Cd \*\*\* - выбор папки  
dir – показывает что есть внутри папки  
Git init – создание репозитория  
git status – проверка репозитория  
git log – показывает коммиты, которые вообще есть  
git log –pretty-oneline – более конкретизированый лог  
git branch –M \*\*\* - переименовать ветку  
git branch – количество веток и их названия  
git branch –D \*\*\* - удалить ветку \*\*\*  
git add . – добавить файлы в репозиторий  
git checkout \*\*\* – переход на другую ветку  
git checkout –b \*\*\* - создание новой ветки с названием \*\*\*  
git commit –m “\*\*\*” – создание коммита с сообщение внутри  
git commit –amend – переименует нынешний коммит   
git merge \*\*\* - соединить ветку с веткой \*\*\*  
git config --list - работа с конфигом гита (показывает, что можно отредактировать)  
git config –global user.name “\*\*\*”  
git remote add origin [https://\*\*\*](https://***) - привязать удаленный репозиторий к локальному гиту  
git remote remove origin – отвязать удаленный источник от локального гита  
git push origin main – отправить изменения сделанные на локальном компьютере в ветке main на удаленный репозиторий (гитхаб)  
git clone <remote-repo-url> - скопировать данные удаленного репозитория на локальный компьютер (c созданием репозитория на локальном компе)  
git pull – копирование данных с удаленного репозитория без создания репозитория на локальном компе  
git clone -b <branchname> <remote-repo-url> - скопировать нужную ветку с гита  
git remote –v – показывает к какому удаленному источнику я подключен  
git remote set-url origin \*\*\* - поменять удаленный источник на другой  
git reset –hard HEAD~\* (\* - число сколько коммитов надо удалить) – возврат на старый коммит с удалением более новых  
git reset –soft HEAD~\* - удаляет предыдущие коммиты сохраняя нынешний  
git push origin –delete \*\*\* - удаляет ветку на гитхабе с названием \*\*\*  
git diff –staged – показывает изменения, которые сделал пользователь (перед комитом)

.gitignore – файл в который вносятся документы, которые нужно заигнорировать гиту

**Работа с Git Tag**

git tag v1.0.0 – дать таг коммиту (v1.0.0)  
git push origin v1.0.0 – пушит таг версию на удаленный репозиторий последнему коммиту  
git push –d v1.1.0 – стирает таг 1.1.0 с локального репозитория  
git push origin –delete v.1.1.0 – удаляет таг 1.1.0 с удаленного репозитория  
git checkout v.1.0.0 – переходит на более старую версию тага v1.0.0 (вместо того, чтоб делать это по коммитам)  
git checkout main – переходит на последнюю версию тага  
git tag –a v1.5.0 44f59a506c7e3461574ca9092dc3d5ff86aabeaf – затагирует коммит, который был ранее сделан, но не затагирован по какой-то причине  
git push origin –tags – запушить таги на гитхаб