1). Скачиваем Git <https://git-scm.com/>

2). Установка:

А. Select Components – оставляем все как есть;

Б. Choosing the default editor used by Git - Use Vim…

В. Adjusting your PATH…. – путкт №3

Г. Choosing HTTPS… - далее

Е. Configuring the line.. - №2

Ё. Configuring the terminal – далее

Ж. Далее.

3). Запускаем Git Bash – Start -> Git -> Git Bash;

4). ***mkdir dev*** – создаем дирректорию

5). ***cd dev***

6). ***mkdir my\_project***  - создание проекта

7). ***cd my\_project***

8). ***git init***

9). ***vi index.html*** – создание файла и открытие его в редакторе **vim**

10). ***git status; git add . ; git commit -m ‘commit’***;

11). <https://gist.github.com/drucoder/025d435b132bdba7b51fa24ea5bc90d5> Ссылка на скрипт для отображения текущей ветки и для просмотра истории

a). копируем последнюю строчку (alias…);

b). ***vi ~/.bashrc***;

d). вставляем; Esc, Shift+Z, Shift+Z – выходим из **vim;**

e). ***exit*** – закрываем консоль;

f). открываем консоль, переходим в проект;

g). ***last*** – выводим **log** коммитов с красивым оформлением.

12) Синхронизация с GitHub:

a). ***ssh-keygen.exe*** – генерация ключей;

b). ***cat ~/.ssh/id\_rsa.pub*** – переходим по этому пути;

c). копируем в консоли **ssh-rsa …..;**

d). github -> мой профиль -> settings -> SSH and GPG keys;

e). **title** – обычно имя компьютера, **key** – вставляем ключ, удаляем пробелы **ADD rss…;**

f). github -> мой профиль -> Your profile -> Repositories -> New ….. …. …. **Create repository;**

g). *…or push an exiting repository from the command line* – копируем первую строчку;

h). ***git remote add origin*** [***git@github.com:iamfennics***](mailto:git@github.com:iamfennics)***.....*** – вставляем в **bash;**

i). ***git remote*** – проверяем. Должно вывести ***origin;***

j). ***git push -u origin master*** копируем на git вторую строчку;

k). вставляем в bash;

l). ***last*** – проверяем.

13) Извлечение с GitHub:

a). ***git pull origin***

|  |  |
| --- | --- |
| Командная строка | *Определение* |
| ls | *наличие файлов* |
| *git config --global user.name “usName”* | *Имя на для всех приложений* |
| git congig --global user.email “usEmail” | *Почта для всех приложений* |
| *git config --global user.name “usName”* | *Имя для отдельного проекта* |
| git congig --global user.email “usEmail” | *Почта для отдельного проекта* |
| git init | *создание репозитория* |
| git status | состояние ветки |
| gitk | UI-интерфейс |
| git add <file> - | добавлен на коммит-кандидат |
| git commit -m "Title of commit" | добавлен коммит |
| git commit -a –amend | Добавляет к последнему commit новые изменен. |
|  | Отменяем последний commit не трогая изменен. |
| git reset –hard HEAD^ | Отменяем последний commit |
| git log | просмотр истории комментов |
| git log --pretty=oneline | каждый в одну линию |
| git log --pretty=oneline --max-count=3 | три комита |
| git log --pretty=oneline --all | все |
| git log --pretty=oneline --author="authName" |  |
| git log --pretty=format:"%h - %s: %ad [ %an]" | "сокращ. хеш - комментарий: дата: [автор]" |
| git checkout f3443b | выбор комита по хешу |
| git reset HEAD <fileName> | отмена послдней индексации |
| git revert HEAD | удаление последнего комита |
| git revert f3443b | откат к определён. хешу - КОНФЛИКТ |
| git revert --abort | отмена этого отката |
| git branch | проверка наличия веток |
| git checkout -b “new branch” | создание новой ветки "newbrench" |
| git checkout master | Переход на нужную ветку |
| (master) git merge newBranch | Слияние ветки **newBranch** с веткой **master** |
| wq | Согласится с общим коммитом при КОНФЛИК |
| *Git rebase* | Слияние веток с нарушением истории для локал |
| *git clone https://github.com/accaunt/any-repos.git* | Клонрование репозитория с удалённого сервера |
| git remote add origin https://github.com/reposit.git | Добавление ссылки удаленки на локалку |
| (github) Settings -> Collaborators -> Add collab… | Добавление пользователя в репозиторий |
| git push -u origin master | Перенос ветки на удалёнку |
| git pull origin master | Перенос ветки c удалёнки |
| git pull –rebase origin master | Слияние изменений на local и server по-времени |
| git stash | Перенос с изменениями и сверху локал-изменеия |
|  |  |
|  |  |
|  |  |