

Anleitung: GitHub

Astro-AG der Liebigschule

Patrick Hemberger

18. November 2016

1 Was ist GitHub?

GitHub ist eine Online-Versionsverwaltung, das das Versionskontrollsystem *git* verwendet. Damit ist es möglich, Dateien oder ein Projekt zentral zu verwalten, sodass bei mehreren Bearbeitern (im folgenden: Contributor) keine Verwirrung entsteht, wer jetzt von welcher Datei die aktuellste Version auf seinem Rechner hat, und wie man es jetzt schafft, ohne Verlust alle aktuellen Dateien an einem zu sammeln. Eine Instanz dieser Versionsverwaltung wird *Repository* genannt, kurz *Repo*.

2 Nutzung von Github

2.1 Webtool

Man kann sich online auf github.com einloggen. Die Login-Daten zu unserem Git-Repository befinden sich auf dem gesonderten Logindatenblatt. Man kann online einsehen, welche Dateien zuletzt geändert oder hinzugefügt wurden. Es ist möglich, sich, sofern man einen GitHub-Account besitzt, selbst als Contributor einzutragen, allerdings möge man in diesem Fall bitte seinen Namen, Nutzernamen, eine regelmäßig abgerufene Mailadresse, sowie die eigene Mitgliedszeit in der AG im vorhandenen Tabellendokument eintragen, damit jeder einen Überblick behält, wer hier mitarbeiten kann.

2.2 Verwaltung mit einem Terminal (Linux)

1. Um ein Repo zu bearbeiten, musst du es zunächst in ein *lokales Repo* klonen. Das geschieht folgendermaßen:
 - (a) Rufe über die Menüleiste ein *Terminal* auf. Darin kannst du Befehle schreiben. Wechsele in den Ordner, in den du das Repo klonen möchtest. Nutze dazu die Befehle ***cd*** **<Ordner>** (In der lokalen Ordnerstruktur eine Ebene tiefer, um in den gewählten <Ordner> zu springen) und ***cd*** **..** (In der lokalen Ordnerstruktur eine Ebene höher springen.)

- (b) Rufe den Befehl ***git clone https://github.com/astroAG/AstronomieAG.git*** "**<Name>**" auf. **Beachte:** Der Zusatz "**<Name>**" bezeichnet den Ordner, indem sich nachher das gesamte lokale Repo befindet (z.B. AstroGIT für den Ordner AstroGIT). Wird **<Name>** weggelassen, so wird der Ordner ebenfalls erstellt, erhält allerdings die Bezeichnung unseres Repos auf GitHub (also AstronomieAG).
2. Nachdem du unser Repo geklont hast, kannst du dir jederzeit seinen Status anschauen. Dazu rufst du den Befehl ***git status*** auf. Wenn dein lokales Repo mit dem Online-Repo vollständig übereinstimmt, so gibt das Terminal "Already up-to-date" aus. Hast du hingegen beispielsweise Änderungen im lokalen Repo vorgenommen, so wird dir dies ebenfalls angezeigt.
 3. Um dein lokales Repo auf den aktuellen Stand des Online-Repos zu bringen, wenn ein anderer Contributor zuletzt etwas geändert hat, nutze den Befehl ***git pull***. **Beachte:** Damit könntest du eigene Änderungen zunichte machen. Schau dir, wenn du unsicher bist, zunächst über das Webtool an, welche Dateien verändert wurden, und sichere nötigenfalls Konfliktdateien außerhalb des lokalen Repos, um die Änderungen des anderen Contributors mit deinen manuell zusammenzuführen.
 4. Um eine Datei oder einen Ordner initial zu versionieren oder zum Commit vorzuzeichnen, verwende den Befehl ***git add <Datei/Ordner>***.
 5. Nun kommt der Commit. Mit einem Commit fügst du die zuvor mit *git add...* vorgemerkten Dateien und Ordner zu deinem lokalen Repo hinzu. Um einen Commit durchzuführen, verwende den Befehl ***git commit -m "<Commit-Nachricht>"***. Die **<Commit-Nachricht** ist eine Nachricht, die du anderen Contributors hinterlassen kannst, sodass sie sehen, was du in etwa geändert hast. **Beachte:** Bevor du einen Commit durchführst, prüfe bitte nochmal mit *git status* (siehe oben), ob sich im Online-Repo etwas geändert hat.
 6. Nun kopierst du alle durch den Commit durchgeführten Änderungen aus deinem lokalen Repo in das Online-Repo. Dies geschieht mit dem Befehl ***git push***. **Beachte:** Führe diesen Schritt unmittelbar nach dem Commit durch, sodass keine Commits herumliegen. Wenn das doch passiert, entstehen verschiedene Zweige (Branches), die dann gemerged werden müssen. Mergen ist anstrengend und sprengt (in dieser Version) den Rahmen dieser Anleitung.

2.3 Verwaltung mit einer Windows-Shell

kommt (hoffentlich) noch