„**git commit**“ commited eine änderung im aktuellen branch. „

„**git branch <name>**“ erstellt einen Zweig einen branch mit dem angegeben Namen an der Stelle an der man sich befindet (head).

„**git checkout -b <branchName>**“ Gibt an auf welchen branch man arbeiten möchte

„**git merge <branchName>**“ Verbindet den aktuellen branch auf dem man sich befindet mit dem angegebenen.Merke. wir mergen **<branchName>** in das aktuelle branch in dem wir uns befinden. Wenn nun in dem anderen branch weitergearbeitet wird. Müssen wir neu mergen.

„**git rebase <branchName>**“ Verbinden den aktuellen branch in dem ich mich befinde mit dem angegebenen.

**Referenzen:**

Man kann anstatt branches oder festen Referenzen wie **HEAD** commits angeben. Dazu kann man „**^**“ als direkter Vorgänger oder „**~**“ nutzen.

„**main^**“ ist das vorherige commit auf das main gerade zeigt.

„**main~3**“ bedeutet der 3te Vorgänger von main

„**git branch -f <branchName> <referenz>**“ erstellt einen Branch ohne vorher ausgecheckt also nen aktuellen Branch ausgewählt zu haben. Erstellt oder verschiebt den branch mit dem branchNamen an die Stelle referenz. **HEAD** kann nicht ausgewählt werden.

„**git reset <ref>**“ Setzt den Zeiger der aktuellen Branch auf die Referenz. Somit sieht es so aus als ob die vorherigen Commits nie stattgefunden haben. Immer vom aktuellen branch (checkout) ausgesehen.

„**git revert <ref>**“ anstatt der Referenz an einen anderen knoten zu setzen wird ein neuer commit/Knoten erstellt welcher die Änderungen enthält zum rückgängig machen.

„**git cherry-pick <Commit1> <Commit2> <…>“** setzt die angegebenen Commits unter den Head (also dort wie man gerade ausgecheckt ist). Der zeiger/Head wird auf den letzten Commit der angegeben ist verschoben.

„**git rebase -i <ref>**“ interaktives rebase. Es öffnet sich ein Fenster dort kann man alle Commits auswählen. Auch die Reihenfolge. Und setzt sie hinter der referenz

„**git commit --amend** “ Ändern des aktuellsten Commits. BSP: git commit --amend -m "an updated commit message"

„**git tag <tagname> <ref/commit>** “ gibt der Ref/dem Commit einen eindeutigen tag.