# **------------------ GIT -----------------**

# **Basics Git**

**git version git --version** проверка версии git

**git init** создание локального репозитория

**git help** выводит все команды git

# **Git Config**

**git config** настройка самого git

**git config --system** настройка git для всех пользователей текущей системы

**git config --global** настройка git для конкреного пользователся системы

**git config** или **git config --local** настройка git для текущего репозитория

# **Main**

**Git Status -> Git Stage -> Git Commit**

**git status** показывает изменения в дереве, отслеживаемые и неотслеживаемые файлы git , которые будут или не будут добавленные в git

**git add .** добавление всех файлов в stage(черновая область) для последующего их коммита

**git add my\_file** добавление только указаного файла в область stage (черновая область)

**git commit -m “my\_text”** создание коммита с описанием. Команда git commit -am “my\_text” также автоматически делает git add ы.

# **Git Information**

**git log -> Q** показывает созданные комиты

**git log -p** выводит все коммиты с более подробным выводом

**git show f3e1j61x01sfо5bz1** показывает детальную информацию про указанный коммит

**git reflog** выводит все изменения в проекте с момента создания ( движения, коммиты, ветки)

**git diff** выводит изменения в файлах, которые были после последнего коммита , было +++ добавление кода в файле , было --- удаление кода в файле . Изменения которые еще не находяться в stage

**git diff --staged** выводит изменения в файлах которые, уже были добавленнные в stage

# **Git Come Back**

**git restore .** возвращате все файлы до состояния предидущего коммита (еще не были добавлены в stage через git add .)

**git restore my\_file** возвращает указаный файл до состояния предидущего коммита коммита (еще не был добавлен в stage через git add my\_file)

**git restore --staged .** выбрасывает все измененные файлы из stage ( потом нужно писать git restore .)

**git restore --staged my\_file.txt** выбрасываеть указаный файлы из stage ( потом нужно писать git restore my\_file.txt)

# **Actions**

**git mv my\_file.txt new\_name.txt** переименование файла

**git rm my\_file.txt** удаление файла из stage + и из проекта

**git rm --cached my\_file.txt** удаление файла из stage , а в проекте остаеться

# **Moving**

# **Git Ignore**

**.gitignore** внутри файла хранться файлы, которые не будут добавляться в stage

**my\_file.txt** не будет записываться в stage указаный файл

**\*.txt \*.log** не будут записываться в stage все файлы с указаным розширением

**. \*.txt !my\_file.txt** не будут записывать все файлы с указанм окончанием, кроме второго

**my\_folder/** в stage не будет записываться целая папка с внутри лежащими файлами

**my\_folder / my\_file.txt** не будут записываться в stage только файл с указаным названием, который только лежит в указаной папке

**#** коментарии в файле .gitignore

# **GitHub**

**origin** название удальоного репозитория

**origin / master origin/name\_branch** названия удаленьоных веток в git

**git remote add origin url\_adress** устанавливает связь между локальным репозиторием и удаленным

**git --set-upstream origin master** создаем первую-главную master ветку в удаленом репозиторее (делается один раз вначале)

**git remote -v** показывает к каком удаленым репозитория я поключен

**git push origin name\_branch** пушим указаную ветку в удаленный репозиторий

**git pull** **origin name\_branch** перенос данных (коммитов - файлов - истории) из удаленого репозитория указаной ветки в локальный в такую же ветку

**git fetch** загрузка содержимого из удаленого репозитория (не производит слияние)

**git pull = git fetch + git merge**

**git clone url\_adress** копирует репозиторий и как папку помещает в текущий путь

**git clone url\_adress my\_name** копирует репозиторий и изменяет название главной папки

# **Подключение К Уже Существующей Ветке**

1. **git init**

2. **git remote add origin url\_adress**

3. **git pull origin master(как будет называться ветка)**

# **---------------- Doker --------------**