

Git	
curl -u 'имя_на_Гитхабе' https://api.github.com/user/repos -d{"name":"название_репозитория"} ' - создать репоз. через консоль	
git clone github.com:имя_на_Гитхабе/название_репозитория.git - клонирование удаленного репозитория на ПК	
git push – отправить изменения с локального репозитория на внешний	
git push имя_ветки - отправить изменения в определенную ветку	
git push -u origin название_ветки - запустить локальную ветку в УР	
git push origin --all - запустить все ветки в УР	
git pull – отправить изменения с внешнего на локальный	
--rebase – без ненужных merge-коммитов	
git init - создать локальный репозиторий (директория .git)	
git status - статус локального репозитория	
git fetch - получить изменения с удаленного репозитория	
git add имя_файла - добавить файл под Git	
git add . - добавить все файлы в индекс	
-i - интерактивное добавление изменений в индекс (что оставить, что убрать)	
git commit -m “add имя_файла” - закоммитить (отправить в репозиторий)	
-m - message	
git commit -am ‘add имя_файла’ - добавить файл и закоммитить	
git commit имя_файла -m “add имя_файла” - с явным указанием файла, без add, если файл уже в индексе	
--amend - добавить изменения в существующий коммит(откат коммита (через reset) и выполнению нового коммита с новыми данными)	
--no-edit - не менять описание коммита	
git checkout имя_ветки - переключиться на ветку	
git checkout -b имя_ветки - создать ветку и переключиться на нее	
git checkout хэш_коммита - переключение на коммит	
git branch - показать текущее месторасположение	
git branch имя_ветки - создать ветку	
git branch -d имя_ветки - удалить ветку	
git branch -m имя_ветки - переименовать ветку	
git merge имя_ветки - вмержить ветку в активную (если конфликт, то удалить вручную)	
git diff – посмотреть изменения в файлах не добавленных в индекс	
--staged – в файлах добавленных в индекс	
git diff имя_ветки1 имя_ветки2 - посмотреть изменения в ветках	
git log - история коммитов/список всех выполненных коммитов, отсортированных по дате добавления	
-p - вывод дифа для каждого коммита	
git show хеш_коммита - показать изменения сделанные этим коммитом	
git blame имя_файла - информация об авторе, кто и в каком коммите менял строки в указанном файле	
git clean -fd - отменить последние действия с неотслеживаемыми файлами/папками	
-f - force, -d - directory	
git restore имя_файла - отменить изменения в файле из рабочей директории	
--staged - перемещает изменения, сделанные в файле, из индекса в рабочую директорию	
git revert id_коммита - отменяет коммит по его id (создает новый коммит, в котором выполнены изменения обратные указанному в команде коммиту)	
git reset - отменить коммит и переместить в рабочую директорию	
--mixed - по умолчанию	
--hard - полное удаление, HEAD~ - один коммит от последнего коммита, HEAD~2 - два последних коммита	
git stash - спрятать файлы, директории	
git stash pop - вернуть спрятанные файлы, директории	
.gitignore - файл, в который добавляются имена файлов, директорий, которые будут игнорироваться	
git tag 1.0.0 id_коммита - установить метку для закрепления момента выпуска версий	
git remote add origin github.com:имя_на_Гитхабе/название_репозитория.git	
git branch -M main	-соединение ЛР с УР
git push -u origin main	
git config --global user.name ‘имя_юзера’ - указать имя юзера	
git config --global user.email ‘email_юзера’ - указать email юзера	
git grep слово - поиск в файлах по строке/возвращает список строк во всех файлах в которых есть данная подстрока	
-i - без учета регистра, имя_коммита - поиск в конкретном коммите	
\$(git rev-list --all) - поиск по всей истории/возвращает список хешей коммитов	
git rm имя_файла – удаление файла (rm + git add)	