Commands git:

cd xxx– sowas wie „gehe in Objekt xxx“

ls –

ls -la = list all (Auflisten aller Einträge innerhalb des Objekts)

rm -rf .git = remove .git Datei (diese ist Voraussetzung für Tracking des Projektordners)

git init = Anlegen der Git-Datei

git status = Anzeigen der nicht getrackten Dateien/ auch refresh des Ordners; nach Initial Commit auch Anzeige der abweichenden/geänderten Daten, die synchronisiert werden können

touch .gitignore = Anlegen der Whitelist (welche Dateien sollen public sein?)

* Pflichtteil(?) der .gitignore Datei: .DS\_Store + dann beliebige datei die geblockt werden soll

In Working Directory sind alle Dateien die bei gitstatus angezeigt werden

In Staging Area sind alle Dateien, die in Repository geladen werden sollen

Git add xxxx 🡪 Hinzufügen von xxx

Git add -A 🡪 alles hinzufügen das nicht getracked wird

Git reset xxx 🡪 löschen der temporär gespeicherten Daten in Staging Area

Git reset 🡪 löschen alle

Damit Daten commited werden/ final ins Repository hochgeladen werden, erst git add benutzen!

Dann: git commit -m „-kurze Info bzgl Änderungen an Dateien-“

Git log 🡪 Anzeigen der zuletzt getätigten Commits (von wem, welche änderungen, wann? Etc)

Git diff 🡪 anzeigen der genauen Veränderungen des Codes im Vergleich zum letzten Commit

Clone 🡪 Kopieren der Dateien auf anderes Verzeichnis, Punkt symbolisiert „copy sämtliche Files in folgendes Verzeichnis

Git pull origin master 🡪 Updaten des Repositories (alle Änderungen nun aktuell) um Überschreiben bei gleichzeitigen Zugriff von mehreren Developern zu vermeiden

* Immer Pull bevor PUSH!

Git push origin master 🡪 alle Dateien in den master branch geschoben

Organisation der Branches:

Git branch xxx 🡪 Anlegen des Branches xxx

Git branch 🡪 anzeigen aller lokal verfügbaren Zweige

Git checkout xxx 🡪 Setterfunktion für Zweig xxx (Weiterarbeit nun auf branch xxx)

Um lokalen Branch ins Repository zu übernehmen: git push -u origin xxx

Git branch -a 🡪 anzeigen aller Branches (lokal sowie remote)

Um branches zu mergen:

1. Git checkout master (Update des Remoteordners)
2. Git pull origin master (Update der Lokaldateien)
3. Git branch –merged (Anzeigen der bereits verschmolzenen Branches)
4. Git merge xxx (Merge with master)
5. Git push origin master (Endgültiges mergen in remote)

Deleting Branch:

Git branch –merged

Git branch -d xxx (lokales Löschen)

Git branch – a (Anzeigen des gerade gelöschten lokalen branches; Gegencheck)

Git push origin –delete xxx (endgültiges löschen)

Git branch -a (gegencheck)