

Если git никогда ранее не использовался пользователем, то необходимо установить имя и электронную почту. Для этого требуется выполнить команды, показанные на рисунке 1.

```
ymerw@MINGW64 ~  
$ git config --global user.email "papaum@yandex.ru"  
  
ymerw@MINGW64 ~  
$ git config --global user.name "Sergey Shishkov"
```

Затем нужно указать параметры установки окончаний строк (рисунок 2).

```
Толик@DESKTOP-J475S8M MINGW64 ~  
$ git config --global core.autocrlf true  
  
Толик@DESKTOP-J475S8M MINGW64 ~  
$ git config --global core.safecrlf true  
  
Толик@DESKTOP-J475S8M MINGW64 ~  
$ |
```

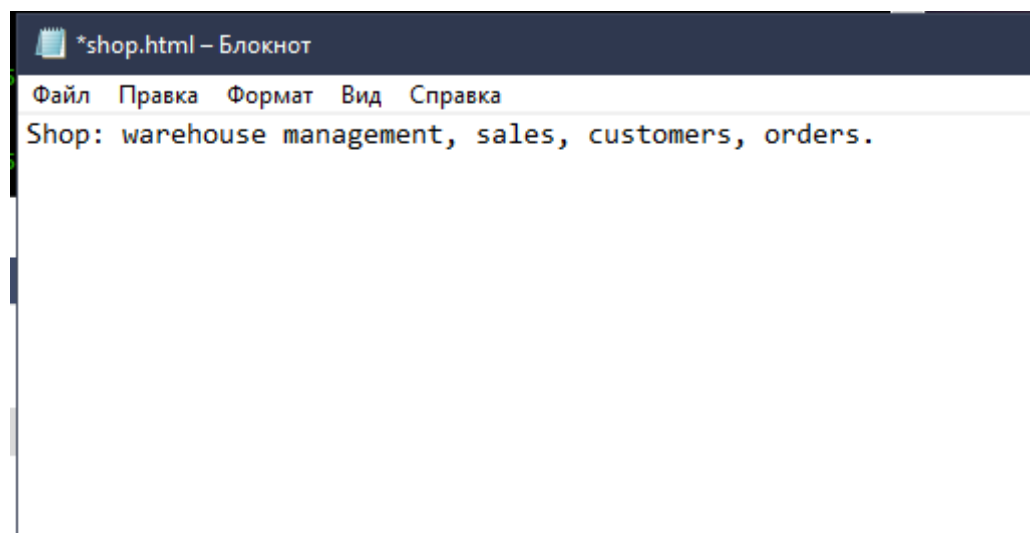
И последним пунктом идет установка отображения Unicode, показанная на рисунке 3.

```
Толик@DESKTOP-J475S8M MINGW64 ~  
$ git config --global core.quotePath off
```

Сначала нужно создать пустой каталог и внутри него файл hello.html (рисунок 4).

```
Толик@DESKTOP-J475S8M MINGW64 ~/shop  
$ mkdir shop  
  
Толик@DESKTOP-J475S8M MINGW64 ~/shop  
$ cd shop  
  
Толик@DESKTOP-J475S8M MINGW64 ~/shop/shop  
$ touch shop.html  
  
Толик@DESKTOP-J475S8M MINGW64 ~/shop/shop  
$
```

После этого в файл необходимо ввести данные, например, «Hello, world», как показано на рисунке 5.



```
*shop.html - Блокнот  
Файл  Правка  Формат  Вид  Справка  
Shop: warehouse management, sales, customers, orders.
```

Для создания репозитория используется команда git init (рисунок 6).

```

Толик@DESKTOP-J475S8M MINGW64 ~/shop/shop
$ git init
Initialized empty Git repository in C:/Users/Толик/shop/shop/.git/

Толик@DESKTOP-J475S8M MINGW64 ~/shop/shop (master)
$ A|

```

Для добавления страницы в репозиторий необходима команда `git add` (рисунок 7).

```

Толик@DESKTOP-J475S8M MINGW64 ~/shop/shop (master)
$ git add shop.html

Толик@DESKTOP-J475S8M MINGW64 ~/shop/shop (master)
$ git commit -m "first commit"
[master (root-commit) 32fa1b6] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 shop.html

Толик@DESKTOP-J475S8M MINGW64 ~/shop/shop (master)
$ |

```

Проверка состояния репозитория осуществляется с помощью команды `git status`. Если в репозитории хранится текущее состояние рабочего каталога и нет изменений, ожидающих записи, будет показано сообщение, как на рисунке 8.

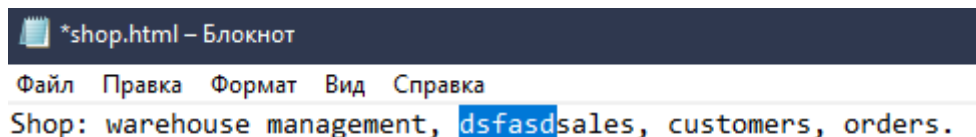
```

Толик@DESKTOP-J475S8M MINGW64 ~/shop/shop (master)
$ git status
On branch master
nothing to commit, working tree clean

Толик@DESKTOP-J475S8M MINGW64 ~/shop/shop (master)
$ |

```

Сначала необходимо внести изменения в файл (рисунок 9).



Если после предыдущего пункта осуществить проверку состояния репозитория, то будет показано данное сообщение (рисунок 10).

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git status
On branch master
nothing to commit, working tree clean

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (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:   shop.html

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

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$

```

Для того, чтобы проиндексировать изменения, нужно осуществить действия, показанные на рисунке 11.

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git add shop.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   shop.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$
```

После этого изменения файлы были проиндексированы. Это значит, что пока изменения не записаны в репозиторий. Если изменения позже не нужно будет фиксировать, то индексацию можно снять командой `git reset`.

Можно зафиксировать изменения отдельными коммитами. Как это сделать, показано на рисунках 12-14.

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git add a.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git add b.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git commit -m "changes for a and b"
[master 8d00a6d] changes for a and b
3 files changed, 1 insertion(+), 1 deletion(-)
create mode 100644 a.html
create mode 100644 b.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$
```

Рисунок 1 - Индексация и коммит для 2 файлов

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git add c.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git commit -m "unrelated change for c"
[master 94a2fb7] unrelated change for c
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 c.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ |
```

Рисунок 2 - Индексация и коммит для третьего файла

Для того, чтобы редактировать комментарий коммита, нужно использовать команду `git commit` без метки `-m`.

```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git commit
hint: Waiting for your editor to close the file...
[main 2021-01-31T08:05:41.046Z] update#setState idle
```

Рисунок 3 - Коммит изменений

После этого будет открыт редактор, в котором необходимо на 1 строке записать комментарий коммита (рисунок 16).

```
≡ COMMIT_EDITMSG ●
C: > Users > HELEN V > Desktop > hello > .git > ≡ COMMIT_EDITMSG
1 Added h2 tag
2 # Please enter the commit message for your changes. Lines
3 # with '#' will be ignored, and an empty message aborts t
4 #
5 # On branch master
6 # Changes to be committed:
7 #   modified:   hello.html
8 #
9
```

Рисунок 4 - Ввод комментария

После выхода из текстового редактора будет указано следующее сообщение (рисунок 17).

```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git commit
hint: Waiting for your editor to close the file...
[main 2021-01-31T08:05:41.046Z] update#setState idle
[main 2021-01-31T08:06:11.084Z] update#setState checking for updates
[main 2021-01-31T08:06:11.304Z] update#setState idle
[master c8f53f9] Added h2 tag
1 file changed, 2 insertions(+), 1 deletion(-)

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ |
```

Рисунок 5 - Коммит-сообщения

После этого еще раз нужно проверить состояние репозитория (рисунок 18).

```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
nothing to commit, working tree clean

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ |
```

Рисунок 6 - Проверка состояния

## Изменения, а не файлы

Для того, чтобы понять, что git фокусируется на изменениях в файле, а не на самом файле, можно проделать следующие действия.

Сначала нужно в файл hello.html добавить теги html и body (рисунок 19), а затем проиндексировать изменения.

```
hello.html X
C: > Users > HELEN V > Desktop > hello > hello.html > html > body > h2
1  <html>
2  <body>
3  <h1>Hello, world</h1>
4  <h2>Second line</h2>
5  </body>
6  </html>
```

Рисунок 7 - Добавление тегов html и body

Затем еще раз нужно добавить изменения в файл (добавить тег head), но изменения не индексировать (рисунок 20).

```
hello.html X
C: > Users > HELEN V > Desktop > hello > hello.html > html > head
1  <html>
2  <head>
3
4  </head>
5  <body>
6  <h1>Hello, world</h1>
7  <h2>Second line</h2>
8  </body>
9  </html>
```

Рисунок 8 - Добавление тега head

Далее нужно проверить статус. На рисунке 21 видно, что файл hello.html указан дважды: первое изменение проиндексировано и готово к коммиту, а второе – нет.

```
HELEN V@LAPTOP-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

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:   hello.html

HELEN V@LAPTOP-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$
```

Рисунок 9 - Проверка состояния

Далее надо произвести коммит проиндексированного изменения и затем еще раз проверить состояние (рисунок 22).

```
HELEN V@LAPTOP-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$ git commit -m "Add standard HTML tags"
[master cab488a] Add standard HTML tags
1 file changed, 6 insertions(+), 2 deletions(-)

HELEN V@LAPTOP-BPH65Q86 MINGW64 ~/Desktop/hello (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:   hello.html

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

HELEN V@LAPTOP-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$ |
```

Рисунок 10 - Коммит 1 изменения и проверка состояния



Нужно добавить второе изменение в индекс и затем проверить состояние (рисунок 23).

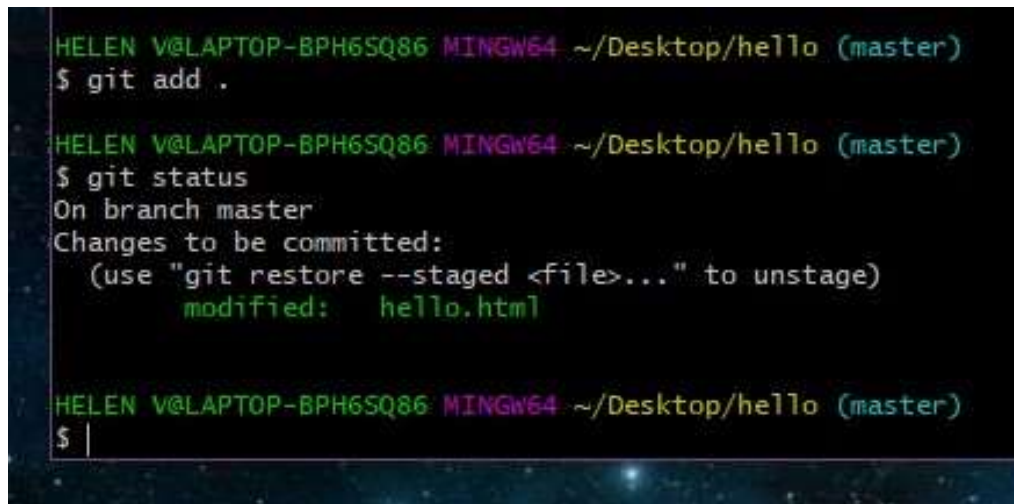
A terminal window with a dark background and a starry space pattern at the bottom. The prompt is 'HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)'. The user enters '\$ git add .' and then '\$ git status'. The output shows 'On branch master' and 'Changes to be committed: (use "git restore --staged <file>..." to unstage) modified: hello.html'. The prompt returns to '\$ |'.

Рисунок 11 - Индексация 2 изменения и проверка состояния

После этого нужно сделать коммит второго изменения (рисунок 24).

A terminal window with a dark background and a starry space pattern at the bottom. The prompt is 'HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)'. The user enters '\$ git commit -m "Add HTML header"'. The output shows '[master d653372] Add HTML header' and '1 file changed, 3 insertions(+)'.

Рисунок 12 - Коммит 2 изменения

Для того, чтобы просмотреть список произведенных изменений в проекте, используется команда git log (рисунок 1).

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git log
commit 94a2fb71f2b97753e9710e06582d60bd3745e1aa (HEAD -> master)
Author: Daniil <danik.aesen@yandex.ru>
Date: Tue Jun 3 11:33:09 2025 +0300

    unrelated change for c

commit 8d00a6d414574db74fad7e0b50cb7409c7bdbb25
Author: Daniil <danik.aesen@yandex.ru>
Date: Tue Jun 3 11:31:28 2025 +0300

    changes for a and b

commit 249c04fe08e526edb439fcb78993506f77a5ffb8
Author: Daniil <danik.aesen@yandex.ru>
Date: Tue Jun 3 11:24:23 2025 +0300

    first commit

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ |

```

На рисунке 1 была выведена полная история. Для того, чтобы увидеть однострочный формат используется команда `git log --pretty=oneline` (рисунок 2).

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git log --pretty=oneline
94a2fb71f2b97753e9710e06582d60bd3745e1aa (HEAD -> master) unrelated change for c
8d00a6d414574db74fad7e0b50cb7409c7bdbb25 changes for a and b
249c04fe08e526edb439fcb78993506f77a5ffb8 first commit

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$

```

Далее на рисунках 3-8 показано несколько вариантов вывода истории изменений.

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git log --pretty=oneline --max-count=2
94a2fb71f2b97753e9710e06582d60bd3745e1aa (HEAD -> master) unrelated change for c
8d00a6d414574db74fad7e0b50cb7409c7bdbb25 changes for a and b

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$

```

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git log --pretty=oneline --since='10 days ago'
94a2fb71f2b97753e9710e06582d60bd3745e1aa (HEAD -> master) unrelated change for c
8d00a6d414574db74fad7e0b50cb7409c7bdbb25 changes for a and b
249c04fe08e526edb439fcb78993506f77a5ffb8 first commit

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git log --pretty=oneline --since='12 days ago'
94a2fb71f2b97753e9710e06582d60bd3745e1aa (HEAD -> master) unrelated change for c
8d00a6d414574db74fad7e0b50cb7409c7bdbb25 changes for a and b
249c04fe08e526edb439fcb78993506f77a5ffb8 first commit

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ |

```



```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git log --pretty=oneline --until='5 minutes ago'
94a2fb71f2b97753e9710e06582d60bd3745e1aa (HEAD -> master) unrelated change for c
8d00a6d414574db74fad7e0b50cb7409c7bdbb25 changes for a and b
249c04fe08e526edb439fcb78993506f77a5ffb8 first commit
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git log --pretty=oneline --author='Daniil'
94a2fb71f2b97753e9710e06582d60bd3745e1aa (HEAD -> master) unrelated change for c
8d00a6d414574db74fad7e0b50cb7409c7bdbb25 changes for a and b
249c04fe08e526edb439fcb78993506f77a5ffb8 first commit
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git log --pretty=oneline --all
94a2fb71f2b97753e9710e06582d60bd3745e1aa (HEAD -> master) unrelated change for c
8d00a6d414574db74fad7e0b50cb7409c7bdbb25 changes for a and b
249c04fe08e526edb439fcb78993506f77a5ffb8 first commit
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git log --pretty=format: "%h %cd %s (%an)" --since='12 days ago'
fatal: ambiguous argument '%h %cd %s (%an)': unknown revision or path not in the work
ing tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ |
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git config --global alias.co checkout
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git config --global alias.ci commit
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git config --global alias.st status
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git config --global alias.br branch
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git config --global alias.type 'cat-file -t'
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git config --global alias.dump 'cat-file -p'
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ |
```

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ alias gs='git status'

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ gs
On branch master
nothing to commit, working tree clean

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$

```

Для того, чтобы вернуть рабочий каталог к предыдущему состоянию, можно использовать следующий способ: для начала нужно узнать хэши предыдущих версий, что можно сделать с помощью ранее заданного алиаса hist (рисунок 12).

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git hist
git: 'hist' is not a git command. See 'git --help'.

The most similar command is
    bisect

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$

```

Далее нужно выполнить команду git checkout с номером нужного хэша (достаточно первых 7 знаков). После этого можно просмотреть содержимое файла с помощью команды cat (рисунок 13).

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git checkout 9234a79
error: pathspec '9234a79' did not match any file(s) known to git

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ cat shop.html
hiii(/dsfs)

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$

```

Для возвращения к последней версии в ветке master (имя ветки по умолчанию) надо ввести команду git checkout master, что показано на рисунке 14.

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git checkout master
Already on 'master'

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ cat shop.html
hiii(/dsfs)

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$

```

Для создания тега используется команда git tag. На рисунке 15 показано, тегом ver1 была названа текущая версия страницы.

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git tag ver1

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$

```

Чтобы перейти к предыдущей версии, можно использовать символ «^», который означает «родитель».

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git checkout ver1^
Note: switching to 'ver1^'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 8d00a6d changes for a and b
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 ((8d00a6d...))
$ |

```

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 ((8d00a6d...))
$ git tag ver1-beta

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 ((ver1-beta))
$

```

Теперь с помощью тегов можно переключаться между версиями (рисунок 18).

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 ((ver1-beta))
$ git checkout ver1
Previous HEAD position was 8d00a6d changes for a and b
HEAD is now at 94a2fb7 unrelated change for c

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 ((ver1))
$ git checkout ver1-beta
Previous HEAD position was 94a2fb7 unrelated change for c
HEAD is now at 8d00a6d changes for a and b

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 ((ver1-beta))
$

```

Для просмотра всех тегов используется команда git tag (рисунок 19).

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 ((ver1-beta))
$ git tag
ver1
ver1-beta

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 ((ver1-beta))
$ |

```

Также можно просмотреть теги в логе, как показано на рисунке 20.

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 ((ver1-beta))
$ git hist master --all
git: 'hist' is not a git command. See 'git --help'.

The most similar command is
    bisect

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 ((ver1-beta))
$

```

Сначала нужно переключиться на последний коммит master (рисунок 21).

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 ((ver1-beta))
$ git checkout master
Previous HEAD position was 8d00a6d changes for a and b
Switched to branch 'master'

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ |
```

Сначала нужно переключиться на последний коммит master (рисунок 21).

```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello ((ver1-beta))
$ git checkout master
Previous HEAD position was cab488a Add standard HTML tags
Switched to branch 'master'
```

Рисунок 13 - Переключение на последний коммит

Далее для работы нужно внести изменение в файл (рисунок 22).

```
<> hello.html ●
C: > Users > HELEN V > Desktop > hello > <> hello.html > html > body
1  <html>
2      <head>
3
4      </head>
5      <body>
6          <h1>Hello, world</h1>
7          <h2>Second line</h2>
8  <!-- Это плохой комментарий (очень плохой!) -->
9      </body>
10 </html>
```

Рисунок 14 - Внесение изменения в файл

После выполнения команды git status будет показано, что есть не проиндексированное изменение (рисунок 23).

```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git st
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:   hello.html

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

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$
```

Рисунок 15 - Не проиндексированное изменение

Для переключения в версию файла без изменений используется команда `git checkout hello.html` (рисунок 24). Команда `git status` покажет, что не было произведено изменений, не зафиксированных в каталоге.

```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git checkout hello.html
Updated 1 path from the index

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git st
On branch master
nothing to commit, working tree clean

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ cat hello.html
<html>
  <head>

  </head>
  <body>
    <h1>Hello, world</h1>
    <h2>Second line</h2>
  </body>
</html>
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
```

Рисунок 16 - Возвращение к версии

### Отмена проиндексированных изменений (перед коммитом)

Для того, чтобы научиться отменять проиндексированные изменения, сначала нужно внести ненужное изменение в файл (рисунок 25). После этого производится индексация (рисунок 26).

```
<> hello.html X
C: > Users > HELEN V > Desktop > hello > <> hello.html > html > head
1  <html>
2  <head>
3  | <!-- Еще один ненужный коммент -->
4  </head>
5  <body>
6  | <h1>Hello, world</h1>
7  | <h2>Second line</h2>
8  </body>
9  </html>
```

Рисунок 17 - Внесение ненужного изменения

```

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git st
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

```

Рисунок 18 - Индексация изменения

Для отмены индексация изменения используется команда `git reset HEAD hello.html` (рисунок 27). Команда `reset` сбрасывает буферную зону к HEAD и очищает ее от проиндексированных изменений. Но для удаления ненужного по-прежнему используется команда `git checkout` (рисунок 28).

```

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git reset HEAD hello.html
Unstaged changes after reset:
M       hello.html

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git st
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:   hello.html

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

```

Рисунок 19 - Очистка буферной зоны

```

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git checkout hello.html
Updated 1 path from the index

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git st
On branch master
nothing to commit, working tree clean

```

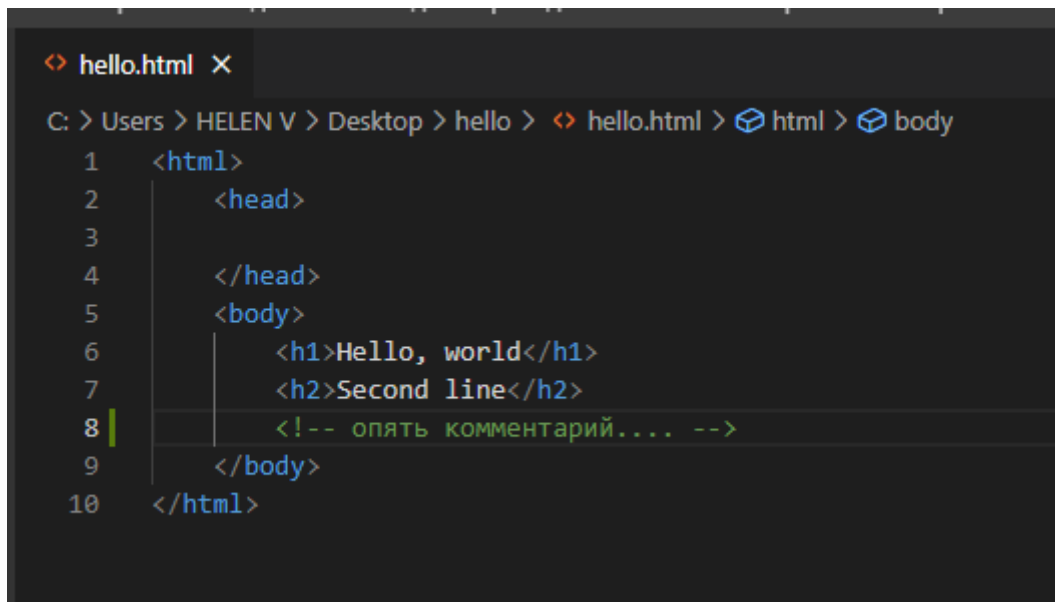
Рисунок 20 - Удаление ненужных изменений

## Отмена коммитов

Для отмены коммита можно использовать способ создания нового коммита, отменяющего изменения.

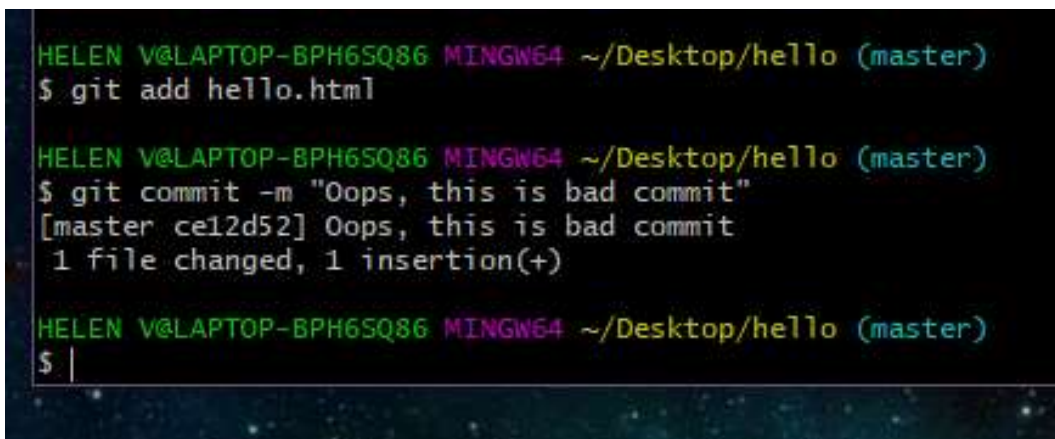
Для начала надо внести изменение, проиндексировать его и записать коммит (рисунки 29-30).





```
hello.html X
C: > Users > HELEN V > Desktop > hello > hello.html > html > body
1 <html>
2   <head>
3
4   </head>
5   <body>
6     <h1>Hello, world</h1>
7     <h2>Second line</h2>
8     <!-- опять комментарий.... -->
9   </body>
10  </html>
```

Рисунок 21 - Внесение изменения в файл

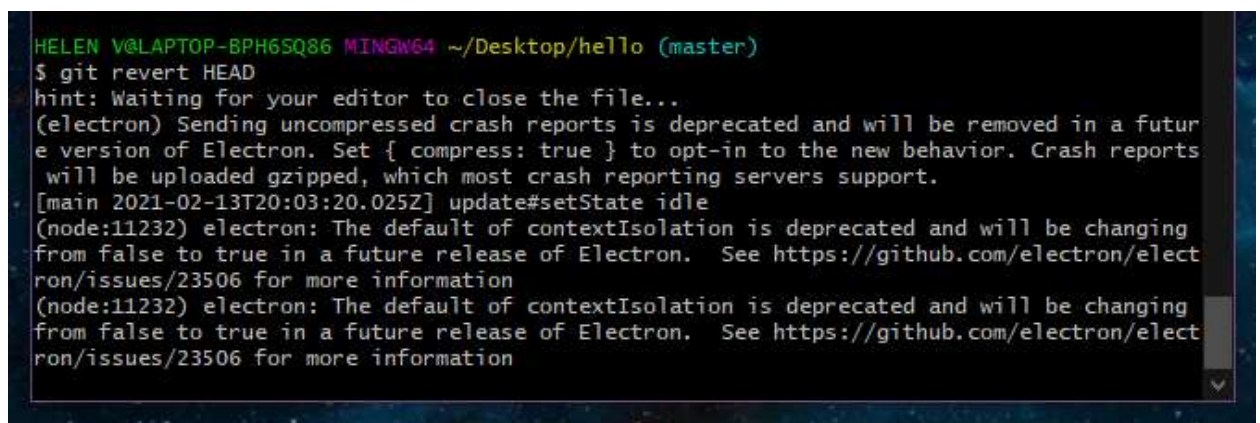


```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git commit -m "Oops, this is bad commit"
[master ce12d52] Oops, this is bad commit
1 file changed, 1 insertion(+)
```

Рисунок 22 - Индексация и коммит

Для создания коммита, который удалит ненужные изменения, используется команда `git revert HEAD` (рисунок 31). После этого будет открыт редактор, в котором можно отредактировать коммит сообщение (рисунок 32), затем надо сохранить файл и закрыть редактор (рисунок 33).



```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git revert HEAD
hint: Waiting for your editor to close the file...
(electron) Sending uncompressed crash reports is deprecated and will be removed in a future
version of Electron. Set { compress: true } to opt-in to the new behavior. Crash reports
will be uploaded gzipped, which most crash reporting servers support.
[main 2021-02-13T20:03:20.025Z] update#setState idle
(node:11232) electron: The default of contextIsolation is deprecated and will be changing
from false to true in a future release of Electron. See https://github.com/electron/elect
ron/issues/23506 for more information
(node:11232) electron: The default of contextIsolation is deprecated and will be changing
from false to true in a future release of Electron. See https://github.com/electron/elect
ron/issues/23506 for more information
```

Рисунок 23 - Выполнение команды `git revert`

```
COMMIT_EDITMSG X
C: > Users > HELEN V > Desktop > hello > .git > COMMIT_EDITMSG
1  Revert "Oops, this is bad commit"
2
3  This reverts commit ce12d52b0101f30fefaf7c66203593dbfac57ae54.
4
5  # Please enter the commit message for your changes. Lines starting
6  # with '#' will be ignored, and an empty message aborts the commit.
7  #
8  # On branch master
9  # Changes to be committed:
10 #   modified:   hello.html
11 #
12
```

Рисунок 24 - Коммит сообщение в редакторе

```
(node:11232) electron: The default of contextIsolation is deprecated and will be changing
from false to true in a future release of Electron. See https://github.com/electron/elect
ron/issues/23506 for more information
[main 2021-02-13T20:03:50.027Z] update#setState checking for updates
[main 2021-02-13T20:03:50.085Z] update#setState idle
[master e04a55f] Revert "Oops, this is bad commit"
1 file changed, 1 deletion(-)

HELEN V@LAPTOP-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$ |
```

Рисунок 25 - Редактор закрыт

При проверке лога будут показаны все коммиты, в том числе и отмененные (рисунок 34).

```
1 file changed, 1 deletion(-)
HELEN V@LAPTOP-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$ git hist
* e04a55f 2021-02-13 | Revert "Oops, this is bad commit" (HEAD -> master) [Elena Verstova]
* ce12d52 2021-02-13 | Oops, this is bad commit [Elena Verstova]
* d653372 2021-01-31 | Add HTML header (tag: ver1) [Elena Verstova]
* cab488a 2021-01-31 | Add standard HTML tags (tag: ver1-beta) [Elena Verstova]
* c8f53f9 2021-01-31 | Added h2 tag [Elena Verstova]
* cf86455 2021-01-31 | Unrelated change for v [Elena Verstova]
* 00106a5 2021-01-31 | Changes for a and b [Elena Verstova]
* 9234a79 2021-01-31 | First Commit [Elena Verstova]

HELEN V@LAPTOP-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$
```

Рисунок 26 - Все коммиты при просмотре лога

Перед удалением коммита последний из них нужно отметить тегом, чтобы не потерять его (рисунок 35).

```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git tag oooops

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git hist
* e04a55f 2021-02-13 | Revert "Oops, this is bad commit" (HEAD -> master, tag: oooops) [Elena Verstova]
* ce12d52 2021-02-13 | Oops, this is bad commit [Elena Verstova]
* d653372 2021-01-31 | Add HTML header (tag: ver1) [Elena Verstova]
* cab488a 2021-01-31 | Add standard HTML tags (tag: ver1-beta) [Elena Verstova]
* c8f53f9 2021-01-31 | Added h2 tag [Elena Verstova]
* cf86455 2021-01-31 | Unrelated change for v [Elena Verstova]
* 00106a5 2021-01-31 | Changes for a and b [Elena Verstova]
* 9234a79 2021-01-31 | First Commit [Elena Verstova]

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$
```

Рисунок 27 - Отметка тегом коммита

Для сброса коммитов используется команда `git reset --hard ver1` (рисунок 36). Она сбрасывает ветку до версии с тегом `ver1`.

```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git reset --hard ver1
HEAD is now at d653372 Add HTML header

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git hist
* d653372 2021-01-31 | Add HTML header (HEAD -> master, tag: ver1) [Elena Verstova]
* cab488a 2021-01-31 | Add standard HTML tags (tag: ver1-beta) [Elena Verstova]
* c8f53f9 2021-01-31 | Added h2 tag [Elena Verstova]
* cf86455 2021-01-31 | Unrelated change for v [Elena Verstova]
* 00106a5 2021-01-31 | Changes for a and b [Elena Verstova]
* 9234a79 2021-01-31 | First Commit [Elena Verstova]

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$
```

Рисунок 28 - Сброс коммита

Но при просмотре лога с помощью команды `git hist --all` отмененные коммиты по-прежнему будут показываться, так как они всё еще находятся в репозитории (рисунок 37).

```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git hist --all
* e04a55f 2021-02-13 | Revert "Oops, this is bad commit" (tag: oooops) [Elena Verstova]
* ce12d52 2021-02-13 | Oops, this is bad commit [Elena Verstova]
* d653372 2021-01-31 | Add HTML header (HEAD -> master, tag: ver1) [Elena Verstova]
* cab488a 2021-01-31 | Add standard HTML tags (tag: ver1-beta) [Elena Verstova]
* c8f53f9 2021-01-31 | Added h2 tag [Elena Verstova]
* cf86455 2021-01-31 | Unrelated change for v [Elena Verstova]
* 00106a5 2021-01-31 | Changes for a and b [Elena Verstova]
* 9234a79 2021-01-31 | First Commit [Elena Verstova]

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$
```

Рисунок 29 - Сброшенные коммиты находятся по-прежнему в репозитории

## Удаление тега

Так как тег «oooops» больше не нужен, его и коммиты, на которые он указывает, можно удалить с помощью команды `git tag -d` (рисунок 38).



```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git tag -d oooops
Deleted tag 'oooops' (was e04a55f)

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git hist --all
* d653372 2021-01-31 | Add HTML header (HEAD -> master, tag: ver1) [Elena Verstova]
* cab488a 2021-01-31 | Add standard HTML tags (tag: ver1-beta) [Elena Verstova]
* c8f53f9 2021-01-31 | Added h2 tag [Elena Verstova]
* cf86455 2021-01-31 | Unrelated change for v [Elena Verstova]
* 00106a5 2021-01-31 | Changes for a and b [Elena Verstova]
* 9234a79 2021-01-31 | First Commit [Elena Verstova]

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$
```

Рисунок 30 - Удаление тега

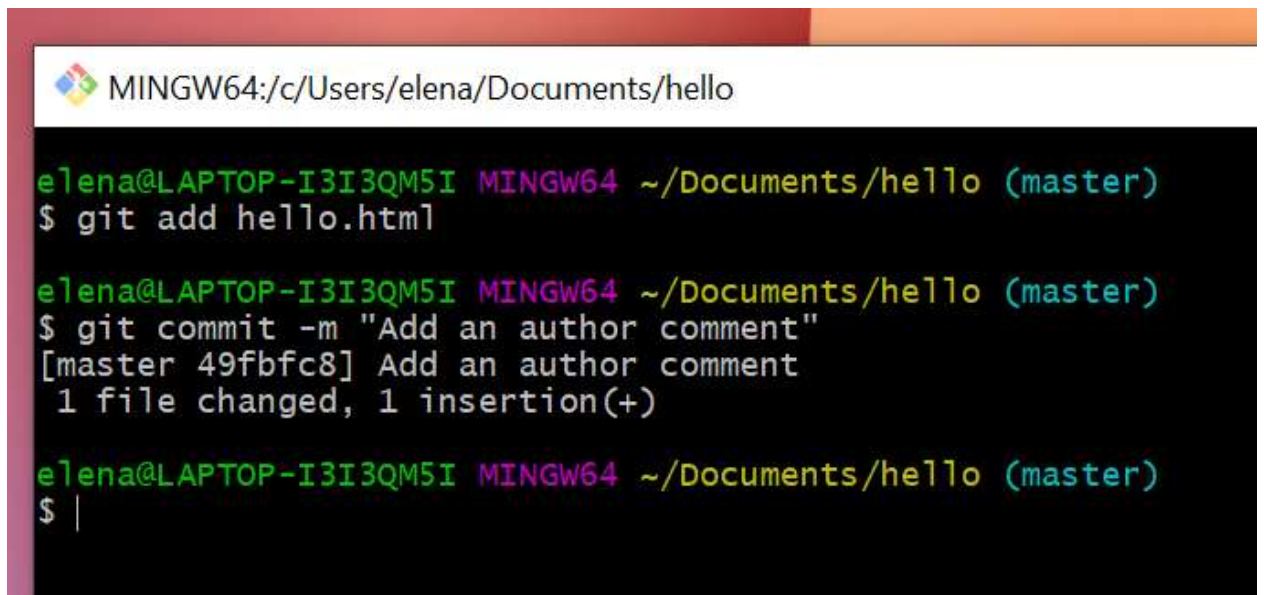
## Работа с Git

### Внесение изменений в коммиты

Для начала будет создан коммит, в который позже будут внесены изменения. На рисунках 1 и 2 происходит добавление комментария в файл hello.html и его индексация и коммит.

```
<> hello.html X
C: > Users > elena > Documents > hello > <> hello.html > ...
1 | <!-- Author: Elena Verstova -->
2 | <html>
3 |   <head>
4 |
5 |   </head>
6 |   <body>
7 |     <h1>Hello, world</h1>
8 |     <h2>The cake is a lie.</h2>
9 |   </body>
10| </html>
```

Рисунок 31 - Добавление комментария в файл



```
MINGW64:/c/Users/elena/Documents/hello

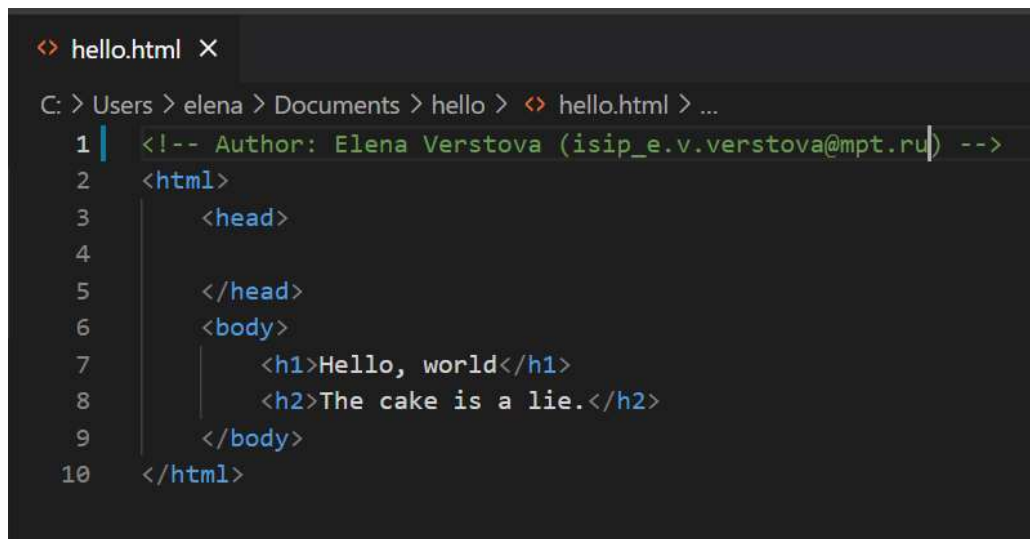
elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git add hello.html

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git commit -m "Add an author comment"
[master 49fbfc8] Add an author comment
1 file changed, 1 insertion(+)

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ |
```

Рисунок 32 - Индексация и коммит

Далее необходимо добавить электронную почту в комментарий (рисунок 3).



```
<> hello.html X
C: > Users > elena > Documents > hello > <> hello.html > ...
1 | <!-- Author: Elena Verstova (isip_e.v.verstova@mpt.ru) -->
2 | <html>
3 |   <head>
4 |
5 |   </head>
6 |   <body>
7 |     <h1>Hello, world</h1>
8 |     <h2>The cake is a lie.</h2>
9 |   </body>
10| </html>
```

Рисунок 33 - Добавление электронной почты

Но для того, чтобы не создавать отдельный коммит ради электронной почты, можно изменить предыдущий так, как показано на рисунке 4.

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git add hello.html

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git commit --amend -m "Add an author/email comment"
[master aefcfd7] Add an author/email comment
Date: Sat Feb 20 22:12:56 2021 +0300
1 file changed, 1 insertion(+)

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$
```

Рисунок 34 - Индексация и изменение коммита

При просмотре истории можно будет заметить, что последний коммит был изменен.

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git hist
* aefcfd7 2021-02-20 | Add an author/email comment (HEAD -> master) [ElenaVerstova]
* d243b6b 2021-02-17 | Added standard HTML tags (tag: ver1) [ElenaVerstova]
* 8de3e80 2021-02-17 | Added tag h2 (tag: ver1-beta) [ElenaVerstova]
* ce52614 2021-02-17 | Unrelated change for v [ElenaVerstova]
* e1f2276 2021-02-17 | Changes for a and b [ElenaVerstova]
* f302a0c 2021-02-17 | First Commit [ElenaVerstova]
```

Рисунок 35 - Последний коммит изменен

## Перемещение файлов

Для перемещения файлов в пределах репозитория используются команды, показанные на рисунке 6. После выполнения данных команды git индексирует эти изменения (удаление файла hello.html и создание файла lib/hello.html).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ mkdir lib

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git mv hello.html lib

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    hello.html -> lib/hello.html
```

Рисунок 36 - Перемещение файла

Далее надо осуществить коммит данного перемещения.



```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git commit -m "Moved hello.html to lib"
[master b5bd827] Moved hello.html to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ |
```

Рисунок 37 – Коммит перемещения

## Подробнее о структуре

Необходимо добавить еще один файл в репозиторий. Это будет файл index.html с кодом, показанным на рисунке 8.

```
<> index.html ●
C: > Users > elena > Documents > hello > <> index.html > html
1  <html>
2      <body>
3          <iframe src="lib/hello.html" width="200" height="200"></iframe>
4      </body>
5  </html>
```

Рисунок 38 - Содержимое файла index.html

Далее нужно проиндексировать и закоммитить файл (рисунок 9).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git add index.html

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git commit -m "Added index.html"
[master c2bcb76] Added index.html
1 file changed, 5 insertions(+)
create mode 100644 index.html

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ |
```

Рисунок 39 - Индексация и коммит

При открытии файла index.html будет виден кусок страницы hello.html (рисунок 10).

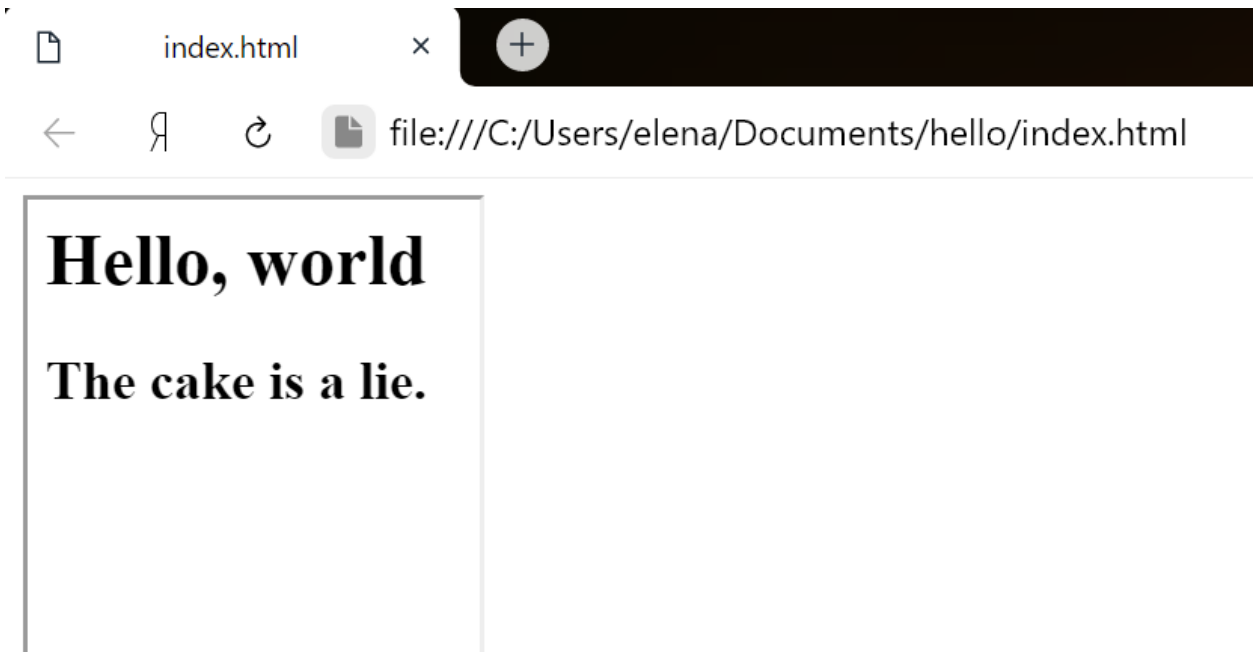


Рисунок 40 - Файл index.html, открытый в браузере

## Каталог .git

Чтобы посмотреть структуру каталога .git необходимо выполнить команду, показанную на рисунке 11.

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ ls -C .git
COMMIT_EDITMSG  ORIG_HEAD  description  index  logs/  refs/
HEAD           config     hooks/       info/  objects/

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$
```

Рисунок 41 - Содержание каталога .git

При аналогичном просмотре каталога objects можно будет увидеть множество каталогов с именами из 2 символов (рисунок 12). Имена каталогов являются первыми двумя буквами хэша.

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ ls -C .git/objects
03/  16/  34/  49/  57/  6e/  88/  ae/  bc/  d2/  e1/  fd/
07/  1e/  37/  4e/  5c/  74/  8d/  af/  c2/  d8/  f3/  info/
12/  24/  44/  50/  67/  7f/  a0/  b5/  ce/  db/  f9/  pack/

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$
```

Рисунок 42 - Содержание каталога objects

При просмотре содержимого любого из каталогов будут показаны файлы, названия которых состоят из 38 символов (рисунок 13).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ ls -C .git/objects
03/  16/  34/  49/  57/  6e/  88/  ae/  bc/  d2/  e1/  fd/
07/  1e/  37/  4e/  5c/  74/  8d/  af/  c2/  d8/  f3/  info/
12/  24/  44/  50/  67/  7f/  a0/  b5/  ce/  db/  f9/  pack/

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ ls -C .git/objects/8d
e3e8034c7ba4cc85a9ae26844a5c51be5f5e66

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$
```

Рисунок 43 - Просмотр каталога 8d

Далее требуется посмотреть файл конфигурации с помощью команды cat (рисунок 14).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
```

Рисунок 44 - Просмотр файла конфигурации

На рисунке 14 показан просмотр файлов в подкаталоге tags и веток в каталоге heads.

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ ls .git/refs
heads/  tags/

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ ls .git/refs/heads
master

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ ls .git/refs/tags
ver1  ver1-beta

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ cat .git/refs/tags/ver1
d243b6b28cd79a79b0d27214ef82f2032d3de69c

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ |
```

Рисунок 45 - Просмотр файлов и веток

Файл HEAD содержит ссылку на текущую ветку (рисунок 16).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ cat .git/HEAD
ref: refs/heads/master

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ |
```

Рисунок 46 - Содержимое файла HEAD

## Работа с объектами git

Для начала необходимо просмотреть последний коммит (рисунок 17).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git hist --max-count=1
* c2bcb76 2021-02-20 | Added index.html (HEAD -> master) [ElenaVerstova]

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ |
```

Рисунок 47 - Последний коммит

Далее надо использовать хэш последнего коммита, используя команды `cat-file -p` и `cat-file -t` (рисунок 18) для просмотра объекта коммита. Также вместо длинных команд можно использовать сокращенные `type` и `dump`, если данные команды были заданы как алиасы.

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git cat-file -t c2bcb76
commit

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git cat-file -p c2bcb76
tree 4e279a99e24ae23e1966cfa86f79adb89c9f5985
parent b5bd82753296f765e6aa0ebb5d60c3e0a93cac3c
author ElenaVerstova <isip_e.v.verstova@mpt.ru> 1613849667 +0300
committer ElenaVerstova <isip_e.v.verstova@mpt.ru> 1613849667 +0300

Added index.html
```

Рисунок 48 - Просмотр объекта коммита

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git type c2bcb76
commit

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git dump c2bcb76
tree 4e279a99e24ae23e1966cfa86f79adb89c9f5985
parent b5bd82753296f765e6aa0ebb5d60c3e0a93cac3c
author ElenaVerstova <isip_e.v.verstova@mpt.ru> 1613849667 +0300
committer ElenaVerstova <isip_e.v.verstova@mpt.ru> 1613849667 +0300

Added index.html

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ |
```

Рисунок 49 - Использование алиасов

Для просмотра дерева каталогов необходимо использовать его хэш (рисунок 20).



```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git cat-file -p 4e279a9
100644 blob 1240583197c9a4507a2fb0d59eb1a82886844e57    a.html
100644 blob af60016690b87c1f1757db52ac1baf43ff3419a8    b.html
100644 blob 03ee26631ef36c173c9c1a9f7e51dd419f72482d    index.html
040000 tree 34c23e296c4e5b2207ddeb4a4aaf565abdf9e0d0    lib
100644 blob 1eded30425a18ead4e57682660b74c8645761b51    v.html

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ |
```

Рисунок 50 - Просмотр дерева каталогов

Затем нужно посмотреть каталог lib (рисунок 21).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git cat-file -p 34c23e296
100644 blob 07978d6ac75191ca3ce7aaca139e95e951fe230b    hello.html

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$
```

Рисунок 51 - Просмотр каталога lib

И затем требуется вывести содержимое файла hello.html (рисунок 22).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git cat-file -p 07978d6a
<!-- Author: Elena Verstova (isip_e.v.verstova@mpt.ru) -->
<html>
  <head>

  </head>
  <body>
    <h1>Hello, world</h1>
    <h2>The cake is a lie.</h2>
  </body>
</html>
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$
```

Рисунок 52 - Вывод содержимого файла hello.html



Аналогичным образом можно просмотреть содержимое файла, каким оно было в самом первом коммите, как показано на рисунке 23. Для этого требуется использовать лишь нужные хэши.

```
elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git hist
* c2bcb76 2021-02-20 | Added index.html (HEAD -> master, tag: ver2) [ElenaVerstova]
* b5bd827 2021-02-20 | Moved hello.html to lib [ElenaVerstova]
* aefcfd7 2021-02-20 | Add an author/email comment [ElenaVerstova]
* d243b6b 2021-02-17 | Added standard HTML tags (tag: ver1) [ElenaVerstova]
* 8de3e80 2021-02-17 | Added tag h2 (tag: ver1-beta) [ElenaVerstova]
* ce52614 2021-02-17 | Unrelated change for v [ElenaVerstova]
* e1f2276 2021-02-17 | Changes for a and b [ElenaVerstova]
* f302a0c 2021-02-17 | First Commit [ElenaVerstova]

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git cat-file -p f302a0c
tree d8a800db068656c3aafc962f7a438c987b49dd60
author ElenaVerstova <isip_e.v.verstova@mpt.ru> 1613562110 +0300
committer ElenaVerstova <isip_e.v.verstova@mpt.ru> 1613562110 +0300

First Commit

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git cat-file -p d8a800d
100644 blob dbe9dba55ea8fd4d5be3868b015e044be0848ec5    hello.html

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git cat-file -p dbe9dba
Hello, world
elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$
```

Рисунок 53 - Просмотр содержимого файла при первом коммите

```
Microsoft Windows [Version 10.0.22631.4169]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\store management\Desktop\store>git add
git: 'add' is not a git command. See 'git --help'.

The most similar command is
  add

C:\store management\Desktop\store>git add .
C:\store management\Desktop\store>git commit
hint: Waiting for your editor to close the file... unix2dos: converting file C:/store management/Desktop/store/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/store management/Desktop/store/.git/COMMIT_EDITMSG to Unix format...
[detached HEAD 981a313] cwe
# files changed, 2 insertions(+)
create mode 100644 cwe.link
create mode 100644 management/bye.html.txt
create mode 100644 store.html.txt
create mode 100644 store/store.html.txt
C:\store management\Desktop\store>git add .
C:\store management\Desktop\store>git commit --amend
hint: Waiting for your editor to close the file... unix2dos: converting file C:/store management/Desktop/store/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/store management/Desktop/store/.git/COMMIT_EDITMSG to Unix format...
[master 09d78dc] cwe
Date: Tue Jun 3 19:07:05 2025 +0300
3 files changed, 3 insertions(+), 2 deletions(-)
create mode 100644 store.html

C:\store management\Desktop\store>git hist
* 09d78dc 2025-06-03 | cwe (HEAD -> master) [Temur Pantsulaya]
* 0c62ca6 2025-06-03 | add.html [Temur]

C:\store management\Desktop\store>git add .
C:\store management\Desktop\store>git commit
hint: Waiting for your editor to close the file... unix2dos: converting file C:/store management/Desktop/store/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/store management/Desktop/store/.git/COMMIT_EDITMSG to Unix format...
Aborting commit due to empty commit message.

C:\store management\Desktop\store>git commit
hint: Waiting for your editor to close the file... unix2dos: converting file C:/store management/Desktop/store/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/store management/Desktop/store/.git/COMMIT_EDITMSG to Unix format...
[master 32df700] prepred legends
3 files changed, 8 insertions(+), 8 deletions(-)
rename sell.html => sell/sell.html (100%)
rename sell.txt => sell/sell.txt (100%)
rename sell_inf.txt => sell/sell_inf.txt (100%)
```

```

C:\store management\Desktop\store>echo hello,world >"T.txt"
C:\store management\Desktop\store>git add .
C:\store management\Desktop\store>git commit
hint: Waiting for your editor to close the file... unix2dos: converting file C:/store management/Desktop/store/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/store management/Desktop/store/.git/COMMIT_EDITMSG to Unix format...
[master f0b8b0f] cwww
1 file changed, 1 insertion(+)
create mode 100644 T.txt
C:\store management\Desktop\store>git status
On branch master
nothing to commit, working tree clean
C:\store management\Desktop\store>git log --3
fatal: unrecognized argument: --3
C:\store management\Desktop\store>git log -3
commit f0b8b0fa7a9d97c446e407ed40ce649d3cdc02dc (HEAD -> master)
Author: Temur Pantsulaya <pantsulaya07@bk.ru>
Date: Tue Jun 3 16:58:56 2025 +0300

    cwww

commit 32d0780c272ac0271d6ba85e345b4cdf29eab2888
Author: Temur Pantsulaya <pantsulaya07@bk.ru>
Date: Tue Jun 3 16:45:57 2025 +0300

    prepad legenda

commit e9d78dc53cc31ad10af4076081992f16ddb04f0d
Author: Temur Pantsulaya <pantsulaya07@bk.ru>
Date: Tue Jun 3 14:47:05 2025 +0300

    cww
C:\store management\Desktop\store>dir
Том в устройстве C имеет метку Local Disk
Серийный номер тома: EC5E-5C36

```

```

C:\store management\Desktop\store>dir
Том в устройстве C имеет метку Local Disk
Серийный номер тома: EC5E-5C36

Содержимое папки C:\store management\Desktop\store

03.06.2025 16:49 <DIR>      .
03.06.2025 16:19 <DIR>      ..
03.06.2025 16:45 <DIR>      sells
03.06.2025 16:40              13 store.html
03.06.2025 16:40              15 T.txt
                2 файлов      28 байт
                3 папок      195 472 101 376 байт свободно

C:\store management\Desktop\store>type T.txt
hello,world

C:\store management\Desktop\store>type -?
Не удается найти указанный файл.

C:\store management\Desktop\store>C:\> notepad somefile.txt
"C:\\" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\store management\Desktop\store>T.txt
C:\store management\Desktop\store>

```

```
MINGW64/c:/store management/Desktop/store
Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ ls -C .git
COMMITMESSAGE  config      index  objects/
COMMIT_EDITMSG description info/  refs/
HEAD           hooks/      logs/

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ ls -C .git/objects
00/ 1d/ 34/ 6c/ 95/ bf/ d2/ f8/
06/ 29/ 39/ 6d/ 98/ c4/ e6/ info/
0c/ 2d/ 3b/ 71/ a4/ c9/ e9/ pack/
19/ 32/ 4b/ 7f/ ad/ ca/ f2/

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ git hist
* f8b8b8f 2025-06-03 | cwww (HEAD -> master) [Temur]
```

```
MINGW64/c:/store management/Desktop/store
$ ls -C .git/objects
00/ 1d/ 34/ 6c/ 95/ bf/ d2/ f8/
06/ 29/ 39/ 6d/ 98/ c4/ e6/ info/
0c/ 2d/ 3b/ 71/ a4/ c9/ e9/ pack/
19/ 32/ 4b/ 7f/ ad/ ca/ f2/

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ git hist
* f8b8b8f 2025-06-03 | cwww (HEAD -> master) [Temur Pantsulaya]
* 32df700 2025-06-03 | prepod legenda [Temur Pantsulaya]
* e9d70dc 2025-06-03 | cwe [Temur Pantsulaya]
* 0c62ca6 2025-06-03 | add html [Temur]
```

Для начала необходимо создать ветку style с помощью команды `git checkout -b style` (рисунок 1).

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git checkout -b style
fatal: a branch named 'style' already exists

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git status
On branch style
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   lib/shop.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ |

```

Затем нужно создать файл стилей (рисунок 2) и внести в него код, показанный на рисунке 3.

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ touch lib/shop.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ |

```

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git add lib/shop.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git commit -m "added css stylesheet"
[style 908470f] added css stylesheet
1 file changed, 1 insertion(+), 1 deletion(-)

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$

```

Далее требуется изменить основную страницу hello.html и закоммитить изменения (рисунки 5-6).

```

shop.html ● index.html
C: > Users > Толик > Desktop > shop1 > lib > shop.html > html
1  <!--Autor: Daniil (danik.arsen@yandex.ru)-->
2  <html>
3      <head>
4          <link rel="stylesheet" type="text/css" media="all" href="style.css">
5      </head>
6      <body>
7          <h1>hiiii(/dsfs)</h1>
8          <h2>fsdsdfdfs</h2>
9          <h3>!_@(#cx.vg,m)</h3>
10     </body>
11     </html>

```

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git add lib/shop.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git commit -m "shop.html uses style.css"
On branch style
nothing to commit, working tree clean

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$

```

```

1  <html>
2      <head>
3          <link rel="stylesheet" type="text/css" media="all" href="lib/style.css">
4      </head>
5      <body>
6          <iframe src="lib/hello.html" width="200" height="200"></iframe>
7      </body>
8  </html>

```

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git add index.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git commit -m "update index.html"
[style d2fe1db] update index.html
1 file changed, 3 insertions(+)

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ |

```

После выполнения предыдущих действий была создана новая ветка style с 3 коммитами.

## Навигация по веткам

При просмотре истории, как на рисунке 9, можно увидеть, что теперь в проекте 2 ветки.

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git hist --all
git: 'hist' is not a git command. See 'git --help'.

The most similar command is
    bisect

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ |

```

Для переключения на ветку master используется команда `git checkout master` (рисунок 10). После переключения на нужную ветку при выводе файла `hello.html` можно увидеть, что изменения отсутствуют (по причине того, что они закоммичены в другой ветке).



```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ cat lib/shop.html
<!--Autor: Daniil (danik.arsen@yandex.ru)-->
<html>
  <head>
    <link rel="stylesheet" type="text/css" media="all" href="style.css">
  </head>
  <body>
<h1>hiiii (/dsfs)</h1>
<h2>fsdsdfdfs</h2>
<h3>!_@(#cx.vg,m)</h3>
</body>
</html>
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ |

```

При переключении на ветку style файл hello.html будет иметь другое содержание (рисунок 11).

```

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git checkout style
Switched to branch 'style'

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style)
$ cat lib/hello.html
<!-- Author: Elena Verstova (isip_e.v.verstova@mpt.ru) -->
<html>
  <head>
    <link rel="stylesheet" type="text/css" media="all" href="style.css">
  </head>
  <body>
    <h1>Hello, world</h1>
    <h2>The cake is a lie.</h2>
  </body>
</html>
elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style)
$ |

```

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git add read.txt

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$

```

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git merge master
Already up to date.

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$

```



```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style)
$ git merge master
hint: Waiting for your editor to close the file...
(electron) Sending uncompressed crash reports is deprecated and will be removed
in a future version of Electron. Set { compress: true } to opt-in to the new beh
avior. Crash reports will be uploaded gzipped, which most crash reporting server
s support.
[main 2021-02-24T11:53:30.275Z] update#setState idle
(node:45768) electron: The default of contextIsolation is deprecated and will be
changing from false to true in a future release of Electron. See https://githu
b.com/electron/electron/issues/23506 for more information
(node:45768) electron: The default of contextIsolation is deprecated and will be
changing from false to true in a future release of Electron. See https://githu
b.com/electron/electron/issues/23506 for more information
[main 2021-02-24T11:54:00.303Z] update#setState checking for updates
[main 2021-02-24T11:54:00.790Z] update#setState idle
Merge made by the 'recursive' strategy.
 README | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README
```

Для того, чтобы создать конфликт необходимо перейти в ветку master и внести изменения в файл hello.html (рисунки 19-21).

```
олик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git checkout master
error: Your local changes to the following files would be overwritten by checkou
t:
    lib/shop.html
Please commit your changes or stash them before you switch branches.
Aborting

олик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
```

```
1  <!--Autor: Daniil (danik.arsen@yandex.ru)-->
2  <html>
3  <head>
4      <!-- no style-->
5      <link rel="stylesheet" type="text/css" media="all" href="style.css">
6  </head>
7  <body>
8  <h1>hiiii(/dsfs)</h1>
9  <h2>fsdsdfdfs</h2>
10 <h3>!_@(#cx.vg,m)</h3>
11 </body>
12 </html>
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git add lib/shop.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git commit -m 'life is great'
[style c518b21] life is great
 2 files changed, 10 insertions(+), 3 deletions(-)
 create mode 100644 read.txt

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$
```

После выполнения предыдущих действий при просмотре веток можно будет увидеть конфликт в виде, как на рисунке 22.

```
eLena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git hist --all
* 9dadf96 2021-02-24 | Life is great! (HEAD -> master) [ElenaVerstova]
* 6efc8fb 2021-02-24 | Merge branch 'master' into style (style) [ElenaVerstova]
* 3559bed 2021-02-24 | Added README [ElenaVerstova]
* 2a8fdf7 2021-02-24 | Update index.html [ElenaVerstova]
* 09ae13e 2021-02-24 | Hello.html uses style.css [ElenaVerstova]
* ea17c64 2021-02-24 | Added css stylesheet [ElenaVerstova]
* c2bcb76 2021-02-20 | Added index.html (tag: ver2) [ElenaVerstova]
* b5bd827 2021-02-20 | Moved hello.html to lib [ElenaVerstova]
* aefcfd7 2021-02-20 | Add an author/email comment [ElenaVerstova]
* d243b6b 2021-02-17 | Added standard HTML tags (tag: ver1) [ElenaVerstova]
* 8de3e80 2021-02-17 | Added tag h2 (tag: ver1-beta) [ElenaVerstova]
* ce52614 2021-02-17 | Unrelated change for v [ElenaVerstova]
* e1f2276 2021-02-17 | Changes for a and b [ElenaVerstova]
* f302a0c 2021-02-17 | First Commit [ElenaVerstova]
```

При попытке объединить ветку style с master будет показана ошибка из-за конфликта, как показано на рисунке 23.

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git checkout style
Already on 'style'

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git merge master
Already up to date.

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$
```

При открытии файла `hello.html` конфликт будет показан (рисунок 24).

```

C:\Users> elena > Documents > hello > lib > hello.html > html > head > ?
1  <!-- Author: Elena Verstova (isip_e.v.verstova@yandex.ru) -->
2  <html>
3  |   <head>
4  |   |   <<<<<< HEAD (Current Change)
5  |   |   |   <link rel="stylesheet" type="text/css" media="all" href="style.css">
6  |   |   =====
7  |   |   |   <!-- no style -->
8  |   |   >>>>>> master (Incoming Change)
9  |   |   </head>
10 |   </body>
11 |   <h1>Hello, world. Life is great</h1>
12 |   <h2>The cake is a lie.</h2>
13 |   </body>
14 </html>

```

Рисунок 54 - Просмотр файла hello.html при наличии конфликта

Чтобы решить конфликт, нужно внести изменения вручную (рисунок 25).

```
hello.html ●
C: > Users > elena > Documents > hello > lib > <> hello.html > html > head
1  <!-- Author: Elena Verstova (isip_e.v.verstova@mpt.ru) -->
2  <html>
3      <head>
4          <link rel="stylesheet" type="text/css" media="all" href="style.css">
5      </head>
6      <body>
7          <h1>Hello, world. Life is great</h1>
8          <h2>The cake is a lie.</h2>
9      </body>
10 </html>
```

Рисунок 55 - Решение конфликта вручную

Затем следует произвести индексацию и коммит (рисунок 26).

```
elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style|MERGING)
$ git add lib/hello.html

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style|MERGING)
$ git commit -m "Merged master fixed conflict."
[style 945ce49] Merged master fixed conflict.
```

Рисунок 56 - Индексация и коммит

Для начала необходимо создать ветку style с помощью команды git checkout -b style (рисунок 1).

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git checkout -b style
fatal: a branch named 'style' already exists

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git status
On branch style
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   lib/shop.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ |
```

Затем нужно создать файл стилей (рисунок 2) и внести в него код, показанный на рисунке 3.

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ touch lib/shop.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ |
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git add lib/shop.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git commit -m "added css stylesheet"
[style 908470f] added css stylesheet
1 file changed, 1 insertion(+), 1 deletion(-)

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$
```

Далее требуется изменить основную страницу hello.html и закоммитить изменения (рисунки 5-6).

```
<> shop.html ● <> index.html
C: > Users > Толик > Desktop > shop1 > lib > <> shop.html > html
1  <!--Autor: Daniil (danik.arsen@yandex.ru)-->
2  <html>
3      <head>
4          <link rel="stylesheet" type="text/css" media="all" href="style.css">
5      </head>
6      <body>
7          <h1>hiiii(/dsfs)</h1>
8          <h2>fsdsdfdfs</h2>
9          <h3>!_@(#cx.vg,m)</h3>
10     </body>
11 </html>
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git add lib/shop.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git commit -m "shop.html uses style.css"
On branch style
nothing to commit, working tree clean

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$
```

```
1  <html>
2      <head>
3          <link rel="stylesheet" type="text/css" media="all" href="lib/style.css">
4      </head>
5      <body>
6          <iframe src="lib/hello.html" width="200" height="200"></iframe>
7      </body>
8  </html>
```

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git add index.html

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git commit -m "update index.html"
[style d2fe1db] update index.html
1 file changed, 3 insertions(+)

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ |

```

После выполнения предыдущих действий была создана новая ветка style с 3 коммитами.

## Навигация по веткам

При просмотре истории, как на рисунке 9, можно увидеть, что теперь в проекте 2 ветки.

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git hist --all
git: 'hist' is not a git command. See 'git --help'.

The most similar command is
    bisect

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ |

```

Для переключения на ветку master используется команда `git checkout master` (рисунок 10). После переключения на нужную ветку при выводе файла `hello.html` можно увидеть, что изменения отсутствуют (по причине того, что они закоммичены в другой ветке).

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ cat lib/shop.html
<!--Autor: Daniil (danik.arsen@yandex.ru)-->
<html>
  <head>
    <link rel="stylesheet" type="text/css" media="all" href="style.css">
  </head>
  <body>
<h1>hiiii(/dsfs)</h1>
<h2>fsdsdfdfs</h2>
<h3>!_@(#cx.vg,m)</h3>
  </body>
</html>
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ |

```

При переключении на ветку style файл `hello.html` будет иметь другое содержание (рисунок 11).



```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git checkout style
Switched to branch 'style'

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style)
$ cat lib/hello.html
<!-- Author: Elena Verstova (isip_e.v.verstova@mpit.ru) -->
<html>
  <head>
    <link rel="stylesheet" type="text/css" media="all" href="style.css">
  </head>
  <body>
    <h1>Hello, world</h1>
    <h2>The cake is a lie.</h2>
  </body>
</html>
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style)
$ |
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git add read.txt
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git merge master
Already up to date.
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$
```

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style)
$ git merge master
hint: Waiting for your editor to close the file...
(electron) Sending uncompressed crash reports is deprecated and will be removed
in a future version of Electron. Set { compress: true } to opt-in to the new beh
avior. Crash reports will be uploaded gzipped, which most crash reporting server
s support.
[main 2021-02-24T11:53:30.275Z] update#setState idle
(node:45768) electron: The default of contextIsolation is deprecated and will be
changing from false to true in a future release of Electron. See https://githu
b.com/electron/electron/issues/23506 for more information
(node:45768) electron: The default of contextIsolation is deprecated and will be
changing from false to true in a future release of Electron. See https://githu
b.com/electron/electron/issues/23506 for more information
[main 2021-02-24T11:54:00.303Z] update#setState checking for updates
[main 2021-02-24T11:54:00.790Z] update#setState idle
Merge made by the 'recursive' strategy.
 README | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README
```

Для того, чтобы создать конфликт необходимо перейти в ветку master и внести изменения в файл hello.html (рисунки 19-21).



При попытке объединить ветку style с master будет показана ошибка из-за конфликта, как показано на рисунке 23.

```
Толик@DESKTOP-J47558M MINGW64 ~/desktop/shop1 (style)
$ git checkout style
Already on 'style'

Толик@DESKTOP-J47558M MINGW64 ~/desktop/shop1 (style)
$ git merge master
Already up to date.

Толик@DESKTOP-J47558M MINGW64 ~/desktop/shop1 (style)
$
```

При открытии файла hello.html конфликт будет показан (рисунок 24).

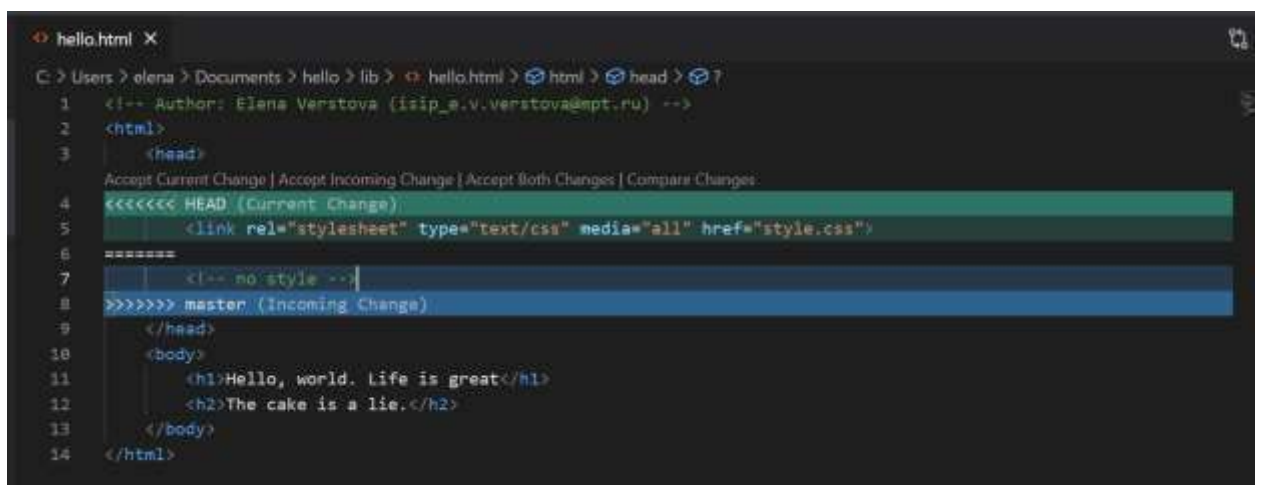


Рисунок 57 - Просмотр файла hello.html при наличии конфликта

Чтобы решить конфликт, нужно внести изменения вручную (рисунок 25).

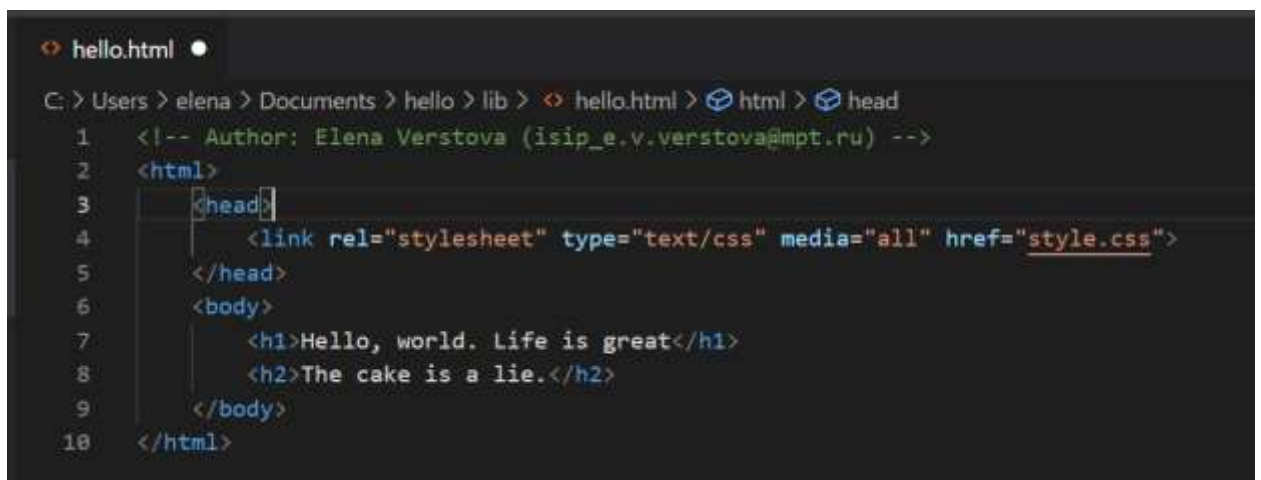


Рисунок 58 - Решение конфликта вручную

Затем следует произвести индексацию и коммит (рисунок 26).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style|MERGING)
$ git add lib/hello.html

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style|MERGING)
$ git commit -m "Merged master fixed conflict."
[style 945ce49] Merged master fixed conflict.
```

Рисунок 59 - Индексация и коммит

## Работа с Git

Сначала необходимо внести изменения в оригинальный репозиторий. Для этого нужно перейти в данный репозиторий (рисунок 1).

```
Толик@DESKTOP-J475S8M MINGW64 ~
$ cd desktop/shop1

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$
```

Далее надо внести изменения в файл README (рисунок 2) и затем произвести индексацию и коммит (рисунок 3).

```
≡ README ×

C: > Users > elena > Documents > hello > ≡ README
1   This is the Hello World example from the Git tutorial
2 | (changed in original)
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git add read.txt

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git commit -m "changed read.txt in original repo"
On branch style
nothing to commit, working tree clean
```

Далее требуется перейти в клонированный репозиторий и извлечь изменения с помощью команды `git fetch` (рисунок 4) и просмотреть историю (рисунок 5).

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ cd ../cloned_chop1

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/cloned_chop1 (style)
$ git fetch

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/cloned_chop1 (style)
$

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/cloned_hello (master)
$ git hist --all
* 5906087 2021-03-10 | Changed README in original repo (origin/master, origin/HEAD) [ElenaVerstova]
* 9dadf96 2021-02-24 | Life is great! (HEAD -> master, origin/style) [ElenaVerstova]
* 3559bed 2021-02-24 | Added README [ElenaVerstova]
* c2bcb76 2021-02-20 | Added index.html (tag: ver2) [ElenaVerstova]
* b5bd827 2021-02-20 | Moved hello.html to lib [ElenaVerstova]
* aefcfd7 2021-02-20 | Add an author/email comment [ElenaVerstova]
* d243b6b 2021-02-17 | Added standard HTML tags (tag: ver1) [ElenaVerstova]
* 8de3e80 2021-02-17 | Added tag h2 (tag: ver1-beta) [ElenaVerstova]
* ce52614 2021-02-17 | Unrelated change for v [ElenaVerstova]
* e1f2276 2021-02-17 | Changes for a and b [ElenaVerstova]
* f302a0c 2021-02-17 | First Commit [ElenaVerstova]
```

Рисунок 60 - Просмотр истории

При попытке вывести содержимое файла `README` можно увидеть, что изменения не были внесены (рисунок 6).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/cloned_hello (master)
$ cat README
This is the Hello world example from the Git tutorial
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/cloned_hello (master)
$
```

Рисунок 61 - Вывод содержимого файла `README`

Далее нужно слить извлеченные изменения в ветку `master` (рисунок 7).

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/cloned_chop1 (style)
$ git merge origin/master
Already up to date.

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/cloned_chop1 (style)
$
```

И после выполнения предыдущего действия при выводе `README` можно будет увидеть последние изменения (рисунок 8).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/cloned_hello (master)
$ cat README
This is the Hello world example from the Git tutorial
(changed in original)
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/cloned_hello (master)
$
```

Рисунок 62 - Вывод содержимого файла `README`



Также существует команда, объединяющая функции git fetch и git merge, которая показана на рисунке 9.

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/cloned_chop1 (style)
$ git pull
Already up to date.

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/cloned_chop1 (style)
$
```

Далее требуется добавить локальную ветку, которая будет отслеживать удаленную ветку (рисунок 10).

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/cloned_chop1 (style)
$ git branch --track style origin/style
fatal: a branch named 'style' already exists

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/cloned_chop1 (style)
$ git branch -a
* style
  remotes/origin/HEAD -> origin/style
  remotes/origin/master
  remotes/origin/style

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/cloned_chop1 (style)
$ git hist --max-count=2
git: 'hist' is not a git command. See 'git --help'.

The most similar command is
    bisect

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/cloned_chop1 (style)
$ |
```

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/cloned_chop1 (style)
$ cd ..

Толик@DESKTOP-J475S8M MINGW64 ~/desktop
$ git clone --bare shop shop.git
fatal: repository 'shop' does not exist

Толик@DESKTOP-J475S8M MINGW64 ~/desktop
$ ls shop.git
ls: cannot access 'shop.git': No such file or directory

Толик@DESKTOP-J475S8M MINGW64 ~/desktop
$
```

Для добавления удаленного репозитория используется команда, показанная на рисунке 12.

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop
$ cd shop1

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git remote add shared ../shop.git

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$
```

Затем требуется научиться отправлять изменения в удаленный репозиторий. Для этого сначала надо внести изменения, проиндексировать и произвести коммит (рисунок 13-14).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git daemon --verbose --export-all --base-path=.
[111720] Ready to rumble
```

Рисунок 63 - Настройка сервера

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git clone ~/Documents/hello.git network_hello
Cloning into 'network_hello'...
done.

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ cd network_hello

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello/network_hello (master)
$ ls
README  a.html  b.html  index.html  lib/  v.html
```

Рисунок 64 - Клонирование проекта

## Работа с Git

Для того, чтобы просмотреть список настроенных удалённых репозиторий, необходимо запустить команду `git remote` (рисунок 1).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents
$ git clone https://github.com/schacon/ticgit
Cloning into 'ticgit'...
remote: Enumerating objects: 1857, done.
Receiving object remote: Total 1857 (delta 0), reused 0 (delta 0), pack-reused 1857
Receiving objects: 100% (1857/1857), 334.06 KiB | 253.00 KiB/s, done.
Resolving deltas: 100% (837/837), done.

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents
$ cd ticgit

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ git remote
origin

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ |
```

Рисунок 65 – Клонирование репозитория и просмотр удаленных репозиторий

Можно указать ключ `-v`, чтобы просмотреть адреса для чтения и записи, привязанные к репозиторию (рисунок 2).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$
```

Рисунок 66 - Просмотр удаленных репозитория с ключом -v

Для добавления удаленного репозитория с новым именем используется команда git remote add (рисунок 3).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ git remote add pb https://github.com/paulboone/ticgit

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)
pb https://github.com/paulboone/ticgit (fetch)
pb https://github.com/paulboone/ticgit (push)
```

Рисунок 67 - Добавление удаленного репозитория

После задания имени репозиторию впоследствии его можно использовать вместо указания полного пути (рисунок 4).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ git fetch pb
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Total 43 (delta 22), reused 22 (delta 22), pack-reused 21
Unpacking objects: 100% (43/43), 5.99 KiB | 16.00 KiB/s, done.
From https://github.com/paulboone/ticgit
* [new branch]      master      -> pb/master
* [new branch]      ticgit      -> pb/ticgit

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$
```

Рисунок 68 - Использование имени вместо пути

Для получения данных из удалённых проектов используется команда git fetch (рисунок 5).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ git fetch ~/Documents/cloned_hello
remote: Enumerating objects: 39, done.
remote: Counting objects: 100% (39/39), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 39 (delta 12), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (39/39), 3.42 KiB | 10.00 KiB/s, done.
From C:/Users/elena/Documents/cloned_hello
* branch          HEAD      -> FETCH_HEAD

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ |
```

Рисунок 69 - Получение данных из удаленных проектов

Для отправки изменений в удаленный репозиторий используется команда `git push` (рисунок 6).

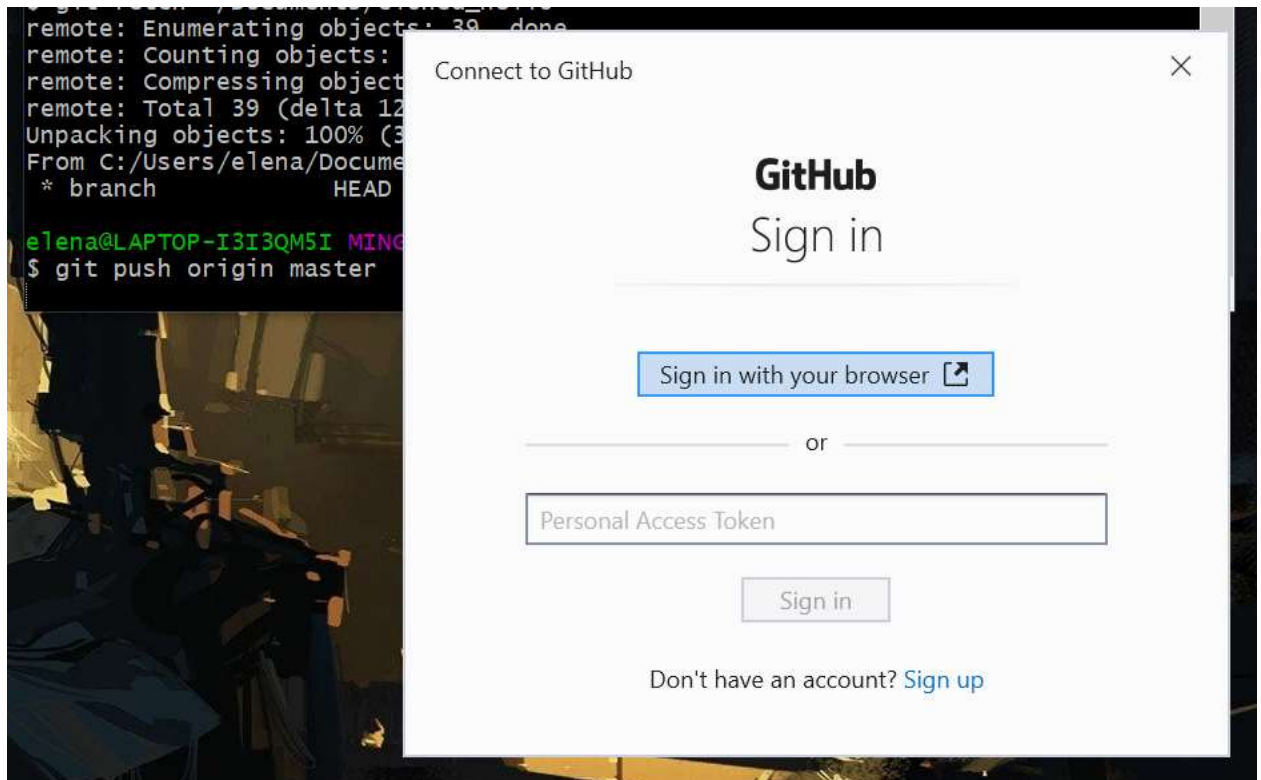


Рисунок 70 - Отправка изменений в удаленный репозиторий

Для получения информации об одном из удалённых репозиториях, можно использовать команду `git remote show` (рисунок 7).



```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/schacon/ticgit
  Push URL: https://github.com/schacon/ticgit
  HEAD branch: master
  Remote branches:
    master tracked
    ticgit tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (up to date)

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ |
```

Рисунок 71 - Информация об удаленном репозитории

Для переименования удаленных репозиториях используется команда `git remote rename` (рисунок 8).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ git remote rename pb paul

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ git remote
origin
paul

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$
```

Рисунок 72 - Переименование удаленного репозитория

Для удаления удаленного репозитория нужно выполнить команду `git remote remove` (рисунок 9).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ git remote remove paul

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ git remote
origin

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/ticgit (master)
$ |
```

Рисунок 73 - Удаление удаленного репозитория



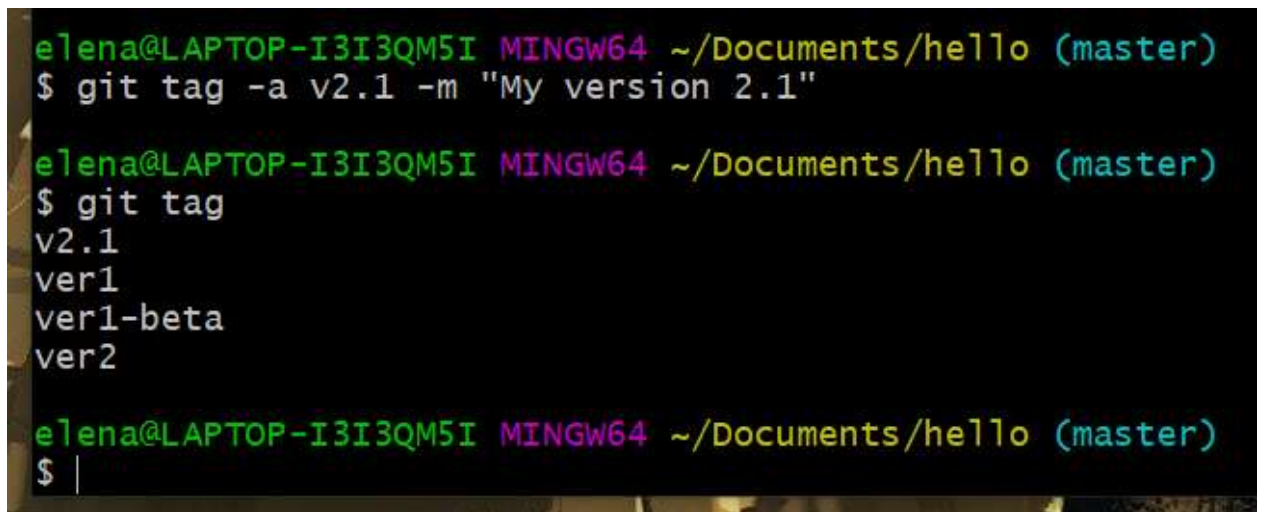
Просмотреть существующие теги можно с помощью команды git tag (рисунок 10).



```
elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git tag
ver1
ver1-beta
ver2
```

Рисунок 74 - Просмотр тегов

Для создания аннотированной метки нужно выполнить команду, показанную на рисунке 11.



```
elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git tag -a v2.1 -m "My version 2.1"

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git tag
v2.1
ver1
ver1-beta
ver2

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ |
```

Рисунок 75 - Создание аннотированной метки

Команда git show осуществляет просмотр данных тегов вместе с коммитом (рисунок 12).

```

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git show v2.1
tag v2.1
Tagger: ElenaVerstova <isip_e.v.verstova@mppt.ru>
Date:   Fri Mar 19 11:18:30 2021 +0300

My version 2.1

commit c231127b40601b59292a5e3b0de3854e5341524c (HEAD -> master, tag: v2.1, shared/master)
Author: ElenaVerstova <isip_e.v.verstova@mppt.ru>
Date:   Wed Mar 10 14:35:49 2021 +0300

    Added shared comment to README

diff --git a/README b/README
index 9091eb5..61fdad9 100644
--- a/README
+++ b/README
@@ -1,2 +1,2 @@
- This is the Hello World example from the Git tutorial
- (changed in original)
\ No newline at end of file
+ (Changed in the original and pushed to shared)
\ No newline at end of file
:...skipping...
tag v2.1
Tagger: ElenaVerstova <isip_e.v.verstova@mppt.ru>
Date:   Fri Mar 19 11:18:30 2021 +0300

```

Рисунок 76 - Просмотр данных тега

Для создания легковесной метки не нужно передавать опции -a, -s и -m, надо указать только название (рисунок 13). Просмотр данных такой метки осуществляется также с помощью git show (рисунок 14).

```

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git tag v2.1-lw

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git tag
v2.1
v2.1-lw
ver1
ver1-beta
ver2

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ |

```

Рисунок 77 - Создание легковесной метки

```

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git show v2.1-lw
commit c231127b40601b59292a5e3b0de3854e5341524c (HEAD -> master, tag: v2.1-lw, t
ag: v2.1, shared/master)
Author: ElenaVerstova <isip_e.v.verstova@mpt.ru>
Date:   Wed Mar 10 14:35:49 2021 +0300

    Added shared comment to README

diff --git a/README b/README
index 9091eb5..61fdad9 100644
--- a/README
+++ b/README
@@ -1,2 +1,2 @@
 This is the Hello World example from the Git tutorial
-(changed in original)
 \ No newline at end of file
+(Changed in the original and pushed to shared)
:...skipping...
commit c231127b40601b59292a5e3b0de3854e5341524c (HEAD -> master, tag: v2.1-lw, t
ag: v2.1, shared/master)
Author: ElenaVerstova <isip_e.v.verstova@mpt.ru>
Date:   Wed Mar 10 14:35:49 2021 +0300

    Added shared comment to README

diff --git a/README b/README
index 9091eb5..61fdad9 100644

```

Рисунок 78 - Просмотр данных тега

Для отметки определенного коммита тегом надо указать его хэш (рисунки 15-17).

```

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git log --pretty=oneline
c231127b40601b59292a5e3b0de3854e5341524c (HEAD -> master, tag: v2.1-lw, tag: v2.
1, shared/master) Added shared comment to README
5906087d2ffabe869dd34d6c5b2ec9ac096b6731 Changed README in original repo
9dadf96814ea7867a7271cb13882cc5b8bf062fe (style) Life is great!
3559bed39f209c82d41ba9a97363d292e18c5342 Added README
c2bcb764686a36610deccd4090f0e124c2fdb0e4 (tag: ver2) Added index.html
b5bd82753296f765e6aa0ebb5d60c3e0a93cac3c Moved hello.html to lib
aefcfd7f3a22e33c57a2925359205fa53a32b5b3 Add an author/email comment
d243b6b28cd79a79b0d27214ef82f2032d3de69c (tag: ver1) Added standard HTML tags
8de3e8034c7ba4cc85a9ae26844a5c51be5f5e66 (tag: ver1-beta) Added tag h2
ce52614162e1ea0e0ba63049b2c063a56f2d20a6 Unrelated change for v
e1f22769d657733ef4db0e7dd536a62a901a3395 Changes for a and b
f302a0ccb31ec3bd823f0bc11151e14317ccaef8 First Commit

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$

```

Рисунок 79 - Просмотр истории

```

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git tag -a ver3 5906087
hint: Waiting for your editor to close the file...
(electron) Sending uncompressed crash reports is deprecated and will be removed
in a future version of Electron. Set { compress: true } to opt-in to the new beh
avior. Crash reports will be uploaded gzipped, which most crash reporting server
s support.
[main 2021-03-19T08:22:54.139Z] update#setState idle
(node:28676) electron: The default of contextIsolation is deprecated and will be
changing from false to true in a future release of Electron. See https://githu
b.com/electron/electron/issues/23506 for more information
(node:28676) electron: The default of contextIsolation is deprecated and will be
changing from false to true in a future release of Electron. See https://githu
b.com/electron/electron/issues/23506 for more information

```

Рисунок 80 - Создание тега определенному коммиту

```

≡ TAG_EDITMSG ●
C: > Users > elena > Documents > hello > .git > ≡ TAG_EDITMSG
1 Tag message1
2 #
3 # Write a message for tag:
4 # ver3
5 # Lines starting with '#' will be ignored.
6

```

Рисунок 81 - Ввод сообщения в текстовом редакторе

Данные этого тега можно просмотреть аналогичным образом (рисунок 18).

```

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git show ver3
tag ver3
Tagger: ElenaVerstova <isip_e.v.verstova@mpt.ru>
Date:   Fri Mar 19 11:22:53 2021 +0300

Tag message1

commit 5906087d2ffabe869dd34d6c5b2ec9ac096b6731 (tag: ver3)
Author: ElenaVerstova <isip_e.v.verstova@mpt.ru>
Date:   Wed Mar 10 14:24:40 2021 +0300

    Changed README in original repo

diff --git a/README b/README
index 3f16465..9091eb5 100644
--- a/README
+++ b/README
@@ -1,2 @@
-This is the Hello World example from the Git tutorial
\ No newline at end of file
+This is the Hello World example from the Git tutorial
+(changed in original)
\ No newline at end of file

```

Рисунок 82 - Просмотр данных тега

По умолчанию, команда `git push` не отправляет теги на удалённые сервера. Нужно выполнить команду `git push shared` (рисунок 19).

```

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git push shared v2.1
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 170 bytes | 34.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To ../hello.git
 * [new tag]          v2.1 -> v2.1

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$

```

Рисунок 83 - Отправка тега на удаленный сервер

Можно использовать опцию `--tags` для команды `git push`. В таком случае все теги отправятся на удалённый сервер (рисунок 20).



```

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git push shared --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 169 bytes | 42.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To ../hello.git
 * [new tag]          v2.1-1w -> v2.1-1w
 * [new tag]          ver3 -> ver3

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ |

```

Рисунок 84 - Отправка всех тегов на сервер

Для того, чтобы удалить тег, надо использовать команду `git tag` с параметром `-d` (рисунок 21).

```

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git tag -d v2.1-1w
Deleted tag 'v2.1-1w' (was c231127)

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$

```

Рисунок 85 - Удаление тега

Для удаления тега с сервера используется команда, показанная на рисунке 22.

```

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git push shared :refs/tags/v2.1-1w
To ../hello.git
- [deleted]          v2.1-1w

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$

```

Рисунок 86 - Удаление тегов с сервера

Для того, чтобы получить версии файлов, на которые указывает тег, можно выполнить `git checkout` для тега. Однако, это переведёт репозиторий в состояние «detached HEAD». Если в состоянии «detached HEAD» внести изменения и сделать коммит, то тег не изменится, при этом новый коммит не будет относиться ни к какой из веток, а доступ к нему можно будет получить только по его хэшу. Поэтому в таком случае следует создать новую ветку (рисунки 23-24).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git checkout v2.1
Note: switching to 'v2.1'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false
HEAD is now at c231127 Added shared comment to README
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello ((v2.1))
$ |
```

Рисунок 87 - Переключение на метку

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello ((v2.1))
$ git checkout -b version2 v2.1
Switched to a new branch 'version2'

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ |
```

Рисунок 88 - Создание новой ветки

Можно создать псевдонимы (алиасы) для команд. Создание алиасов и примеры их использования показаны на рисунках 25-30.

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ git config --global alias.co checkout

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ git config --global alias.br branch

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ git config --global alias.ci commit

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ git config --global alias.st status
```

Рисунок 89 - Задание алиасов

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ git config --global alias.unstage 'reset HEAD --'

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ |
```

Рисунок 90 - Создание псевдонима исключения файла из индекса

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ git unstage fileA

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ git reset HEAD -- fileA

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ |
```

Рисунок 91 - Использование созданного псевдонима

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ git config --global alias.last 'log -1 HEAD'

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$
```

Рисунок 92 - Создание алиаса для просмотра последнего коммита

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ git last
commit c231127b40601b59292a5e3b0de3854e5341524c (HEAD -> version2, tag: v2.1, shared/master, master)
Author: ElenaVerstova <isip_e.v.verstova@mpt.ru>
Date:   Wed Mar 10 14:35:49 2021 +0300

    Added shared comment to README

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ |
```

Рисунок 93 - Результат работы созданного алиаса

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ git config --global alias.visual '!gitk'

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ |
```

Рисунок 94 - Создание псевдонима внешней команды



```
tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ git clone https://github.com/schacon/ticgit
Cloning into 'ticgit'...
remote: Enumerating objects: 167, done.
remote: Total 167 (delta 0, reused 0 (delta 0), pack-reused 167) (from 1)
Receiving objects: 100% (167/167), 24.06 KiB | 3.06 MiB/s, done.
Resolving deltas: 100% (97/97), done.
tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ cd ticgit
tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store/ticgit (master)
$ git remote
origin
tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)
tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store/ticgit (master)
$ git remote add pd https://github.com/paulbowne/ticgit
tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)
pd https://github.com/paulbowne/ticgit (fetch)
pd https://github.com/paulbowne/ticgit (push)
tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store/ticgit (master)
$ git fetch pd
fatal: 'pd' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store/ticgit (master)
$ git fetch --remote
fatal: 'remote' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store/ticgit (master)
$ git push origin master
info: please complete authentication in your browser...
remote: Permission to schacon/ticgit.git denied to TenurHuber.
fatal: unable to access 'https://github.com/schacon/ticgit/': The requested URL returned error: 403

tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store/ticgit (master)
$

tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store/ticgit (master)
$ git remote show origin
* remote origin
Fetch URL: https://github.com/schacon/ticgit
Push URL: https://github.com/schacon/ticgit
HEAD branch: master
Remote branches:
  master tracked
  ticgit tracked
Local branch configured for 'git pull':
  master merges with remote master
Local ref configured for 'git push':
  master pushes to master (up to date)

tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store/ticgit (master)
$ git remote rename pb paul
error: No such remote: 'pb'

tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store/ticgit (master)
$ git remote remove paul
error: No such remote: 'paul'

tenur@INBOOK_V1_PLUS MINGW64 /c/store management/Desktop/store/ticgit (master)
$ git remote origin
error: unknown subcommand: 'origin'
usage: git remote [-v] [--verbose]
       or: git remote add [-t <branch>] [-m <master>] [-f] [--tags] |
       --no-tags] [--mirror=<fetch|push>] <name> <url>
```

```
$ git remote origin
error: unknown subcommand: 'origin'
usage: git remote [-v | --verbose] [-m <master>] [-f] [--tags | --no-tags] [--mirror=<fetch|push>] <name> <url>
or: git remote rename [--no-progress] <old> <new>
or: git remote remove <name>
or: git remote set-head <name> (-a | --auto | -d | --delete | <branch>)
or: git remote [-v | --verbose] show [-n] <name>
or: git remote prune [-n | --dry-run] <name>
or: git remote [-v | --verbose] update [-p | --prune] [(<group> | <remote>)...]
or: git remote set-branches [--add] <name> <branch>...
or: git remote get-url [--push] [--all] <name>
or: git remote set-url [--push] <name> <newurl> [<oldurl>]
or: git remote set-url --add <name> <newurl>
or: git remote set-url --delete <name> <url>

-v, --[no-]verbose    be verbose; must be placed before a subcommand

Temyp@INBOOK_V1_PLUS HINGW64 /c/store managment/Desktop/store/ticgit
it (master)
$ git tag

Temyp@INBOOK_V1_PLUS HINGW64 /c/store managment/Desktop/store/ticgit
it (master)
$ git tag -a v2.1 -m "My version 2.1"

Temyp@INBOOK_V1_PLUS HINGW64 /c/store managment/Desktop/store/ticgit
it (master)
$ git tag
v2.1

Temyp@INBOOK_V1_PLUS HINGW64 /c/store managment/Desktop/store/ticgit
it (master)
$ git show v2.1

commit 847256809a3d518cd36b8f81859401416fe8d945 (HEAD -> master, tag: v2.1, origin/master, origin/HEAD)
Author: Jeff Welling <jeff.welling@gmail.com>
Date:   Tue Apr 26 17:29:17 2011 -0700

    Added note to clarify which is the canonical TicGit-ng repo

diff --git a/README.mkd b/README.mkd
index ab92035..9ea9ff9 100644
--- a/README.mkd
+++ b/README.mkd
@@ -1.1,1.6 @@
+Note: the original TicGit author has pulled all the TicGit-ng changes into his repository, creating a potentially confusing situation. The canonical TicGit repo, this one, is not consistently maintained. For up to date TicGit-ng info and code, check the canonical TicGit-ng repository at:
+https://github.com/jeffwelling/ticgit
+
+## TicGit-ng ##
+
+This project provides a ticketing system built on Git that is kept in a

Temyp@INBOOK_V1_PLUS HINGW64 /c/store managment/Desktop/store/ticgit
it (master)
$ git tag v2.1-1w

Temyp@INBOOK_V1_PLUS HINGW64 /c/store managment/Desktop/store/ticgit
it (master)
$ git tag
v2.1
v2.1-1w

Temyp@INBOOK_V1_PLUS HINGW64 /c/store managment/Desktop/store/ticgit
it (master)
$ git show v2.1-1w
commit 847256809a3d518cd36b8f81859401416fe8d945 (HEAD -> master, tag: v2.1-1w, tag: v2.1, origin/master, origin/HEAD)
Author: Jeff Welling <jeff.welling@gmail.com>
Date:   Tue Apr 26 17:29:17 2011 -0700

    Added note to clarify which is the canonical TicGit-ng repo

diff --git a/README.mkd b/README.mkd
index ab92035..9ea9ff9 100644
--- a/README.mkd
+++ b/README.mkd
@@ -1.1,1.6 @@
+Note: the original TicGit author has pulled all the TicGit-ng changes into his repository, creating a potentially confusing situation. The canonical TicGit repo, this one, is not consistently maintained. For up to date TicGit-ng info and code, check the canonical TicGit-ng repository at:
+https://github.com/jeffwelling/ticgit
+
+## TicGit-ng ##
+
+This project provides a ticketing system built on Git that is kept in a

Temyp@INBOOK_V1_PLUS HINGW64 /c/store managment/Desktop/store/ticgit
it (master)
```



```
MINGW64 ~/store management/Desktop/ticgit
tic (master)
$ git log --pretty=oneline
047256809a3d518cd3688f81859401416fa86945 (HEAD -> master, tag: v2.1)
047256809a3d518cd3688f81859401416fa86945 (HEAD -> master, tag: v2.1-lw, tag: v2.1, origin/master, origin/HEAD) Added note to clarify wh
ich is the canonical ticgit-ng repo
c095b7e68d8f7560dc820b1e20adff427429707 Merge branch 'hotfix-1.0.2.2'
e4932587f8ee041365849c8ac9fa56bf4a000a Bumped version to 1.0.2.2470d7afa8de9aa612cf711a7c0d702bdbc3aid Fixed bug where 'ti comment'
was prepopulated with a ruby object
02015390494e8ae2e10391c95df89ee44ce4ec1a Merge branch 'hotfix-1.0.2.1'
071eb64661414d963368056ddb3d66e77609f8a6 Bumped version to 1.0.2.137092ec5e38287b9856b3d52bb4c0a0ae2d776b0 Fixed bug where the comments
(#{s) appeared as first comment
2afa7ce172ecc2b44bbda855e6caca1d6c7f7c1 Merge branch 'release-1.0.2'
48f34b3b28a3df2278430cf8b87b45c971c67d95 Bumped version to 1.0.2
4f7043d484f9ff7852cc1c0c0af823961042fa8 Added *.swp to .gitignore, for files vim uses
31d95151f52c16c02aa7a65fa8a514317ddb8b85 Fix rdoc rake task
0178a88eb37d1013ec343f9ee470ac56e8926cc rspec 2 task fixed
f490146353247138a7251590762364a2cc82f1c9 rakefile
49f6112bb4652490d0505578d58ddf2306316101c Merge branch 'ruby19' into deving
4c31fec749a7247728825d8d0ab834ff91ad72d Only use monkeypatch if we aren't using 1.9
00954536b63574bbd36a7bd4581f0dddc126d948 Merge branch 'ruby19' into deving
280f1a60d42af38c364f67f783638178b84410b Revert "Changed cli_spec back, but filed a bug to sort ti help commands"
c67e166b1e4a4ce898b888e45f12f5e490020270 Merge branch 'ruby19' into deving
d94e91cfe0de7936c00746fac7f5cb085091425c Changed cli_spec back, but filed a bug to sort ti help commands
7f4450bb0249139201cbbaa40c2cb91f7f01ccb green specs and looks like it's ok on ruby 1.9.2
5b85fe901f21d3ad660c8003cc0451d3a582ecc1 Merge branch 'hotfix-1.0.1.3' into deving
d75ad827ce8c16ae4c296ed5e159f5d01100865e Merge branch 'hotfix-1.0.1'
047256809a3d518cd3688f81859401416fa86945 (HEAD -> master, tag: v2.1-lw, tag: v2.1, origin/master, origin/HEAD) Added note to clarify wh
ich is the canonical Ticgit-ng repo
```

```
MINGW64 ~/store management/Desktop/tic
$ git tag -a v2.1 047256809a3d518cd3688f81859401416fa86945
fatal: failed to resolve '047256809a3d518cd3688f81859401416fa86945' as a valid ref.

MINGW64 ~/store management/Desktop/tic
$ git tag -a v2.1-lw 047256809a3d518cd3688f81859401416fa86945
fatal: failed to resolve '047256809a3d518cd3688f81859401416fa86945' as a valid ref.

MINGW64 ~/store management/Desktop/tic
$ git push shared:refs
error: unable to delete 'refs/': remote ref does not exist
error: failed to push some refs to '://store.git'

MINGW64 ~/store management/Desktop/tic
$ git push shared:refs/tags/v2.1-lw
warning: deleting a non-existent ref
to //store.git
- [deleted] v2.1-lw

MINGW64 ~/store management/Desktop/tic
$ git checkout v2.1
error: pathspec 'v2.1' did not match any file(s) known to git

MINGW64 ~/store management/Desktop/tic
$ git checkout -b v2.1
fatal: 'v2.1' is not a commit and a branch 'v2.1' cannot be created from it

MINGW64 ~/store management/Desktop/tic
$ git config --global alias.br branch

MINGW64 ~/store management/Desktop/tic
$ git config --global alias.br status

MINGW64 ~/store management/Desktop/tic
$ git config --global alias.unstage 'reset HEAD --'

MINGW64 ~/store management/Desktop/tic
$ git status files
output: changes after reset:
T.txt
$ git checkout
$ git reset HEAD -- T.txt
output: changes after reset:
T.txt
$ git checkout
$ git reset HEAD -- T.txt
output: changes after reset:
T.txt
$ git checkout
```

```
MINGW64 ~/store management/Desktop/tic
$ git push shared:refs/tags/v2.1-lw
warning: deleting a non-existent ref
to //store.git
- [deleted] v2.1-lw

MINGW64 ~/store management/Desktop/tic
$ git checkout v2.1
error: pathspec 'v2.1' did not match any file(s) known to git

MINGW64 ~/store management/Desktop/tic
$ git checkout -b v2.1
fatal: 'v2.1' is not a commit and a branch 'v2.1' cannot be created from it

MINGW64 ~/store management/Desktop/tic
$ git config --global alias.br branch

MINGW64 ~/store management/Desktop/tic
$ git config --global alias.br status

MINGW64 ~/store management/Desktop/tic
$ git config --global alias.unstage 'reset HEAD --'

MINGW64 ~/store management/Desktop/tic
$ git status files
output: changes after reset:
T.txt
$ git checkout
$ git reset HEAD -- T.txt
output: changes after reset:
T.txt
$ git checkout
$ git reset HEAD -- T.txt
output: changes after reset:
T.txt
$ git checkout
$ git log
git: 'log' is not a git command. see 'git --help'.
The most similar command is
stage

MINGW64 ~/store management/Desktop/tic
$ git config --global alias.unstage 'git'

MINGW64 ~/store management/Desktop/tic
$
```

## Работа с Git

Команда `git branch` делает несколько больше, чем просто создаёт и удаляет ветки. При запуске без параметров, можно получить простой список имеющихся веток (рисунок 1). Символ `*`, стоящий перед веткой `master` указывает на ветку, на которую указывает HEAD).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git branch
* master
  newbranch
  testing

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ |
```

Рисунок 95 - Список существующих веток

Чтобы посмотреть последний коммит на каждой из веток, необходимо выполнить команду `git branch -v` (рисунок 2).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git branch -v
* master      c26e2f0 New CHanges
  newbranch   c26e2f0 New CHanges
  testing     0c5465f CHanges

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$
```

Рисунок 96 - Список веток с последними коммитами

Опции `--merged` и `--no-merged` могут отфильтровать этот список для вывода только тех веток, которые слиты или ещё не слиты в текущую ветку (рисунки 3-4).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git branch --merged
* master
  newbranch
  testing
```

Рисунок 97 - Список веток слитых с текущей

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git branch --no-merged
```

Рисунок 98 - Список веток не слитых с текущей

Затем следует удалить ветку `testing` (рисунок 5). При наличии ошибок для удаления можно использовать параметр `-D`.

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git branch --d testing
Deleted branch testing (was 0c5465f).

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$
```

Рисунок 99 - Удаление ветки

Для получения списка удалённых веток и дополнительной информации используется команда `git remote show` (рисунок 6).

pus

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two/repo1 (main)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/Eternity-blip/repo1
  Push URL: https://github.com/Eternity-blip/repo1
  HEAD branch: main
  Remote branch:
    main tracked
  Local branches configured for 'git pull':
    main merges with remote main
    serv merges with remote main
  Local ref configured for 'git push':
    main pushes to main (local out of date)
```

Рисунок 100 - Просмотр удаленных веток

Для отправления изменений на удалённый сервер используется команда `git push <remote> <branch>` (рисунок 7).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two/repo1 (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 285 bytes | 95.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Eternity-blip/repo1
  6cdd680..f6b94bc main -> main
```

Рисунок 101 - Отправка изменений

Далее при получении обновлений с сервера будет показана ссылка на удаленную ветку (рисунок 8).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two/repo1 (main)
$ git fetch origin
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 693 bytes | 18.00 KiB/s, done.
From https://github.com/Eternity-blip/repo1
   f6b94bc..68cbf5f  main       -> origin/main
```

Рисунок 102 - Выполнение команды git fetch

При необходимости можно создать локальную ветку на основе удаленной (рисунок 9).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two/repo1 (main)
$ git checkout -b serv origin/main
Switched to a new branch 'serv'
Branch 'serv' set up to track remote branch 'main' from 'origin'.
```

Рисунок 103 - Создание ветки на основе удаленной ветки

Для удаления веток на удаленном сервере используется команда, показанная на рисунке 10.

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two/repo1 (serv)
$ git push origin :serv
To https://github.com/Eternity-blip/repo1
   - [deleted]          serv
```

Рисунок 104 - Удаление ветки на сервере

Простой способ выполнить слияние двух веток – это команда merge. Другой способ – использование команды rebase, что означает перебазирование (рисунок 11). Это работает следующим образом: берётся общий родительский снимок двух веток (текущей, и той, поверх которой вы выполняете перебазирование), определяется дельта каждого коммита текущей ветки и сохраняется во временный файл, текущая ветка устанавливается на последний коммит ветки, поверх которой выполняется перебазирование, а затем по очереди применяются дельты из временных файлов.

Далее после этого надо переключиться на ветку master и выполнить перемотку.

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git checkout exp
Switched to branch 'exp'

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (exp)
$ git rebase master
Successfully rebased and updated refs/heads/exp.
```

Рисунок 105 - Перемещение изменений

При наличии ответвления от ветки (сначала было ответвление на ветку se, а затем от нее на ветку cl), чтобы переместить изменения можно осуществить действия, показанные на рисунках 12-15.

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git rebase --onto master se cl
Successfully rebased and updated refs/heads/cl.
```

Рисунок 106 - Перемещение изменений с параметром onto

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (cl)
$ git checkout master
Switched to branch 'master'

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git merge cl
Updating fafe213..08d256f
Fast-forward
 w.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 w.txt
```

Рисунок 107 - Слияние веток master и cl

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git rebase master se
Successfully rebased and updated refs/heads/se.
```

Рисунок 108 - Перемещение изменений

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (se)
$ git checkout master
Switched to branch 'master'

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git merge se
Updating 08d256f..4a98db2
Fast-forward
 e.txt | 0
 q.txt | 0
 repo1 | 1 +
 3 files changed, 1 insertion(+)
 create mode 100644 e.txt
 create mode 100644 q.txt
 create mode 160000 repo1
```

Рисунок 109 - Слияние веток master и se

После этого перемещение будет осуществлено и ветки можно удалить (рисунок 16).



```

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git branch -d cl
Deleted branch cl (was 08d256f).

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git branch -d se
Deleted branch se (was 4a98db2).

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$

```

Рисунок 110 - Удаление веток cl и se

```

MINGW64/c:/store management/Desktop/store
TevyePINBOOK_YI_PLUS MINGW64 /c:/store management/Desktop/store (master)
$ git branch
* master
  shared
  style
  test01
  testing

TevyePINBOOK_YI_PLUS MINGW64 /c:/store management/Desktop/store (master)
$ git branch -v
* master      2b7642c Merge branch 'test01'
  shared      c3d9c96 [behind 5] Changed store.html in original repo
  style        afea5bc комментарии
  test01       b5ea40a New changes
  testing      c3d9c96 Changed store.html in original repo

TevyePINBOOK_YI_PLUS MINGW64 /c:/store management/Desktop/store (master)
$ git branch --merged
* master
  shared
  testing

TevyePINBOOK_YI_PLUS MINGW64 /c:/store management/Desktop/store (master)
$ git branch --no-merged
  style
  test01

TevyePINBOOK_YI_PLUS MINGW64 /c:/store management/Desktop/store (master)
$ git branch -d testing
Deleted branch testing (was c3d9c96).

TevyePINBOOK_YI_PLUS MINGW64 /c:/store management/Desktop/store (master)
$ git remote show origin
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

TevyePINBOOK_YI_PLUS MINGW64 /c:/store management/Desktop/store (master)

```

```
MINGW64/c/store management/Desktop/store2
Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ git remote show shared
* remote shared
Fetch URL: ../store.git
Push URL: ../store.git
HEAD branch: master
Remote branches:
  Temur312 tracked
  master tracked
  style tracked
Local refs configured for 'git push':
  Temur312 pushes to Temur312 (up to date)
  master pushes to master (fast-forwardable)
  style pushes to style (fast-forwardable)

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ git push shared
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream shared master

To have this happen automatically for branches without a tracking
upstream, see 'push.autosetupremote' in 'git help config'.

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ git push master
fatal: 'master' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ git index --add
bash: git: command not found

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ git fetch origin
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.
```

```
MINGW64/c/store management/Desktop/store2
Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ git checkout master
D      README/index.html
D      README/index.html.txt
D      README/index.html.txt.orig
D      README/index2.html
D      README/test.rb.txt
D      ticgit
Already on 'master'

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ git merge se
merge: se - not something we can merge
g

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ git branch -d cl
error: branch 'cl' not found

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store (master)
$ cd ../

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop
$ cd store2

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store2 (master)
$ git push store2
fatal: 'store2' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store2 (master)
$ git branch
* master

Temyp@INBOOK_Y1_PLUS MINGW64 /c/store management/Desktop/store2 (master)
$ git push master
fatal: 'master' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

```

C:\WINDOWS\system32\cmd.exe
C:\store management\Desktop\store2 (master)
$ git branch
* master

C:\store management\Desktop\store2 (master)
$ git add .

C:\store management\Desktop\store2 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   f.txt
        modified:   notepad
        modified:   store.htm

C:\store management\Desktop\store2 (master)
$ git commit -m "testur"
[master c031df0] testur
 3 files changed, 4 insertions(+), 1 deletion(-)

C:\store management\Desktop\store2 (master)
$ git push master
fatal: 'master' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

C:\store management\Desktop\store2 (master)
$ git push
Counting objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (5/5), 380 bytes | 380.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)

```

```

C:\store management\Desktop\store2 (master)
$ git push master
fatal: 'master' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

C:\store management\Desktop\store2 (master)
$ git add .

C:\store management\Desktop\store2 (master)
$ git commit -m "testur"
[master ad842fa] testur
 3 files changed, 4 insertions(+), 1 deletion(-)

C:\store management\Desktop\store2 (master)
$ git push master
fatal: 'master' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

C:\store management\Desktop\store2 (master)
$

```