**Система контроля версий**

[[](https://camo.githubusercontent.com/cac4fa3d223a7759153322482888f8ded4beae52/687474703a2f2f6d656469612e77336775792e636f6d2f77702d636f6e74656e742f75706c6f6164732f323031352f30322f6769742e6a7067)](https://camo.githubusercontent.com/cac4fa3d223a7759153322482888f8ded4beae52/687474703a2f2f6d656469612e77336775792e636f6d2f77702d636f6e74656e742f75706c6f6164732f323031352f30322f6769742e6a7067)

**Git** - распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. На сегодняшний день его поддерживает Джунио Хамано. Git поддерживает быстрое разделение и слияние версий, включает инструменты для визуализации и навигации по нелинейной истории разработки. Как и Darcs, BitKeeper, Mercurial, Bazaar и Monotone, Git предоставляет каждому разработчику локальную копию всей истории разработки, изменения копируются из одного репозитория в другой.

**Видео**

https://www.youtube.com/watch?v=SfSYrebXLDE&index=3&list=LLz-au3MVw-WWUQSCZsFR7Eg

**Установка**

Под [Windows](https://git-scm.com/download/win) или под Linux:

sudo apt-get update

sudo apt-get install git

**1 шаг: открываем терминал и устанавливаем первоначальные конфигурации**

git config --global user.name 'имя\_пользователя\_github'

git config --global user.email 'ваша\_электронная\_почта'

git config --global color.diff 'auto'

git config --global color.status 'auto'

git config --global color.branch 'auto'

git config --global credential.helper store

git config --global push.default simple

git config --global core.autocrlf false

git config --global core.eol lf

**2 шаг: клонируем репозиторий**

mkdir C:/develop/ #cоздаем папку на диске C (если Windows)

git clone https://github.com/user/title-repository # копируем свой репозиторий на компьютер

cd title-repository/ #заходим в локальный репозиторий

**3 шаг: обновляем изменения на гитхабе**

# производим изменения

# git отслеживает только изменения в файлах

# (создание, удаление, редактирование)

# git не видит созданную пустые директории

# чтобы директории были сохранены на сервере

# в пустых директориях необходимо присутствие

# новых, созданных или перемещенных в них файлов

git add . #производится индексирование файлов на предмет изменения в них (git add --all)

git commit -m "update" #фиксируем публикацию, комментируем изменения

git push -f #отправляем на сервер GitHub