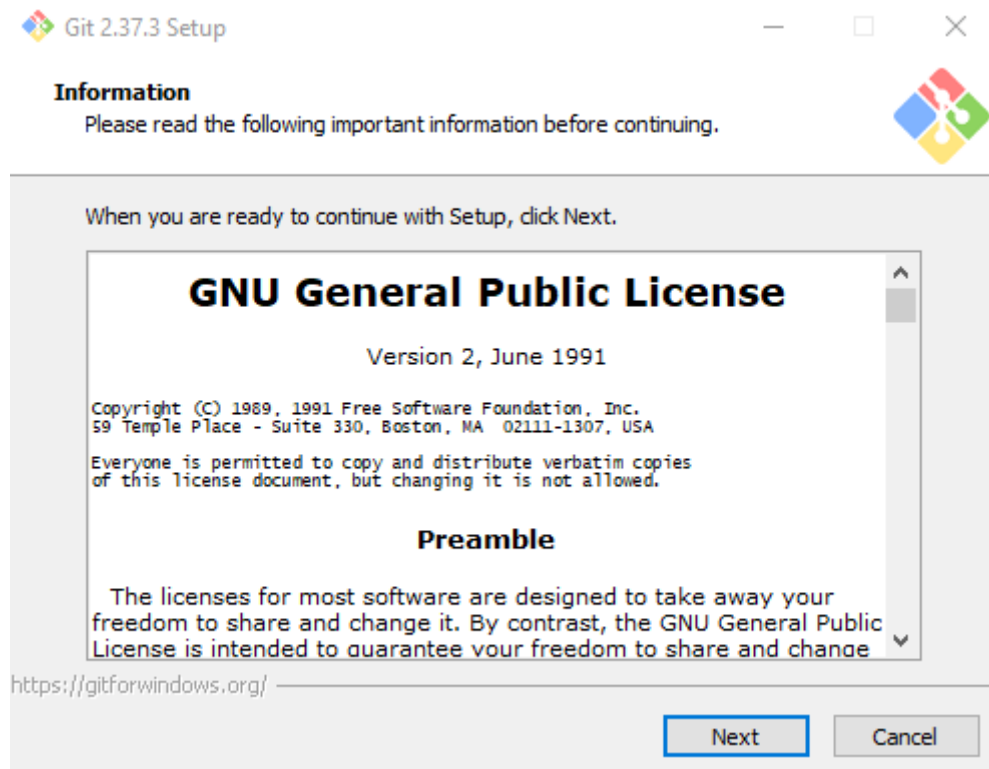


ИСПОЛЬЗОВАНИЕ СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ GIT

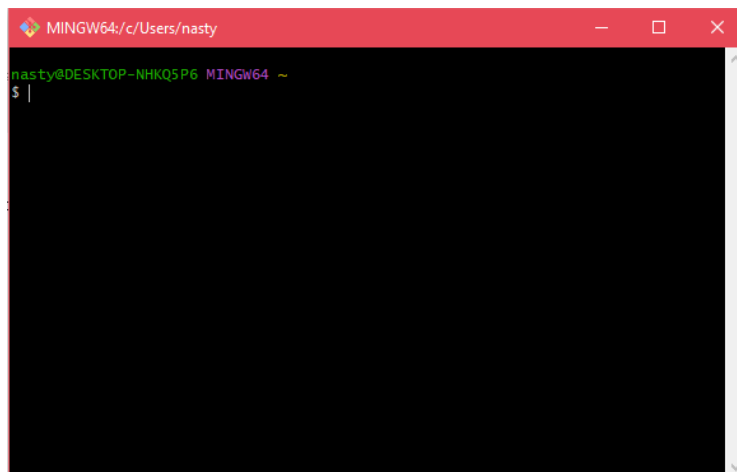
Данное краткое руководство демонстрирует основные команды в терминале Bash для Windows.

Скачиваем Bash по [ссылке](#).

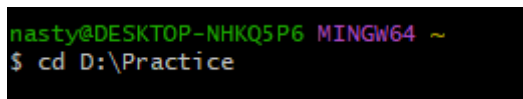
Устанавливаем.



Так выглядит командная строка.



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



Git init создает репозиторий.

```
nasty@DESKTOP-NHKQ5P6 MINGW64 /d/Practice
$ git init
Initialized empty Git repository in D:/Practice/.git/
```

Git status проверяет изменения коммитов.

```
nasty@DESKTOP-NHKQ5P6 MINGW64 /d/Practice (master)
$ git status
On branch master

No commits yet
```

Touch [название файла] создает файл.

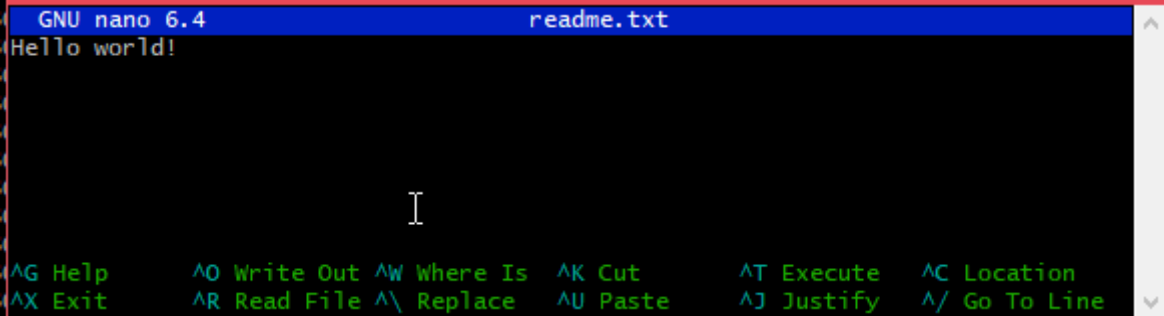
```
nasty@DESKTOP-NHKQ5P6 MINGW64 /d/Practice (master)
$ touch readme.txt
```

Nano [название файла] – начать редактирование файла.

```
nasty@DESKTOP-NHKQ5P6 MINGW64 /d/Practice (master)
$ nano readme.txt
```

ctr+S – сохранить.

ctr+X – закрыть редактор.

A screenshot of the GNU nano 6.4 text editor. The title bar shows 'GNU nano 6.4' and 'readme.txt'. The main text area contains 'Hello world!'. At the bottom, there is a status bar with various keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, and ^_ Go To Line.

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

```
nasty@DESKTOP-NHKQ5P6 MINGW64 /d/Practice (master)
$ git add readme.txt
warning: in the working copy of 'readme.txt', LF will be replaced by CRLF the ne
xt time Git touches it
```

С помощью команды commit -m «коммитим».

```
nasty@DESKTOP-NHKQ5P6 MINGW64 /d/Practice (master)
$ git commit -m "First file"
[master (root-commit) 4428d6b] First file
1 file changed, 1 insertion(+)
create mode 100644 readme.txt
```

```
nasty@DESKTOP-NHKQ5P6 MINGW64 /d/Practice (master)
$ git status
On branch master
nothing to commit, working tree clean
```

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

```
nasty@DESKTOP-NHKQ5P6 MINGW64 /d/Practice (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   readme.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Создадим 2 копии текстового файла readme: readmefirst и readmesecund. Добавим их с помощью команды git add "*.txt"

Имя	Дата изменения	Тип	Размер
.git	27.09.2022 19:55	Папка с файлами	
readme	27.09.2022 19:55	Текстовый докум...	1 КБ
readme1	27.09.2022 19:56	Текстовый докум...	0 КБ
readme2	27.09.2022 19:56	Текстовый докум...	0 КБ

```
nasty@DESKTOP-NHKQ5P6 MINGW64 /d/Practice (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   readme.txt
        new file:   readme1.txt
        new file:   readme2.txt
```

git log – история коммитов.

```
nasty@DESKTOP-NHKQ5P6 MINGW64 /d/Practice (master)
$ git log
commit 4428d6b82073b4aa42b8158dc15f9f804eabab95 (HEAD -> master)
Author: nestessia <nastya_kriychkova@bk.ru>
Date:   Tue Sep 27 19:48:23 2022 +0300

    First file
```

git log –summary – более детальная информация.


```
nasty@DESKTOP-NHKQ5P6 MINGW64 /d/Practice (master)
$ git log --summary
commit 4428d6b82073b4aa42b8158dc15f9f804eabab95 (HEAD -> master)
Author: nestessia <nastya_kriychkova@bk.ru>
Date:   Tue Sep 27 19:48:23 2022 +0300

    First file

    create mode 100644 readme.txt
```

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

Owner * Repository name *

 nestessia / Hello world! ✓

Great repository names are about [shiny-funicular?](#)

Your new repository will be created as **Hello-world-**.

Description (optional)

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 авторизоваться 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

```
$ git pull origin master
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 1.24 KiB | 4.00 KiB/s, done.
From https://github.com/nestessia/Hello-world-
* branch            master       -> FETCH_HEAD
   4428d6b..cfe50f5  master       -> origin/master
error: Your local changes to the following files would be overwritten by merge:
    readme.txt
Please commit your changes or stash them before you merge.
Aborting
Updating 4428d6b..cfe50f5
```

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`.