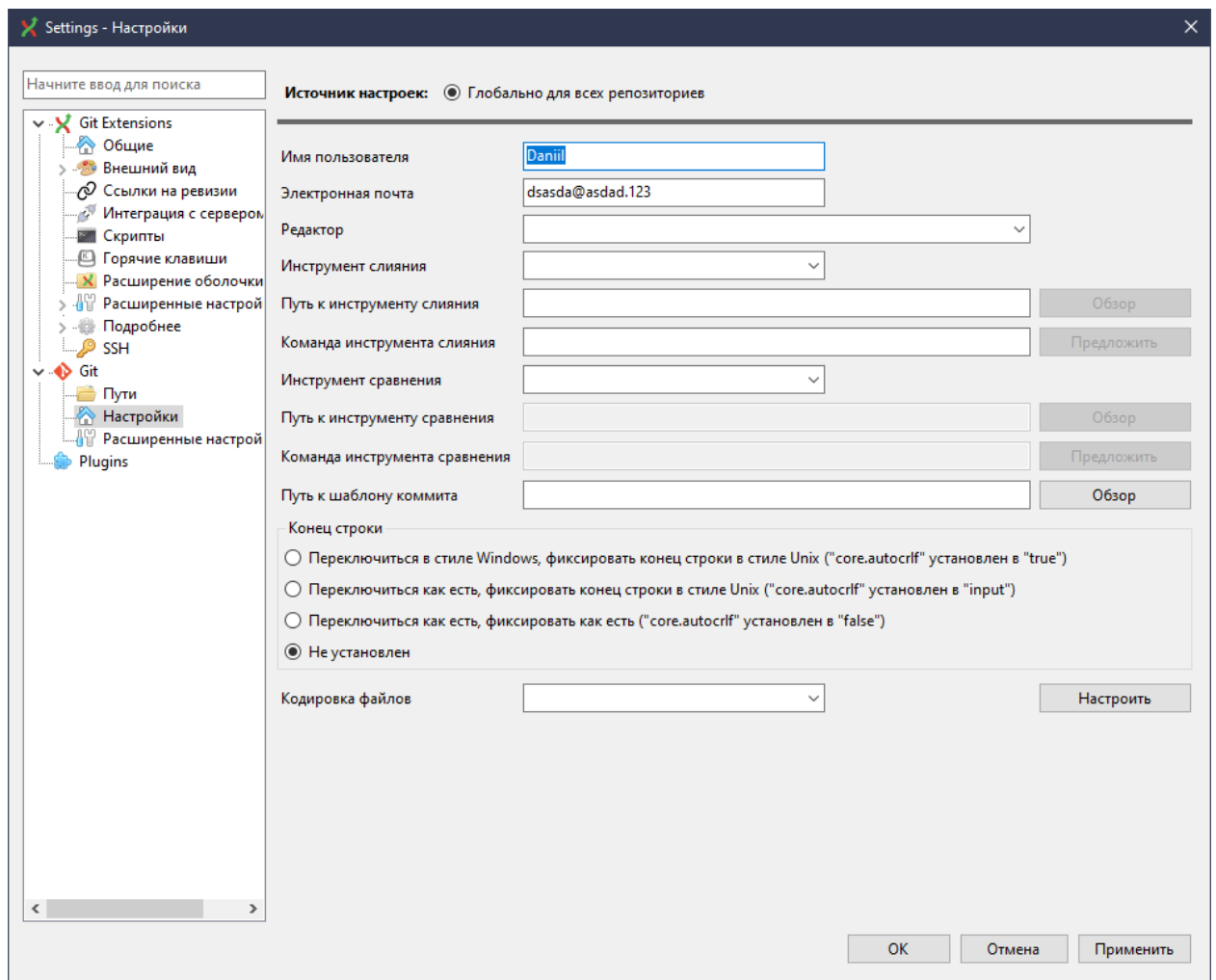


Если git никогда ранее не использовался пользователем, то необходимо установить имя и электронную почту. Для этого требуется выполнить команды, показанные на рисунке 1.



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

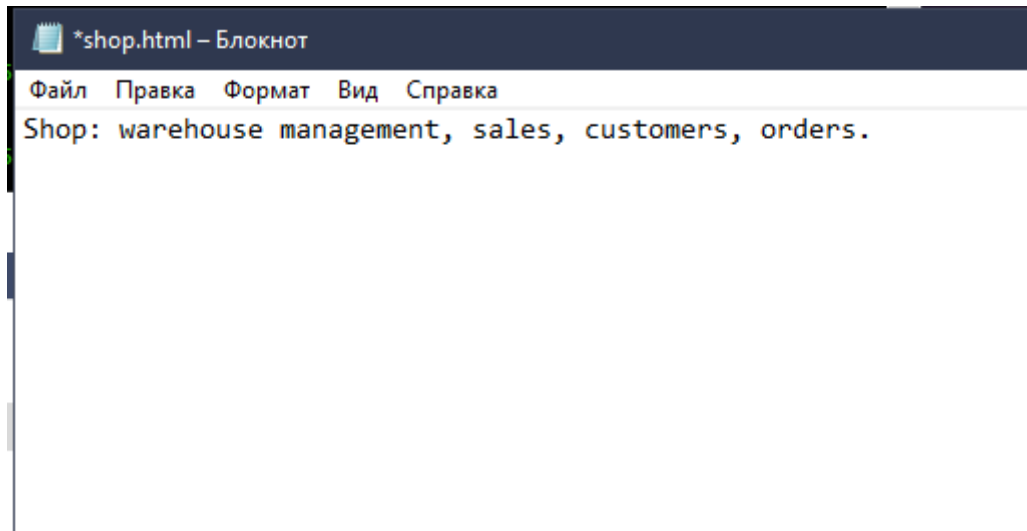
Толик@DESKTOP-J475S8M MINGW64 ~/shop
$ cd shop

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.



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.

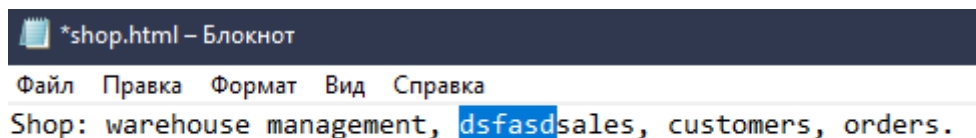
```

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-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

HELEN V@LAPTOP-BPH65Q86 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

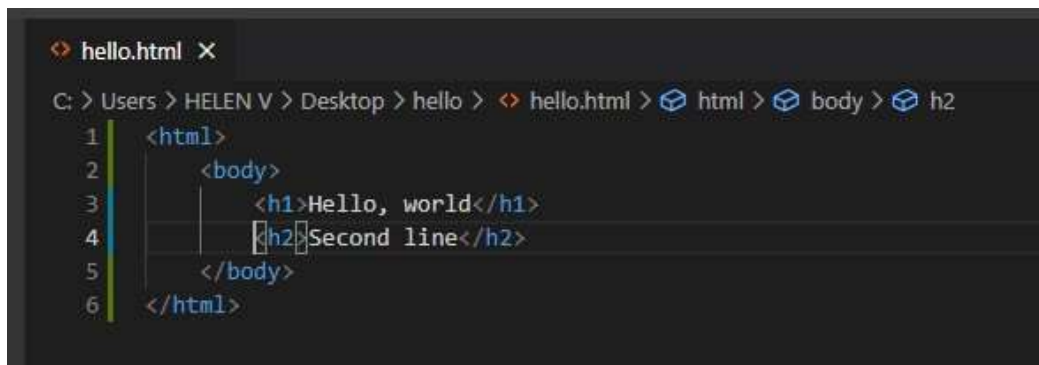
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ |
```

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.

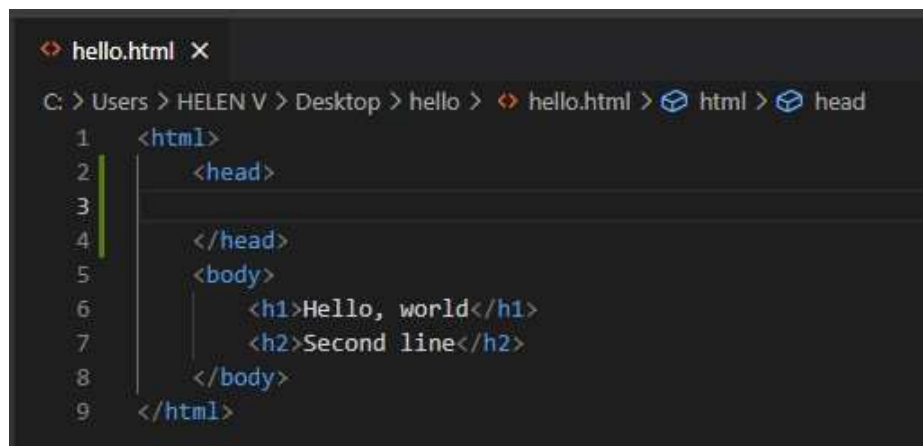
Сначала нужно в файл hello.html добавить теги html и body (рисунок 19), а затем проиндексировать изменения.



```
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

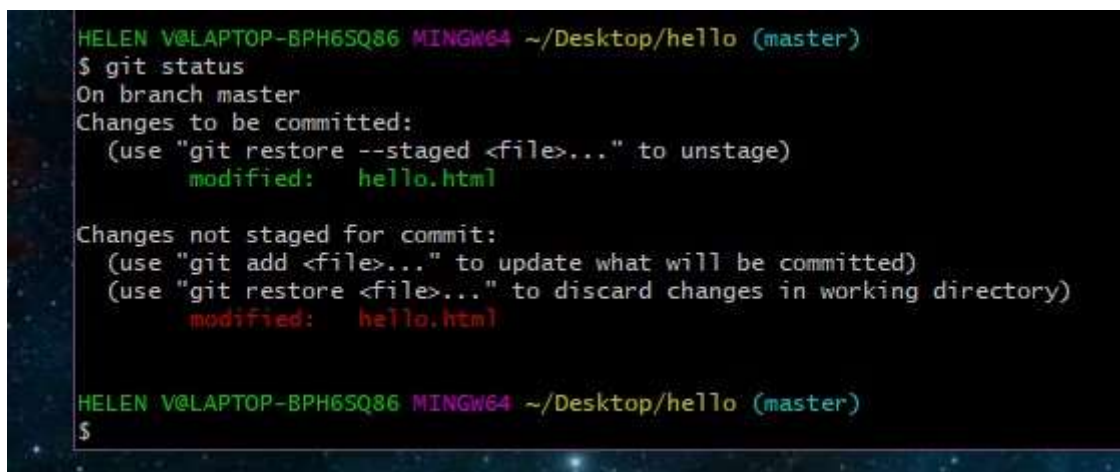
Затем еще раз нужно добавить изменения в файл (добавить тег head), но изменения не индексировать (рисунок 20).



```
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-BPH6SQ86 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-BPH6SQ86 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).

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

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

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

Рисунок 11 - Индексация 2 изменения и проверка состояния

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

```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git commit -m "Add HTML header"
[master d653372] Add HTML header
1 file changed, 3 insertions(+)

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

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 working 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
hiiii(/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
hiiii(/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).

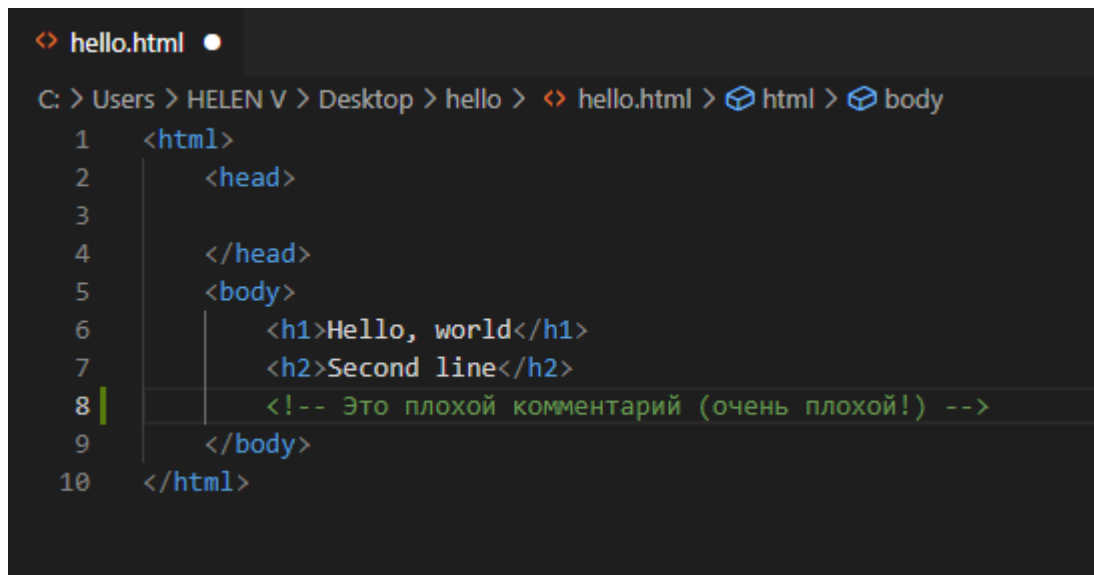
```

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

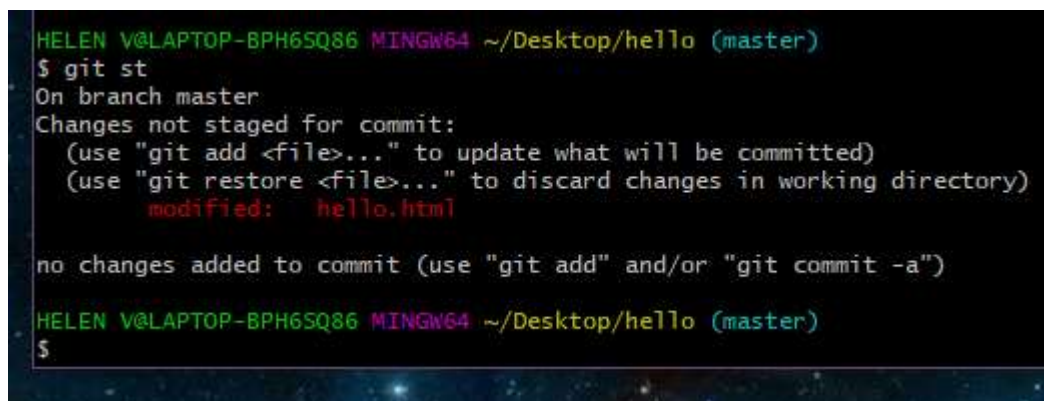
Далее для работы нужно внести изменение в файл (рисунок 22).



```
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-BPH65Q86 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-BPH65Q86 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>

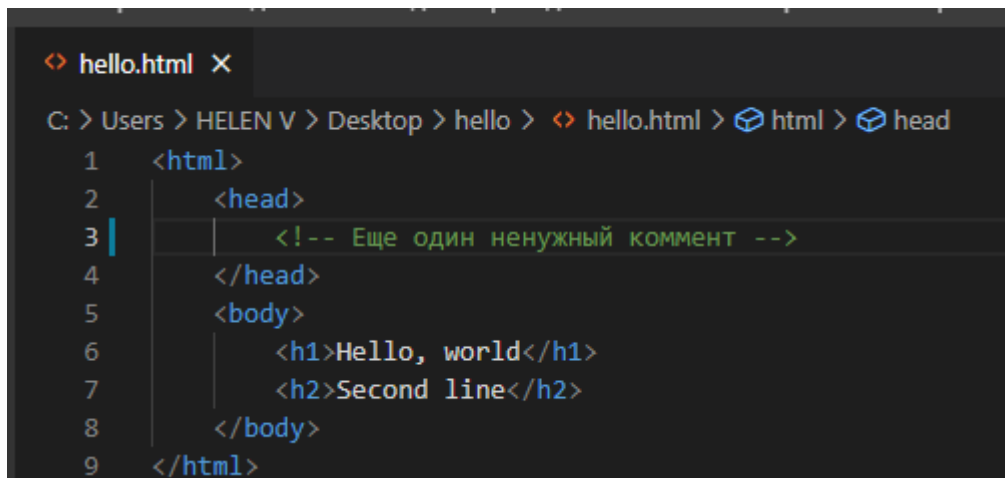
  </head>
  <body>
    <h1>Hello, world</h1>
    <h2>Second line</h2>
  </body>
</html>
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)

```

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
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-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$ git reset HEAD hello.html
Unstaged changes after reset:
M    hello.html

HELEN V@LAPTOP-BPH65Q86 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-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$ git checkout hello.html
Updated 1 path from the index

HELEN V@LAPTOP-BPH65Q86 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-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

HELEN V@LAPTOP-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$ git commit -m "Oops, this is bad commit"
[master ce12d52] Oops, this is bad commit
1 file changed, 1 insertion(+)

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

```

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-BPH65Q86 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/electron/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/electron/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

```
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
[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-BPH6SQ86 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-BPH6SQ86 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-BPH6SQ86 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

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

```
<> shop.html
1 | <!--Автор: Daniil-->
2 | hiii(/dsfs)
3 | fsdsdfdfs
4 | !_@(#cx.vg,m)
```

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

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git commit -m "add an author comment"
[master 52dfce4] add an author comment
1 file changed, 4 insertions(+), 1 deletion(-)

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

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

```
<> shop.html
1 | <!--Автор: Daniil (danik.arsen@yandex.ru)-->
2 | hiii(/dsfs)
3 | fsdsdfdfs
4 | !_@(#cx.vg,m)
```

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

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

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git commit --amend -m "add an author/email comment"
[master b694347] add an author/email comment
Date: Tue Jun 3 13:30:21 2025 +0300
1 file changed, 4 insertions(+), 1 deletion(-)

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

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

```
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)
$ |
```

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).

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

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

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

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

Далее надо осуществить коммит данного перемещения.

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git commit -m "moved shop.html to lib"
[master 58d569e] moved shop.html to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename shop.html => lib/shop.html (100%)

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

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

```
<> index.html > html
1  <html>
2    <body>
3      <iframe src="lib/hello.html" width="200" height="200"></iframe>
4    </body>
5  </html>
```

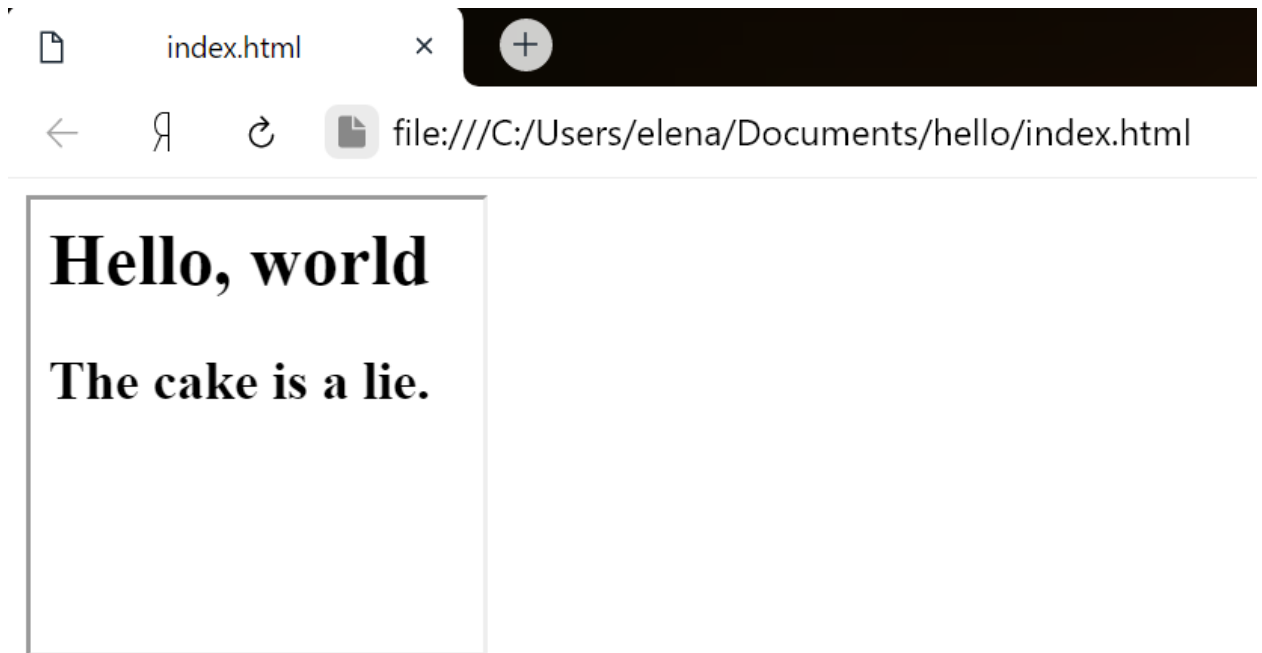
Далее нужно проиндексировать и закоммитить файл (рисунок 9).

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

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git commit -m "added index.html"
[master f185699] added index.html
1 file changed, 5 insertions(+)
create mode 100644 index.html

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


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



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

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ ls -c .git
objects/      index  logs/  config  hooks/
COMMIT_EDITMSG HEAD    refs/  info/   description

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

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

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ ls -c .git/objects
f1/  f9/  66/  b6/  98/  94/  8d/  e6/  24/  e0/  pack/
ba/  58/  80/  52/  6e/  67/  9c/  07/  af/  info/

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

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

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ ls -c .git/objects/8d
00a6d414574db74fad7e0b50cb7409c7bdbb25
```

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

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true

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

```

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

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ ls .git/refs
heads/  tags/

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ ls .git/refs/heads
master

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ ls .git/refs/tags
ver1  ver1-beta

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ ls .git/refs/tags/ver1
.git/refs/tags/ver1

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

```

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

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ cat .git/HEAD
ref: refs/heads/master

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

```

Для начала необходимо посмотреть последний коммит (рисунок 17).

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git hist --max-count=1
git: 'hist' is not a git command. See 'git --help'.

The most similar command is
    bisect

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git hist --max-count=1
git: 'hist' is not a git command. See 'git --help'.

The most similar command is
    bisect

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

```

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.

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (master)
$ git cat-file -t c2bcb76
fatal: Not a valid object name c2bcb76

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

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

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 d8a800db068656c3aaafc962f7a438c987b49dd60
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 31 - Просмотр содержимого файла при первом коммите

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)
```



```

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).

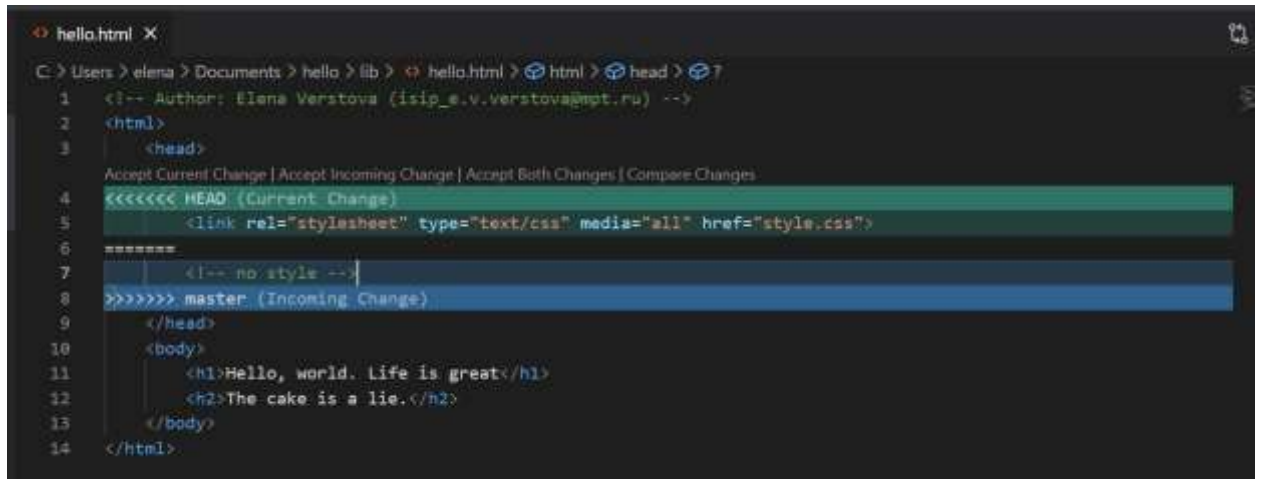


Рисунок 32 - Просмотр файла hello.html при наличии конфликта

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

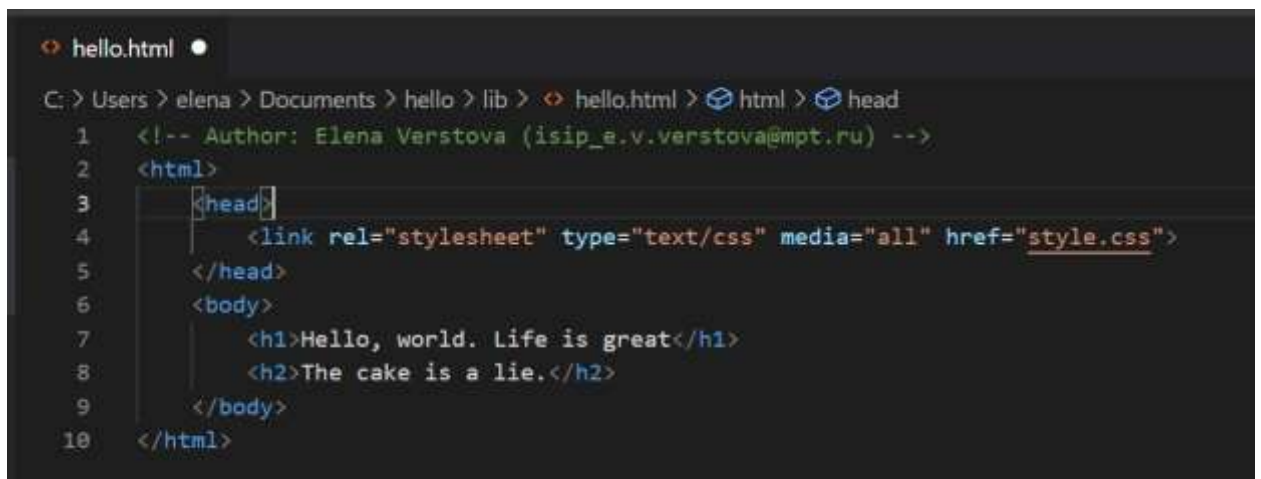


Рисунок 33 - Решение конфликта вручную

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 34 - Индексация и коммит

Для сброса ветки необходимо применить команду `reset --hard` до требуемой точки (рисунки 1-2).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style)
$ git checkout style
Already on 'style'

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style)
$ git hist
* 945ce49 2021-02-24 | Merged master fixed conflict. (HEAD -> style) [ElenaVerstova]
* 9dadf96 2021-02-24 | Life is great! (master) [ElenaVerstova]
* 6efc8fb 2021-02-24 | Merge branch 'master' into 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]

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

Рисунок 35 - Просмотр истории

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style)
$ git reset --hard c2bcb76
HEAD is now at c2bcb76 Added index.html

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style)
$ git hist --all
* 9dadf96 2021-02-24 | Life is great! (master) [ElenaVerstova]
* 3559bed 2021-02-24 | Added README [ElenaVerstova]
* c2bcb76 2021-02-20 | Added index.html (HEAD -> style, 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 36 - Сброс ветки style

Сброс ветки master

Аналогичные действия нужно произвести и для ветки master (рисунки 3-4).

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

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git hist
* 9dadf96 2021-02-24 | Life is great! (HEAD -> master) [ElenaVerstova]
* 3559bed 2021-02-24 | Added README [ElenaVerstova]
* c2bcb76 2021-02-20 | Added index.html (tag: ver2, style) [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 37 - Переключение на master и просмотр истории

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git reset --hard 9dadf96
HEAD is now at 9dadf96 Life is great!

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git hist --all
* 9dadf96 2021-02-24 | Life is great! (HEAD -> master) [ElenaVerstova]
* 3559bed 2021-02-24 | Added README [ElenaVerstova]
* c2bcb76 2021-02-20 | Added index.html (tag: ver2, style) [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]

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

Рисунок 38 - Сброс ветки master

Перебазирование

Команду rebase можно использовать вместо команды merge (рисунок 5).


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

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

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style)
$ git hist
* 9dadf96 2021-02-24 | Life is great! (HEAD -> style, master) [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 39 - Перебазирование веток

Слияние в ветку master

Далее требуется произвести слияние веток с помощью merge (рисунки 6-7).

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

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git merge style
Already up to date.

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

Рисунок 40 - Слияние веток

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git hist
* 9dadf96 2021-02-24 | Life is great! (HEAD -> master, 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]

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

Рисунок 41 - Просмотр истории

Клонирование репозитория

Далее требуется научиться делать копии репозитория. Для этого необходимо перейти в рабочий каталог и затем использовать команду `git clone`. Все данные действия показаны на рисунке 8.

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

Толик@DESKTOP-J475S8M MINGW64 ~
$ pwd
/c/Users/Толик

Толик@DESKTOP-J475S8M MINGW64 ~
$ cd desktop

Толик@DESKTOP-J475S8M MINGW64 ~/desktop
$ git clone shop1 cloned_chop1
Cloning into 'cloned_chop1'...
done.

Толик@DESKTOP-J475S8M MINGW64 ~/desktop
$ ls
1.1git.docx
```

Просмотр клонированного репозитория

После этого можно просмотреть клонированный репозиторий (рисунки 9-10).

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop
$ cd cloned_chop1

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/cloned_chop1 (style)
$ ls
a.html b.html c.html index.html lib/ read.txt

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

```
elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/cloned_hello (master)
$ git hist --all
* 9dadf96 2021-02-24 | Life is great! (HEAD -> master, origin/style, origin/master, origin/HEAD) [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]

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

Origin

Origin – имя по умолчанию. Просмотр данных о нем возможен с помощью команд, показанных на рисунке 11.

```

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

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/cloned_chop1 (style)
$ git remote show origin
* remote origin
  Fetch URL: C:/Users/Толик/Desktop/shop1
  Push URL: C:/Users/Толик/Desktop/shop1
  HEAD branch: style
  Remote branches:
    master tracked
    style tracked
  Local branch configured for 'git pull':
    style merges with remote style
  Local ref configured for 'git push':
    style pushes to style (up to date)

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

```

Для просмотра удаленных веток используется команда `git branch -a` (рисунок 12).

```

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)
$ |

```

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 X
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)
$
```

```
elena@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 42 - Просмотр истории

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

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

Рисунок 43 - 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 44 - 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).

```

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

```

Рисунок 45 - Настройка сервера

```

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

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

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

```

Рисунок 46 - Клонирование проекта

Работа с Git

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

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop1 (style)
$ git clone https://github.com/schacon/ticgit
Cloning into 'ticgit'...
remote: Enumerating objects: 1857, done.
remote: Total 1857 (delta 0), reused 0 (delta 0), pack-reused 1857 (from 1)
Receiving objects: 100% (1857/1857), 334.06 KiB | 2.63 MiB/s, done.
Resolving deltas: 100% (837/837), done.

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

Толик@DESKTOP-J475S8M MINGW64 ~/desktop
$ cd ticgit

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/ticgit (master)
$ git remote
origin

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

```

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

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)

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

```

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

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/ticgit (master)
$ git remote add pb https://github.com/paulboone/ticgit
bash: $'\302\223\302\223git': command not found

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)

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

```

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

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/ticgit (master)
$ git fetch pb
fatal: 'pb' 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.

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

```

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

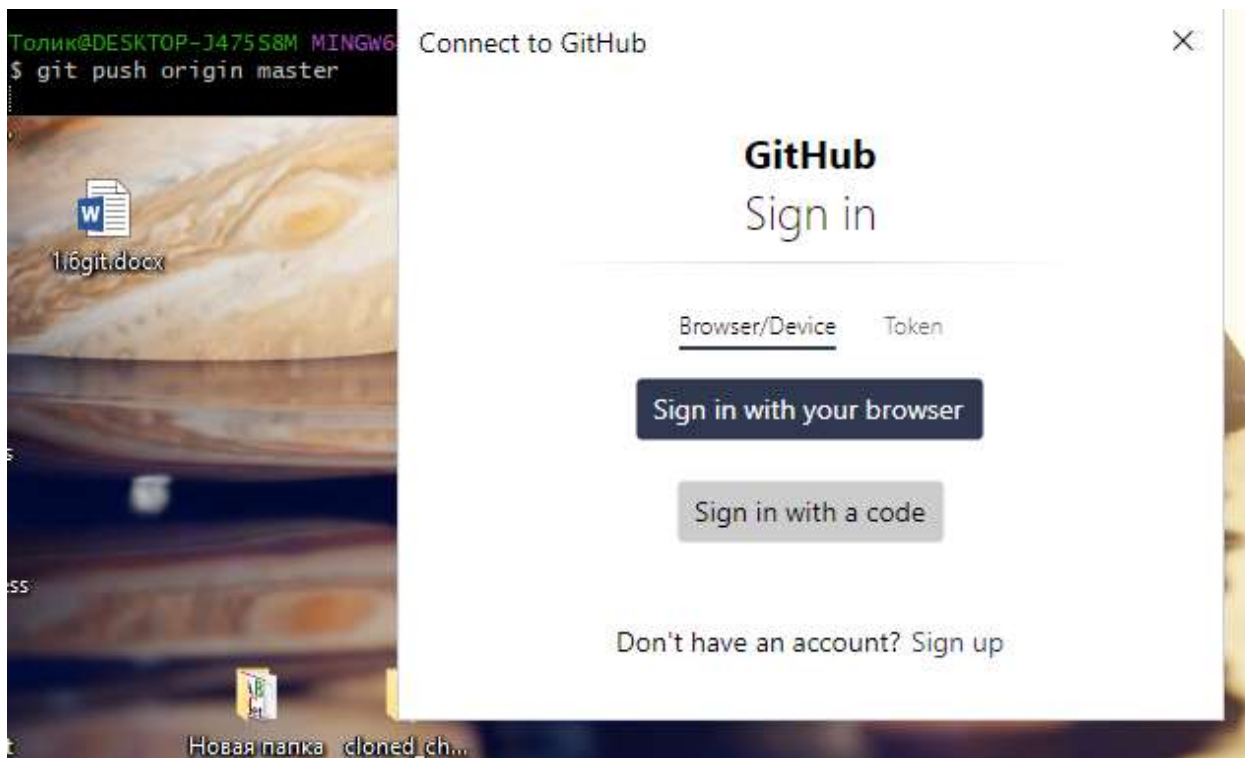

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/ticgit (master)
$ git fetch ~/desktop/cloned_chop1
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 29 (delta 8), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (29/29), 2.44 KiB | 45.00 KiB/s, done.
From C:/Users/Толик/desktop/cloned_chop1
 * branch                HEAD                -> FETCH_HEAD

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

```

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



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

```

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/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)

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

```

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

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/ticgit (master)
$ git remote rename pb paul
error: No such remote: 'pb'

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/ticgit (master)
$ git remote
origin

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

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

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/ticgit (master)
$ git remote remove paul
error: No such remote: 'paul'

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/ticgit (master)
$ git remote
origin

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

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

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/ticgit (master)
$ git tag
```

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

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/ticgit (master)
$ git tag -a v2.1 -m "my version 2.1"

Толик@DESKTOP-J475S8M MINGW64 ~/desktop/ticgit (master)
$ git tag
v2.1

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

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

```

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
@@ -1,3 +1,6 @@
-Note: the original TicGit author has pulled all the TicGit-ng changes into his
repository, creating a potentially confusing situation. The schacon TicGit repo,
this one, is not consistently maintained. For up to date TicGit-ng info and cod
e, check the canonical TicGit-ng repository at
https://github.com/jeffwelling/ticgit
...skipping...
tag v2.1
Tagger: Daniil <danik.aesen@yandex.ru>
Date:   Thu Jun 5 12:56:06 2025 +0300

my version 2.1

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
@@ -1,3 +1,6 @@
-Note: the original TicGit author has pulled all the TicGit-ng changes into his repository, creating a potent
ally confusing situation. The schacon TicGit repo, this one, is not consistently maintained. For up to date T
icGit-ng info and code, check the canonical TicGit-ng repository at
https://github.com/jeffwelling/ticgit
-
-## TicGit-ng ##

```

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).

```

Толик@DESKTOP-J475S8M MINGW64 ~
$ git tag v2.1-lw
fatal: not a git repository (or any of the parent directories): .git

Толик@DESKTOP-J475S8M MINGW64 ~
$ git tag
fatal: not a git repository (or any of the parent directories): .git

Толик@DESKTOP-J475S8M MINGW64 ~
$

```

```

Толик@DESKTOP-J475S8M MINGW64 ~
$ git show v2.1-lw
fatal: not a git repository (or any of the parent directories): .git

Толик@DESKTOP-J475S8M MINGW64 ~
$ |

```

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

```
eIena@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

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

Рисунок 47 - Просмотр истории

```
eIena@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 48 - 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 49 - 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 50 - 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 51 - 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 52 - 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 53 - 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 54 - 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).


```

elena@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
elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello ((v2.1))
$ |

```

Рисунок 55 - Переключение на метку

```

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

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

```

Рисунок 56 - Создание новой ветки

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

```

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

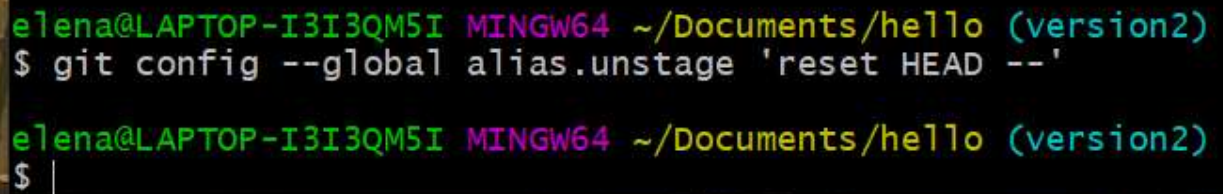
elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ git config --global alias.br branch

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

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

```

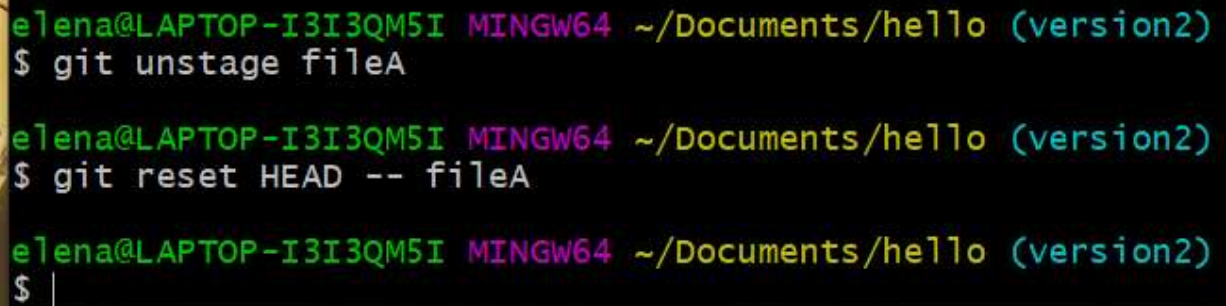
Рисунок 57 - Задание алиасов



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

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

Рисунок 58 - Создание псевдонима исключения файла из индекса

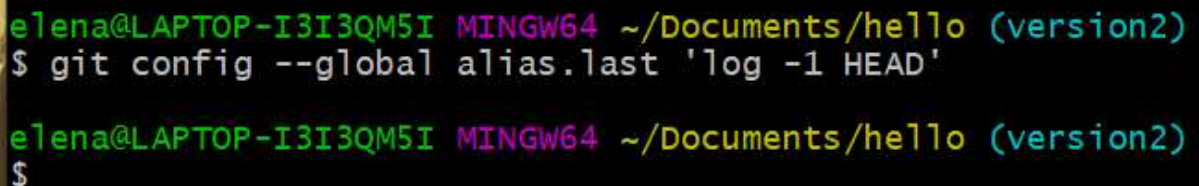


```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ git unstage fileA

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

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

Рисунок 59 - 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 60 - Создание алиаса для просмотра последнего коммита

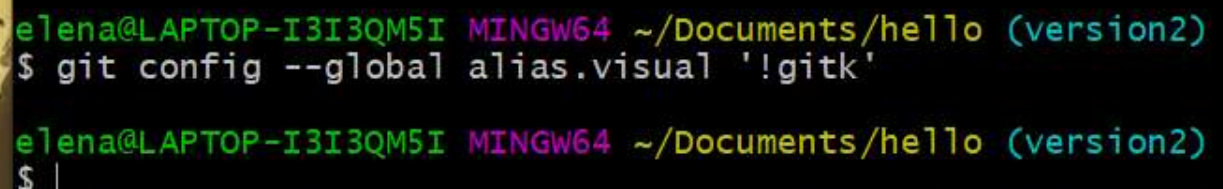


```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (version2)
$ git last
commit c231127b40601b59292a5e3b0de3854e5341524c (HEAD -> version2, tag: v2.1, shared/master, master)
Author: ElenaVerstova <isip_e.v.verstova@mpit.ru>
Date: Wed Mar 10 14:35:49 2021 +0300

    Added shared comment to README

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

Рисунок 61 - 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 62 - Создание псевдонима внешней команды

Работа с Git

Для начала следует создать репозиторий, создать 3 файла и добавить их в коммит (рисунок 1).

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop_two
$ git add read.txt test.rb license
fatal: not a git repository (or any of the parent directories): .git
```

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git commit -m 'Initial commit of my project'
[master (root-commit) d649d41] Initial commit of my project
3 files changed, 5 insertions(+)
create mode 100644 LICENSE
create mode 100644 README
create mode 100644 test.rb
```

Рисунок 1 - Индексация и коммит 3 файлов

Затем надо создать ветку testing и переключиться на нее (рисунки 2-3).

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop_two
$ git branch testing
fatal: not a git repository (or any of the parent directories): .git
```

Рисунок 2 - Создание ветки testing

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

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (testing)
$ |
```

Рисунок 3 - Переключение на ветку testing

Далее надо внести изменения в файл test.rb и создать коммит (рисунок 4).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (testing)
$ git add test.rb

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (testing)
$ git commit -m 'New changes'
[testing 86445a8] New changes
1 file changed, 1 insertion(+), 1 deletion(-)

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (testing)
$ |
```


Рисунок 4 - Индексация и коммит файла test.rb

Затем необходимо переключиться на ветку master и внести изменения в файл test.rb на этой ветке (рисунки 5-6).

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

Рисунок 5 - Переключение на ветку master

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git add test.rb

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git commit -a -m 'Another changes'
[master 078083e] Another changes
1 file changed, 2 insertions(+), 1 deletion(-)

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

Рисунок 6 - Еще индексация и коммит test.rb

Команда git checkout -b позволяет сразу создать и переключиться на ветку (рисунок 7).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git checkout -b iss53
Switched to a new branch 'iss53'

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (iss53)
$ |
```

Рисунок 7 - Создание и переключение на ветку iss53

В новой ветке нужно внести в файл изменения (рисунок 8).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (iss53)
$ git add index.html

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (iss53)
$ git commit -m 'Added html'
[iss53 8b79a37] Added html
1 file changed, 8 insertions(+)
create mode 100644 index.html

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (iss53)
$
```

Рисунок 8 - Индексация и коммит файла index.html

Далее нужно переключить ветку на master (рисунок 9).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (iss53)
$ git checkout master
Switched to branch 'master'

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

Рисунок 9 - Переключение на ветку master

Затем надо на ветке hotfix добавить изменения в файл index.html, а затем слить эту ветку и master (рисунки 10-11).

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

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (hotfix)
$ git commit -a -m 'Fixed the broken email address'
[hotfix a935cd7] Fixed the broken email address
1 file changed, 8 insertions(+)
create mode 100644 index.html

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (hotfix)
$ |
```

Рисунок 10 - Индексация и коммит index.html на ветке hotfix

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

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git merge hotfix
Updating 078083e..a935cd7
Fast-forward
 index.html | 8 ++++++++
1 file changed, 8 insertions(+)
create mode 100644 index.html

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

Рисунок 11 - Переключение на ветку master и объединение с веткой hotfix

После слияния ветку hotfix можно удалить (рисунок 12).¹

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git branch -d hotfix
Deleted branch hotfix (was a935cd7).

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

Рисунок 12 - Удаление ветки hotfix

Затем требуется внести изменения в iss53, переключиться на master и слить эти ветки (рисунки 13-14).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (iss53)
$ git add index.html

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (iss53)
$ git commit -a -m 'Finished the new footer'
[iss53 7483233] Finished the new footer
1 file changed, 3 insertions(+)

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (iss53)
$ |
```

Рисунок 13 - Индексация и коммит index.html на ветке iss53

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (iss53)
$ git checkout master
Switched to branch 'master'

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git merge iss53
Updating 2f780f6..16ea1b3
Fast-forward
 index.html | 1 +
1 file changed, 1 insertion(+)
```

Рисунок 14 - Слияние веток

После этого ветку iss53 нужно удалить (рисунок 15).

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git branch -d iss53
Deleted branch iss53 (was 97393ba).

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

Рисунок 15 - Удаление ветки iss53

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).


```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop_two
$ git branch
fatal: not a git repository (or any of the parent directories): .git

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

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

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop_two
$ git branch -v
fatal: not a git repository (or any of the parent directories): .git

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

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).

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop_two
$ git branch --merged
fatal: not a git repository (or any of the parent directories): .git

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

```
Толик@DESKTOP-J475S8M MINGW64 ~/desktop/shop_two
$ git branch --no-merged
fatal: not a git repository (or any of the parent directories): .git

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

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

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 63 - Удаление ветки

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

```
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 64 - 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 65 - 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 66 - 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 67 - Создание ветки на основе удаленной ветки

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

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two/repo1 (serv)
$ git push origin :serv
To https://github.com/Eternity-blip/repo1
- [deleted]          serv
```

Рисунок 68 - 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.

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

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (exp)
$ git rebase master
Successfully rebased and updated refs/heads/exp.
```

Рисунок 69 - 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.

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git rebase --onto master se cl
Successfully rebased and updated refs/heads/cl.
```

Рисунок 70 - Перемещение изменений с параметром onto

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

e1ena@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 71 - Слияние веток master и cl

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git rebase master se
Successfully rebased and updated refs/heads/se.
```

Рисунок 72 - Перемещение изменений

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

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git merge se
Updating 08d256f..4a98db2
Fast-forward
 e.txt | 0
 q.txt | 0
 rep01 | 1 +
3 files changed, 1 insertion(+)
create mode 100644 e.txt
create mode 100644 q.txt
create mode 160000 rep01
```

Рисунок 73 - Слияние веток master и se

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

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git branch -d cl
Deleted branch cl (was 08d256f).

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git branch -d se
Deleted branch se (was 4a98db2).

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

Рисунок 74 - Удаление веток cl и se