* слияние веток переход в ветку в которую мы хотим влить другую ветку, git merge имя ветки которую вливаем,
* git checkout название ветки – переход в ветку
* git giff просмотр незакомиченных изменений
* git giff <имя файла> просмотр незакомиченных изменений в конкретном файле
* git giff <commit1> <commit2> просмотр незакомиченных изменений в коммите между 1 и 2, если поменять местами будут изменения в обратном порядке
* git restore <имя файла> файлы которй были добавлены в индекс но не хотим комитить
* git restore -- staged <имя файла> до добавления в индекс (git add .)
* git reset --hard удаление всех незакомиченных изменений, но не удалит то что не было добавленное в индекс
* git clean –f удаляет абсолютно все незакомиченные файлы
* git rm –cashed <имя файла> отменение отслеживания файла, если гит игнор создался после создания этого файла и он начал отслеживаться
* сброс коммитов, удаление версии, коммита
* удаление коммитов
* git reset –-soft <хэш коммита> файл остается но откатывается на предыдщий этап
* git reset –-mixed <хэш коммита> удалит и комит тоже
* git reset –-hard <хэш коммита> удаление всех изменений безвозвратное
* переименование коммита git commit –amend –m “новое имя комита”
* отмена слияния веток
* git config --global user.name "apolinariya zalavskaya"
* git config --global user.email "apolinariyazalavskaya@mail.ru" регистрация в системе гит
* pwd проверка работы программы, покажет в какой папке мы находимся
* cd ‘путь к папке’ изменение нашего нахождения
* ls –la показать все имеющиеся документы в этой папке
* делаем изменения в файле, сохраняем их:
* git add .
* git commit -m "initial"
* git show просмотр кто и когда сделал изменения, и следующее
* git log
* vs code \*.расширение – какие файлы необходимо игнорировать и не проверять в гите
* /папка/
* !/папка/ отрицание игнорирования
* Git branch показать какие есть ветки и в какой ветке мы находимся
* Сделать изменения в файле, закомитить,
* Git push добавить на сервер гитхаб
* Git checkout название ветки – переход на другую ветку
* Git checkout --b название ветки – создание новой ветки(необходимо проверить в какой ветки ты изначально находишься)
* Git status покажет в какой ветке находимся
* Делаем изменения в файле в какой-то ветке, не хотим сохранять, но хотим перейти в другую ветку:
* Git checkout
* Git stash
* Git status(git checkout)
* Изменения, комитим, переходим в ветку, которую застешили
* Git stash pop вернулись к работе с этой веткой, закомитили изменения, добавили на сервер
* Перешли в ветку, к которой хотим прибавить изменения из дргуой ветки
* Git merge название ветки, которую хотим влить в ту, в которой находимся
* Git log перечисление всех сделанных комитов, от последнего к первому вниз
* Git init – создание нового репозитория