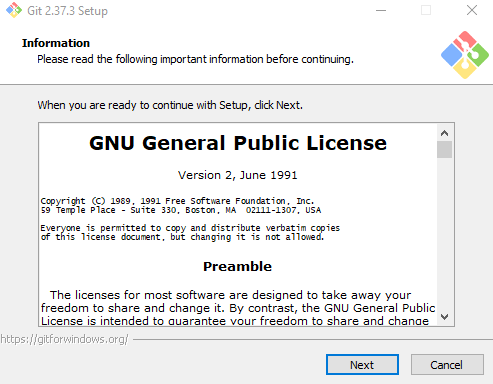
ИСПОЛЬЗОВАНЕ СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ GIT

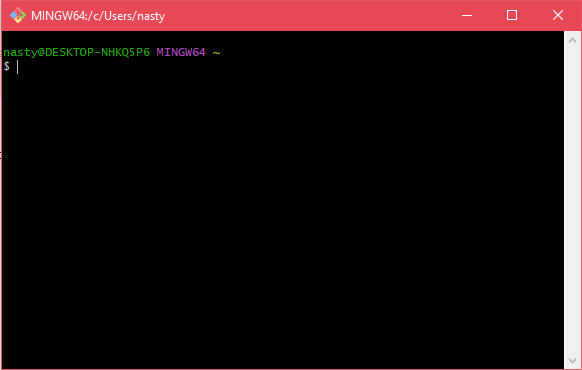
Данное краткое руководство демонстрирует основные команды в терминале Bash для Windows.

Скачиваем Bash по [ссылке](https://gitforwindows.org/).

Устанавливаем.



Так выглядит командная строка.



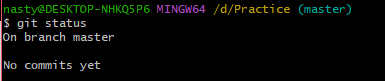
Вводим команду, которая выбирает каталог для репозитория:  
cd [Диск]:\[путь каталога]



Git init создает репозиторий.



Git status проверяет изменения коммитов.

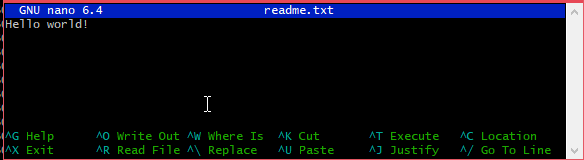


Touch [название файла] создает файл.

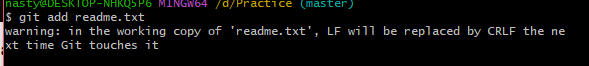


Nano [название файла] – начать редактирование файла.

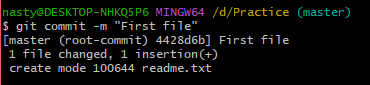
  
ctr+S – сохранить.  
ctr+X – закрыть редактор.

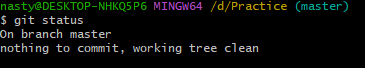


Так как этот файл не отслеживаемый, то его нужно как-то начать отслеживать и в этом нам поможет команда add, для этого вводим git add readme.txt

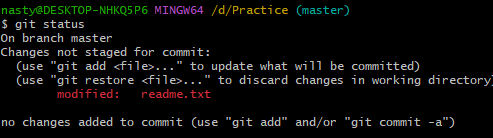


С помощью команды commit –m «коммитим».

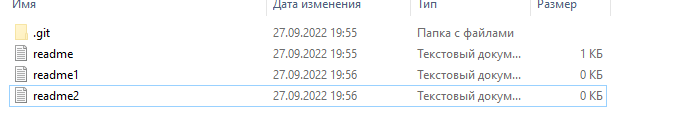




Проверив статус, увидим изменения красным цветом, значит нужно снова добавить его с помощью команды add

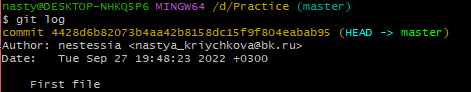


Создадим 2 копии текстового файла readme: readmefirst и readmesecond. Добавим их с помощью команды git add “\*.txt”

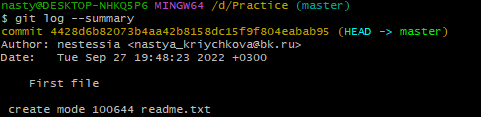




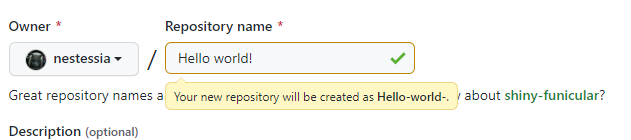
git log – история коммитов.



git log –summary – более детальная информация.



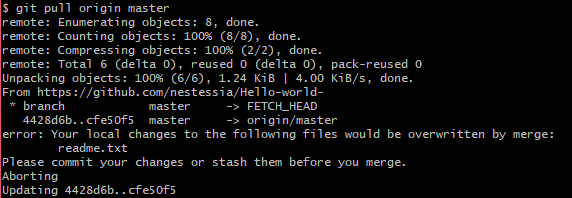
Чтобы выгрузить репозиторий на удаленный репозиторий, переходим в GitHub и создаем пустой репозиторий.



Допустим при работе в команде, кто-то выгрузил файлы с изменениями и вам нужно получить эти файлы обратно. Для этого используем команду git pull origin master. Изменим файл readme в удаленном репозитории и вернем файлы обратно.

Команда git remote add origin URL адрес поможет добавить нам удаленный репозиторий, а команда git push -u origin master поможет нам выгрузить файлы туда, master это ветка с которой мы работаем. Далее нужно авторизироваться через GitHub. После этого мы получим информацию о выгрузке. Также, зайдя на GitHub репозиторий, мы увидим наши файлы, причем указывается время создания файла, а не его появления на Гитхаб.

git pull origin master – показывает изменения файла



Другие полезные команды:

Также, мы можем проверить насколько отличны закоммиченные файлы и файлы, которые мы сами изменили. Для этой цели используется команда git diff HEAD. Head это указатель на закоммиченную версию файла.

Добавить каталог в репозиторий поможет команда mkdir [папка]

Команда git diff --staged позволит нам просмотреть изменения на стадии staged.

Команда git reset позволяет отменить изменения.

Файл можно вернуть в исходное состояние с помощью команды git checkout - -readme.txt

Команда git branch показывает на какой ветке мы сейчас находимся. Чтобы оказаться на другой ветке, будет использовать команду git checkout.

Команда rm -r folder поможет нам удалить каталог и файлы на этой ветке.

Для слияния используем команду git merge clean\_up.