**КОНФИГУРАЦИЯ .GITCONFIG**

git config --global user.name "Andrey K"

git config --global user.email [and13@mail.ru](mailto:and13@mail.ru)

git config –l #просмотр данных выше, хрантися в домашнем каталоге .gitconfig

**РАБОТА С LOCAL REPOSITORY**

git init /путь к папке/ Создаем папку где будет хранилище, инициализируем её

В папке создаются файлы

git status статус гит

git add \* или конкретные файлы – добавить в комит

git commit -m "сообщение" создать снимок файлов

**ИСТОРИЯ ИЗМЕНЕНИЙ, ВОССТАНОВЛЕНИЕ .gitignore**



git log просмотр изменений комитов

git log –цифра просмотр последний комитов по числу в команде

git log –l1 –p посмотреть что было сделано

git checkout -- file1.txt отмена изменение в file1.txt (если что-то неверно сделано в файле)

git diff –staged посмотреть разницу между staged и комитом (что изменилось)

**игнорирование файлов или каталогов для отслеживания:**

создать файл .gitignore там списком файлы, папки или вида\*.log

**ЗАГРУЗКА ПРОЕКТА НА GitHub gitpush**

Загрузить ИЗ ГитХаба:

git clone линк из гитхаб

git push origin загрузить в гитхаб (надо находится в директории репозитория)

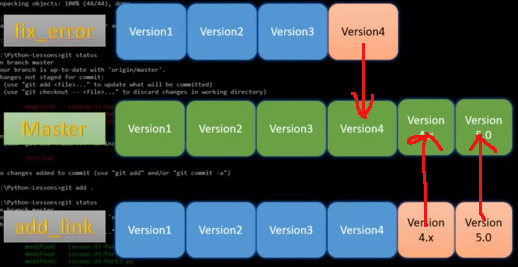
**НАСТРОЙКА ЛОГИНА ЧЕРЕЗ SSH:**

В SETTINGS на сайте в разделе ssh and gpg keys добавить паблик кей (содержимое публичного ключа)

git remote –v в папке репозитория – просмотр текущих ссылок на гитхаб

git remote set-url origin ссылка на ssh сайта установить ссылку на SSH

**СОЗДАНИЕ И РАБОТА С ОТВЕТВЛЕНИЯМИ Branch**



Master – золотая копия, её никто не трогает.

Создается бренч с полной копией где редактируется до рабочего состяния и после объединяется в основной комит. После чего бренч можно удалить

git branch - просмотр бренчей

git branch fix\_error - создать бренч fix\_error

git checkout fix\_error - перейти в этот бренч

git branch –d fixerror - удалить бренч

git checkout –b user\_fix\_error - создать и перейти в бренч user\_fix\_error

git merge user\_fix\_error - объединить бренч, запускается из master бренч

git branch –d user\_fix\_error - удалить бренч

git push origin --delete назв бренча – удалить бренч на гитхабе (удаленно)

**ВОЗВРАТ НА ПРЕДЫДУЩИЕ ВЕРСИИ**

git log - посмотреть хеши версий

git checkout хеш версии - перейти в промежуточную версию комита

git commit --amend - отредактировать комит не создавая новый

git reset --hard HEAD~2 - удалить 2 последних комита, вернется на 2 версии

git reset—soft HEAD~3 - удалить 3 комита, но оставить посл версию текущей

**СИНХРОНИЗАЦИЯ ЛОКАЛЬНОГО РЕПОЗИТОРИЯ С УДАЛЕННЫМ**

git add \*

git stash

git pull