dass Fehler auftreten oder das Programm überhaupt nicht mehr nutzbar ist. In diesem Fall können alle lokalen Dateien auf die aktuellste auf Github vorhandene Version zurückgesetzt werden.

`git checkout -f`

oder

`git reset --hard`

## 7.Branches überblicken und erstellen

Mit der Hilfe von Branches können verschiedene Umgebungen für das Projekt erstellt werden (z.b. „Testbranch“ und „Masterbranch“).

Mit folgendem Befehl werden alle bestehenden Branches sichtbar gemacht. Der derzeit benutzte Branch wird entweder farblich oder mit einem \* gekennzeichnet.

*git branch*

Es können auch neue Branches erstellt werden.

*git branch „BRANCHNAME“*

Um zwischen Branches zu wechseln ist folgender Befehl notwendig:

*git checkout „BRANCHNAME“*

Unnötig gewordene Branches können auch gelöscht werden.

*git branch -d „BRANCHNAME“*

---

Adresse des Projekts auf Github:

<https://github.com/DieSuperSexySystemSchreiber/P-Seminar>

Quelle:

[wiki.ubuntuusers.de/Git](http://wiki.ubuntuusers.de/Git)