|  |  |  |  |
| --- | --- | --- | --- |
| Git init | Initialise | Erstellen/Vorbereiten | Macht einen Dateiordner zum Git Repository |
| Git clone |  | Klonen/Kopieren | Erstellt eine lokale Kopie eines Repositories |
| Git status |  |  | Zeigt Informationen zum Status eines Repositories  - Ist der lokale Ordner auf dem gleichen Stand wie auf dem Server? |
| git add |  | Hinzufügen | Wählt Dateien an, die man im nächsten Commit haben will und meldet sie dafür an |
| Git commit |  | Sich zu etwas bekennen | Ein Commit erstellt einen neuen Speicherpunkt. Auf jeden Commit kann zurückgesprungen werden. Noch wird nichts auf den Server geladen. |
| Git commit --ammend |  | Etwas anhängen | Beim letzten commit wurde etwas übersehen, also kann man hiermit noch unauffällig eine Änderung anfügen. |
| Git reset --soft HEAD~1 |  | Weiches zurücksetzen | Der letzte Commit wird wieder zurückgenommen, die Dateien bleiben aber erhalten. HEAD~1 beschreibt, dass nur ein Commit zurückgenommen wird. In der HEAD-Datei sind alle Commits aufgezeichnet. |
| Git reset --hard HEAD~1 |  | Hartes Zurücksetzen | Lösche den letzten Commit und alle darin gemachten Änderungen. Die Dateien haben dann wieder den Zustand des vorletzten Commits. |
| Git log |  | Stamm ansehen | Listet alle Commits auf, von Neu zu Alt |
| Git pull |  | Ziehen | Aktualisiert die lokalen Dateien vom Server, lässt jedoch teils eigene Änderungen bestehen. |
| Git fetch |  | Fangen | Aktualisiert die lokalen Dateien und löscht alle nicht hochgeladenen Änderungen |
| Git tag |  | Etikett | Zeigt an, wenn einzelne Commits, meistens Wichtige, markiert wurden |
| Git branch neuer\_branch |  | Ast/Zweig | Erstellt eine Abzweigung, die unabhängig vom Ursprungsast oder dem Master-Branch bearbeitet und synchronisiert werden kann. |
| Git checkout neuer\_branch |  | Abmelden/testen | Wechselt in den angegebenen Branch. Die lokalen Dateien haben jetzt den Inhalt dieses Branches. |
| Git checkout master    Git merge neuer\_branch |  | Zusammenführen | Erst wechselt man in den Ziel-Branch  Danach fügt man die Änderungen dorthin ein. Bei jeder Änderung wird abgefragt, welche Version man behalten will. |
| Git branch -d neuer\_branch | Delete | Löschen | Löscht einen Branch |