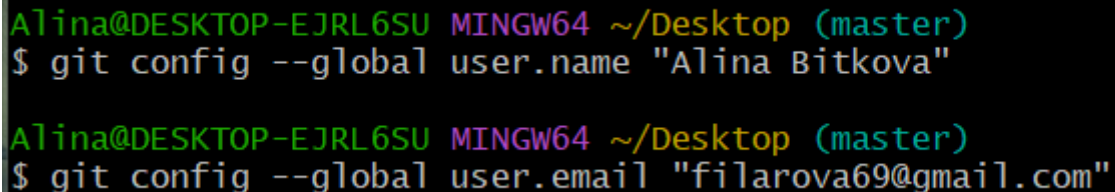


## \_УП\_1

### Работа с Git

#### Подготовка

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

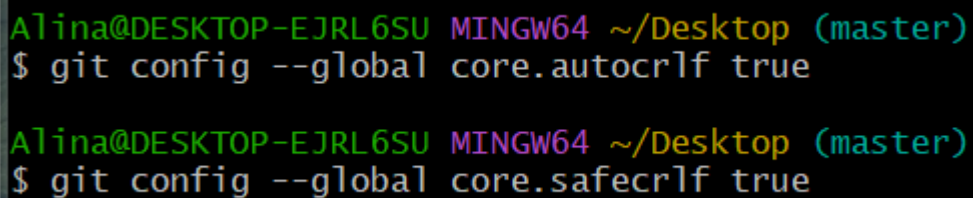


```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ git config --global user.name "Alina Bitkova"

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ git config --global user.email "filarova69@gmail.com"
```

Рисунок 1 - Установка имени и электронной почты

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

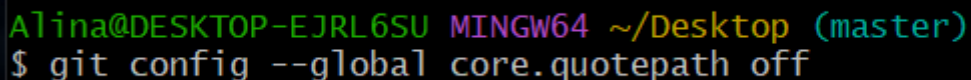


```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ git config --global core.autocrlf true

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ git config --global core.safecrlf true
```

Рисунок 2 - Параметры окончаний строк

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



```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ git config --global core.quotePath off
```

Рисунок 3 - Установка отображения unicode

## Создание проекта

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ mkdir hello

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ cd hello

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ touch hello.html
```

Рисунок 4 - Создание каталога и файла

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

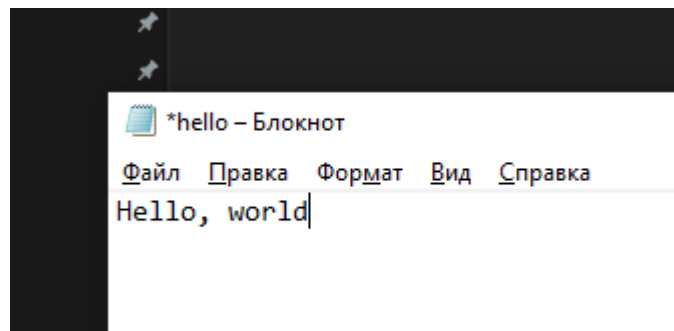


Рисунок 5 - Содержание файла

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

```
HELEN V@LAPTOP-BPH65Q86 MINGW64 ~/Desktop/hello
$ git init
Initialized empty Git repository in C:/Users/HELEN V/Desktop/hello/.git/

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

Рисунок 6

## - Создание репозитория

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -m "First Commit"
[master (root-commit) 8464bf7] First Commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 hello.html
```

Рисунок 7 - Добавление в репозиторий

### Проверка состояния

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.

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Рисунок 8 - Проверка состояния репозитория

### Внесение изменений

Сначала необходимо внести изменения в файл (рисунок 9).

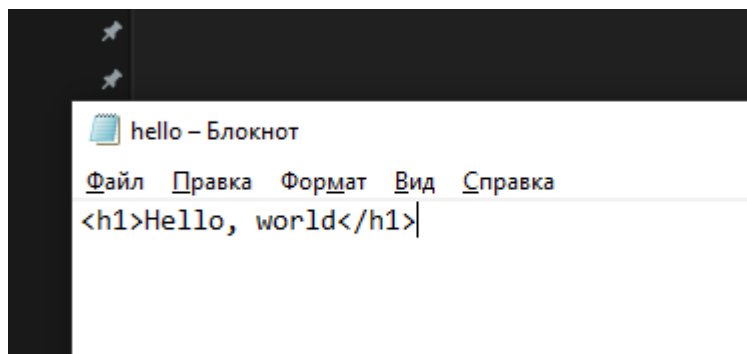


Рисунок 9 - Внесение изменений в файл

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

```
Alina@DESKTOP-EJRL6SU 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")
```

Рисунок 10 - Сообщение о незафиксированных изменениях

### Индексация изменений

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

Рисунок 11 - Команды для индексации изменений

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`.

## Индексация и коммит

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

Имя	Дата изменения	Тип
a	06.06.2025 12:36	Opera GX Web Do.
b	06.06.2025 12:37	Opera GX Web Do.
hello	06.06.2025 12:32	Opera GX Web Do.
v	06.06.2025 12:38	Opera GX Web Do.

Рисунок 12 - Создано 3 файла

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add a.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add b.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -m "Changes for a and b"
[master c098ca6] Changes for a and b
3 files changed, 1 insertion(+)
create mode 100644 a.html
create mode 100644 b.html
```

Рисунок 13 - Индексация и коммит для 2 файлов

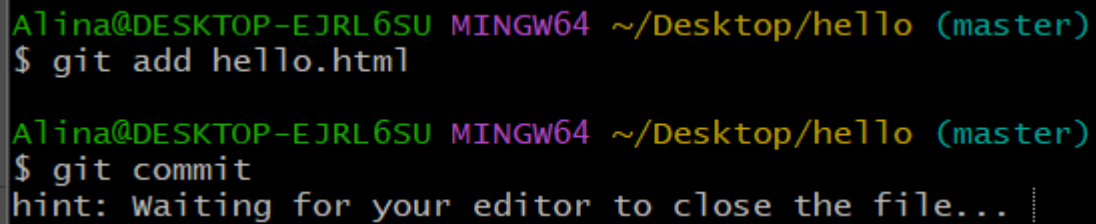
```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add v.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -m "Unrelated change for v"
[master d51374e] Unrelated change for v
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 v.html
```

Рисунок 14 - Индексация и коммит для третьего файла

## Коммит изменений

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

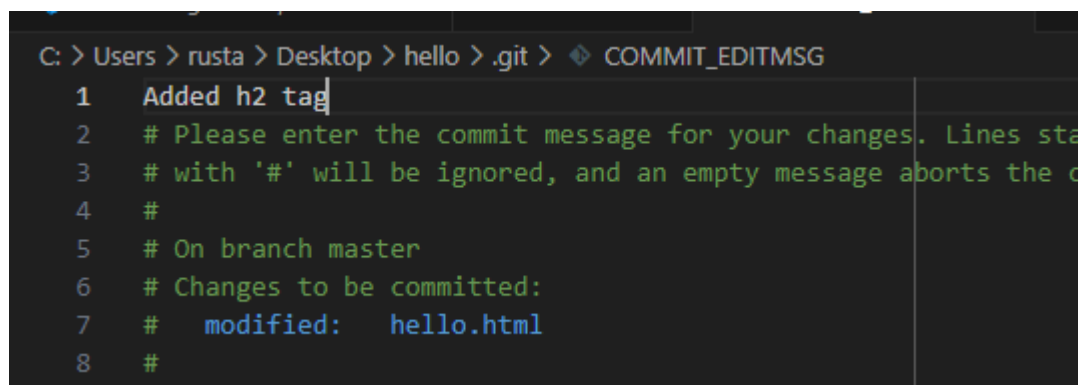


```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit
hint: Waiting for your editor to close the file... !
```

Рисунок 15 - Коммит изменений

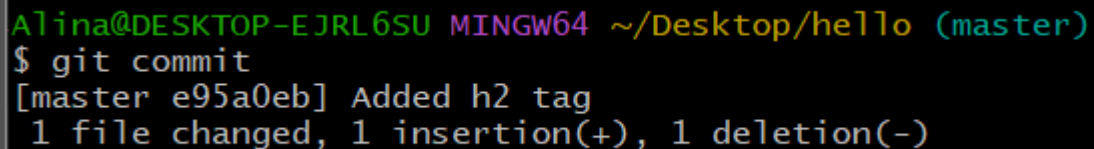
После этого будет открыт редактор, в котором необходимо на 1 строке записать комментарий коммита (рисунок 16).



```
C: > Users > rusta > Desktop > hello > .git > COMMIT_EDITMSG
1  Added h2 tag
2  # Please enter the commit message for your changes. Lines sta
3  # with '#' will be ignored, and an empty message aborts the c
4  #
5  # On branch master
6  # Changes to be committed:
7  #   modified:   hello.html
8  #
```

Рисунок 16 - Ввод комментария

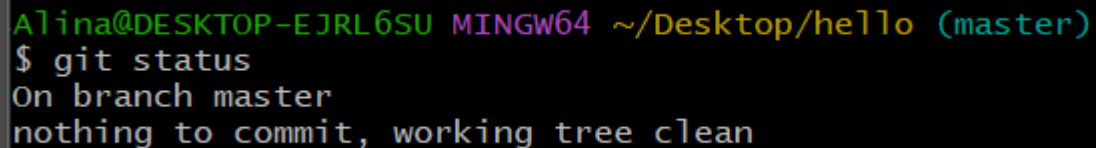
После выхода из текстового редактора будет указано следующее сообщение (рисунок 17).



```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit
[master e95a0eb] Added h2 tag
1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 17 - Коммит-сообщения

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



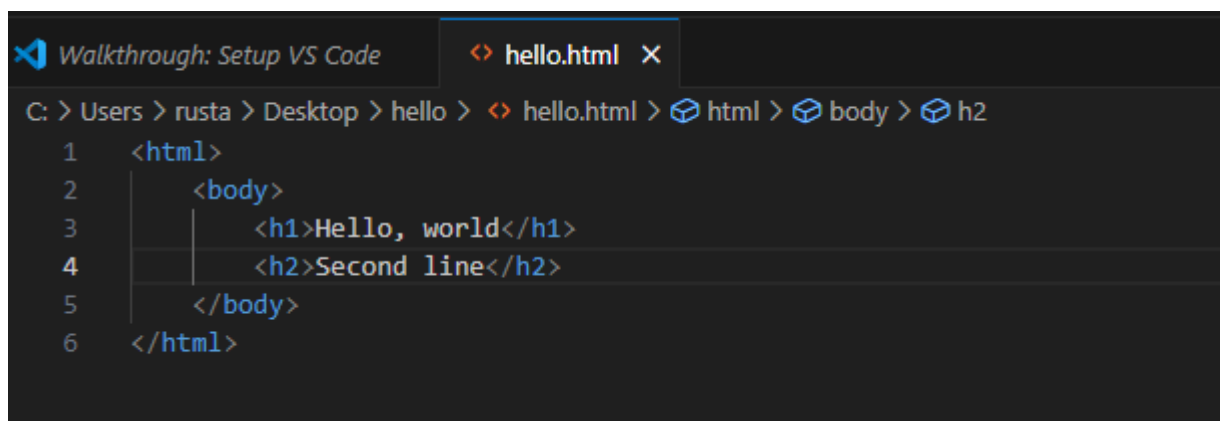
```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Рисунок 18 - 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.



The screenshot shows the Visual Studio Code editor with a file named 'hello.html' open. The breadcrumb navigation at the top indicates the path: 'C: > Users > rusta > Desktop > hello > hello.html > html > body > h2'. The code in the editor is as follows:

```
1 <html>
2   <body>
3     <h1>Hello, world</h1>
4     <h2>Second line</h2>
5   </body>
6 </html>
```

Рисунок 19 - Добавление тегов html и body

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



The screenshot shows the Visual Studio Code editor with the same 'hello.html' file. The breadcrumb navigation now includes 'head': 'C: > Users > rusta > Desktop > hello > hello.html > html > head'. The code in the editor is as follows:

```
1 <html>
2   <head>
3
4 </head>
5   <body>
6     <h1>Hello, world</h1>
7     <h2>Second line</h2>
8   </body>
9 </html>
```

Рисунок 20 - 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.

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

```
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 21 - Проверка состояния

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -m "Add standard HTML tags"
[master 4a76271] Add standard HTML tags
1 file changed, 9 insertions(+), 1 deletion(-)
```

```
Alina@DESKTOP-EJRL6SU 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")
```

Рисунок 22 - Коммит 1 изменения и проверка состояния



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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add .

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

Рисунок 23 - Индексация 2 изменения и проверка состояния

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -m "Add HTML header"
[master 5cad596] Add HTML header
1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 24 - Коммит 2 изменения

\_уп\_2

## Работа с Git

### История

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git log
commit 5cad5961606d0597f0d76cbf73b2b6c4be7cc47e (HEAD -> master)
Author: Alina Bitkova <filarova69@gmail.com>
Date:   Fri Jun 6 14:05:21 2025 +0300

    Add HTML header

commit 4a7627113c015ac76ed0f016330367ca41e21a85
Author: Alina Bitkova <filarova69@gmail.com>
Date:   Fri Jun 6 13:58:11 2025 +0300

    Add standard HTML tags

commit e95a0eb3d42b50af2d58decc6c411653125ef0ce
Author: Alina Bitkova <filarova69@gmail.com>
Date:   Fri Jun 6 13:43:02 2025 +0300

    Added h2 tag

commit d51374e11ee75b5707be00013d1231c20c068055
Author: Alina Bitkova <filarova69@gmail.com>
Date:   Fri Jun 6 12:39:40 2025 +0300

    Unrelated change for v

commit c098ca6fa449d593f4af6addb6f9953d5e08d503
Author: Alina Bitkova <filarova69@gmail.com>
Date:   Fri Jun 6 12:38:16 2025 +0300

```

Рисунок 25 - Просмотр истории изменений

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git log --pretty=oneline
5cad5961606d0597f0d76cbf73b2b6c4be7cc47e (HEAD -> master) Add HTML header
4a7627113c015ac76ed0f016330367ca41e21a85 Add standard HTML tags
e95a0eb3d42b50af2d58decc6c411653125ef0ce Added h2 tag
d51374e11ee75b5707be00013d1231c20c068055 Unrelated change for v
c098ca6fa449d593f4af6addb6f9953d5e08d503 Changes for a and b
8464bf7e14798c1d8e54f59b93068bd080c877e3 First Commit

```

Рисунок 26 - Однострочный формат вывода

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git log --pretty=oneline --max-count=2
5cad5961606d0597f0d76cbf73b2b6c4be7cc47e (HEAD -> master) Add HTML header
4a7627113c015ac76ed0f016330367ca41e21a85 Add standard HTML tags
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)

```

Рисунок 27 - Вывод последних 2 изменений

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git log --pretty=oneline --since='10 days ago'
5cad5961606d0597f0d76cbf73b2b6c4be7cc47e (HEAD -> master) Add HTML header
4a7627113c015ac76ed0f016330367ca41e21a85 Add standard HTML tags
e95a0eb3d42b50af2d58decc6c411653125ef0ce Added h2 tag
d51374e11ee75b5707be00013d1231c20c068055 Unrelated change for v
c098ca6fa449d593f4af6addb6f9953d5e08d503 Changes for a and b
8464bf7e14798c1d8e54f59b93068bd080c877e3 First Commit

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git log --pretty=oneline --since='12 days ago'
5cad5961606d0597f0d76cbf73b2b6c4be7cc47e (HEAD -> master) Add HTML header
4a7627113c015ac76ed0f016330367ca41e21a85 Add standard HTML tags
e95a0eb3d42b50af2d58decc6c411653125ef0ce Added h2 tag
d51374e11ee75b5707be00013d1231c20c068055 Unrelated change for v
c098ca6fa449d593f4af6addb6f9953d5e08d503 Changes for a and b
8464bf7e14798c1d8e54f59b93068bd080c877e3 First Commit

```

Рисунок 28 - Вывод изменений начиная с определенного времени

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git log --pretty=oneline --until='5 minutes ago'
5cad5961606d0597f0d76cbf73b2b6c4be7cc47e (HEAD -> master) Add HTML header
4a7627113c015ac76ed0f016330367ca41e21a85 Add standard HTML tags
e95a0eb3d42b50af2d58decc6c411653125ef0ce Added h2 tag
d51374e11ee75b5707be00013d1231c20c068055 Unrelated change for v
c098ca6fa449d593f4af6addb6f9953d5e08d503 changes for a and b
8464bf7e14798c1d8e54f59b93068bd080c877e3 First Commit

```

Рисунок 29 - Вывод изменений до определенного времени

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git log --pretty=oneline --author='Alina Bitkova'
5cad5961606d0597f0d76cbf73b2b6c4be7cc47e (HEAD -> master) Add HTML header
4a7627113c015ac76ed0f016330367ca41e21a85 Add standard HTML tags
e95a0eb3d42b50af2d58decc6c411653125ef0ce Added h2 tag
d51374e11ee75b5707be00013d1231c20c068055 Unrelated change for v
c098ca6fa449d593f4af6addb6f9953d5e08d503 Changes for a and b
8464bf7e14798c1d8e54f59b93068bd080c877e3 First Commit

```

Рисунок 30 - Вывод изменений, внесенных определенным автором

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git log --pretty=oneline --all
5cad5961606d0597f0d76cbf73b2b6c4be7cc47e (HEAD -> master) Add HTML header
4a7627113c015ac76ed0f016330367ca41e21a85 Add standard HTML tags
e95a0eb3d42b50af2d58decc6c411653125ef0ce Added h2 tag
d51374e11ee75b5707be00013d1231c20c068055 Unrelated change for v
c098ca6fa449d593f4af6addb6f9953d5e08d503 Changes for a and b
8464bf7e14798c1d8e54f59b93068bd080c877e3 First Commit

```

Рисунок 31 - Вывод всех изменений

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git log --all --pretty=format:"%h %cd %s (%an)" --since='12 days ago'
5cad596 Fri Jun 6 14:05:21 2025 +0300 Add HTML header (Alina Bitkova)
4a76271 Fri Jun 6 13:58:11 2025 +0300 Add standard HTML tags (Alina Bitkova)
e95a0eb Fri Jun 6 13:43:02 2025 +0300 Added h2 tag (Alina Bitkova)
d51374e Fri Jun 6 12:39:40 2025 +0300 Unrelated change for v (Alina Bitkova)
c098ca6 Fri Jun 6 12:38:16 2025 +0300 Changes for a and b (Alina Bitkova)
8464bf7 Fri Jun 6 12:30:27 2025 +0300 First Commit (Alina Bitkova)
```

Рисунок 32 - Использование нескольких параметров

## Алиасы

Для настройки алиасов используется команда, показанная на рисунке 9.

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git config --global alias.co checkout

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git config --global alias.ci commit

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git config --global alias.st status

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git config --global alias.br branch

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git config --global alias.hist "log --pretty=format:'%h %ad | %s%d [%an]' --graph --date=short"

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git config --global alias.type 'cat-file -t'

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git config --global alias.dump 'cat-file -p'

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ |
```

Рисунок 33 - Настройка алиасов для некоторых команд

При выполнении алиаса будет выполнена определенная команда и выведены нужные данные (рисунок 10).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git config --global alias.hist "log --pretty=format:'%h %ad | %s%d [%an]' --graph --date=short"

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist
* 5cad596 2025-06-06 | Add HTML header (HEAD -> master) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
```

Рисунок 34 - Выполнение алиаса hist

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ alias gs='git status'

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ gs
On branch master
nothing to commit, working tree clean

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
```

Рисунок 35 - Установка и выполнение алиаса gs

## Получение старых версий

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).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist
* 5cad596 2025-06-06 | Add HTML header (HEAD -> master) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]
```

Рисунок 36 - Просмотр хэшей предыдущих версий

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).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git checkout 8464bf7
Note: switching to '8464bf7'.

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 8464bf7 First Commit
```

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello ((8464bf7...))
$ cat hello.html
<h1>Hello, world</h1>
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello ((8464bf7...))
```

Рисунок 37 - Возвращение к нужной версии и просмотр содержимого файла

## Возвращение к последней версии в ветке master

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello ((5f509c1...))  
$ git checkout master
```

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)  
$ cat hello.html  
<html>  
  <head>  
  
  </head>  
  <body>  
    <h1>Hello, world</h1>  
    <h2>Second line</h2>dfghjk1  
  </body>  
</html>
```

Рисунок 38 - Возвращение к последней версии в ветке master

### Создание тегов версий

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

```
</html>  
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)  
$ git tag ver1
```

Рисунок 39 - Задание тега

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (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 4a76271 Add standard HTML tags
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello ((4a76271...))
$ cat hello.html
<html>
  <head>

  </head>
  <body>
    <h1>Hello, world</h1>
    <h2>Second line</h2>dfghjk
  </body>
</html>
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello ((4a76271...))

```

Рисунок 40 – Переход к предыдущей версии с помощью тега

```

</html>
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello ((4a76271...))
$ git tag ver1-beta

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello ((ver1-beta))

```

Рисунок 41 - Задание тега предыдущей версии

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello ((ver1-beta))
$ git checkout ver1
Previous HEAD position was 4a76271 Add standard HTML tags
HEAD is now at 5cad596 Add HTML header

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello ((ver1))
$ git checkout ver1-beta
Previous HEAD position was 5cad596 Add HTML header
HEAD is now at 4a76271 Add standard HTML tags

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello ((ver1-beta))

```

Рисунок 42 - Переключение между версиями с помощью тегов

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







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

```
Alina@DESKTOP-EJRL6SU 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 47 - 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.

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git checkout hello.html
Updated 1 path from the index

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git st
On branch master
nothing to commit, working tree clean

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ cat hello.html
<html>
  <head>

  </head>
  <body>
    <h1>Hello, world</h1>
    <h2>Second line</h2>dfghjk1
  </body>
</html>
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
```

Рисунок 48 - 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).

```
C: > Users > rusta > Desktop > hello > <> hello.html > html > head
1  <html>
2    <head>
3      <!-- 6e66e6ee6e66e-->
4    </head>
5    <body>
6      <h1>Hello, world</h1>
7      <h2>Second line</h2>
8    </body>
9  </html>
```

Рисунок 49 - Внесение ненужного изменения

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git st
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

Рисунок 50 - 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).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git reset HEAD hello.html
Unstaged changes after reset:
M      hello.html

Alina@DESKTOP-EJRL6SU 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 51 - Очистка буферной зоны

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git checkout hello.html
Updated 1 path from the index

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git st
On branch master
nothing to commit, working tree clean

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)

```

Рисунок 52 - 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).

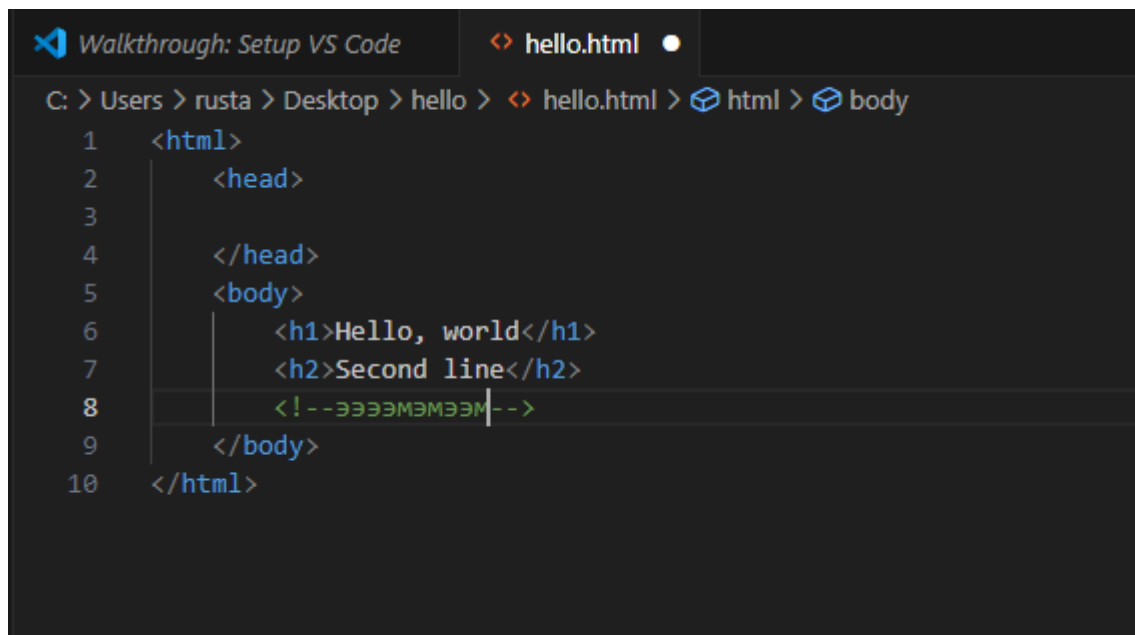


Рисунок 53 - Внесение изменения в файл

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -m "Oops, this is bad commit"
[master 79a5360] Oops, this is bad commit
1 file changed, 2 insertions(+), 1 deletion(-)

```

Рисунок 54 - 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).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git revert HEAD
hint: Waiting for your editor to close the file... ..
```

Рисунок 55 - Выполнение команды git revert

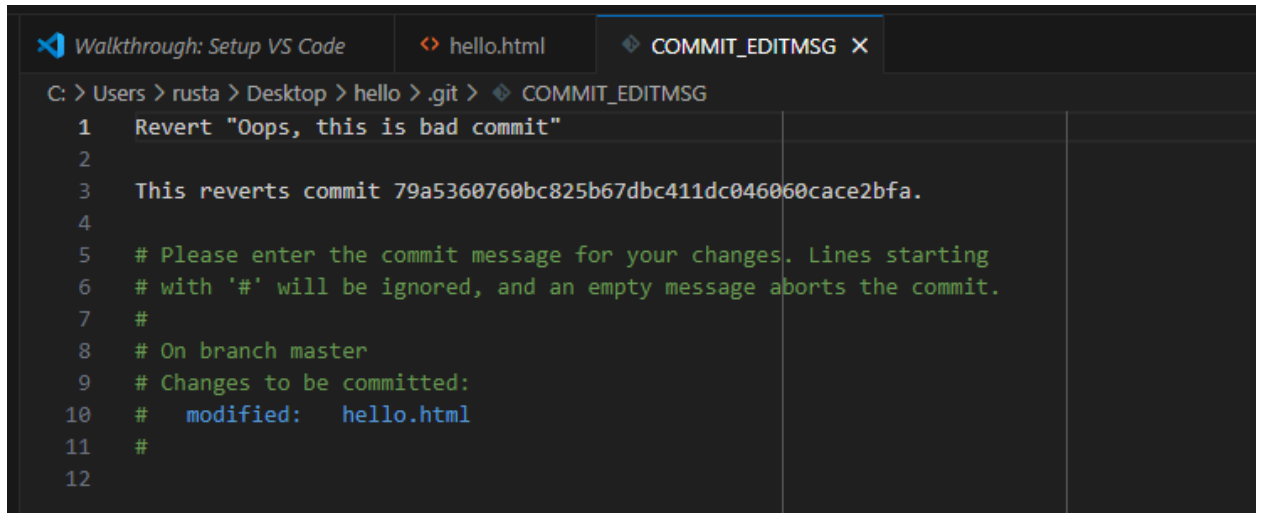


Рисунок 56 - Коммит сообщение в редакторе

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git revert HEAD
[master b54b735] Revert "Oops, this is bad commit"
1 file changed, 1 insertion(+), 2 deletions(-)

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
```

Рисунок 57 - Редактор закрыт

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist
* b54b735 2025-06-06 | Revert "Oops, this is bad commit" (HEAD -> master) [Alina Bitkova]
* 79a5360 2025-06-06 | Oops, this is bad commit [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
```

Рисунок 58 - Все коммиты при просмотре лога

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git tag oooops

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist
* b54b735 2025-06-06 | Revert "Oops, this is bad commit" (HEAD -> master, tag: oooops) [Alina Bitkova]
* 79a5360 2025-06-06 | Oops, this is bad commit [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]
```

Рисунок 59 - Отметка тегом коммита

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git reset --hard ver1
HEAD is now at 5cad596 Add HTML header

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist
* 5cad596 2025-06-06 | Add HTML header (HEAD -> master, tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]
```

Рисунок 60 - 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).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist --all
* b54b735 2025-06-06 | Revert "Oops, this is bad commit" (tag: oooops) [Alina Bitkova]
* 79a5360 2025-06-06 | Oops, this is bad commit [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (HEAD -> master, tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]
```

Рисунок 61 - 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).

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git tag -d oooops
Deleted tag 'oooops' (was b54b735)

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist --all
* 5cad596 2025-06-06 | Add HTML header (HEAD -> master, tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]

```

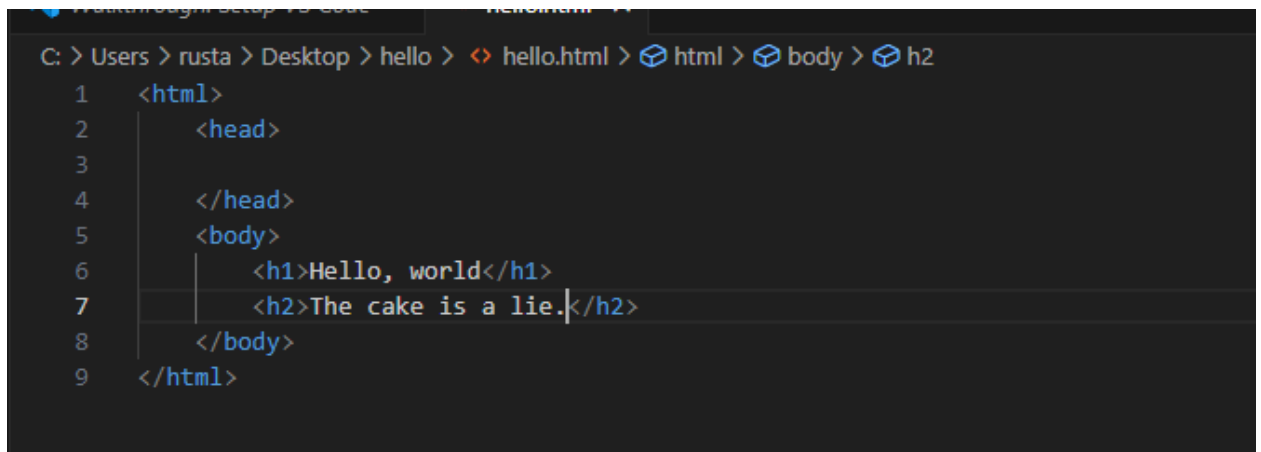
Рисунок 62 - Удаление тега

\_УП\_3

## 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.



```

C: > Users > rusta > Desktop > hello > hello.html > html > body > h2
1  <html>
2      <head>
3
4      </head>
5      <body>
6          <h1>Hello, world</h1>
7          <h2>The cake is a lie.</h2>
8      </body>
9  </html>

```

Рисунок 63 - Добавление комментария в файл

```

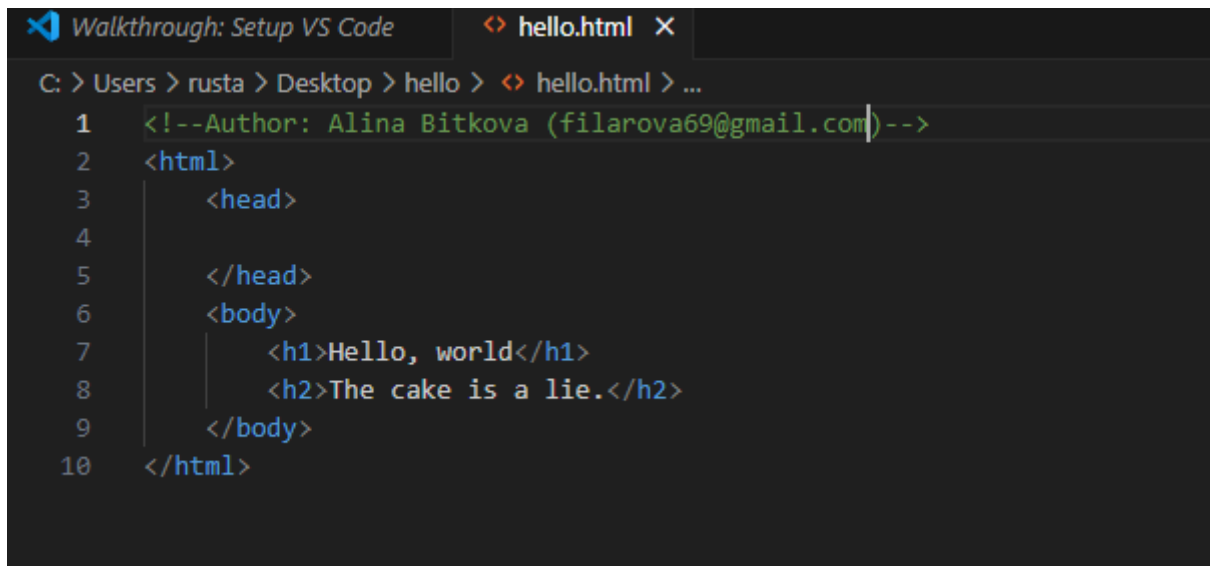
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -m "Add an author comment"
[master 6f2052b] Add an author comment
1 file changed, 1 insertion(+), 1 deletion(-)

```

Рисунок 64 - Индексация и коммит

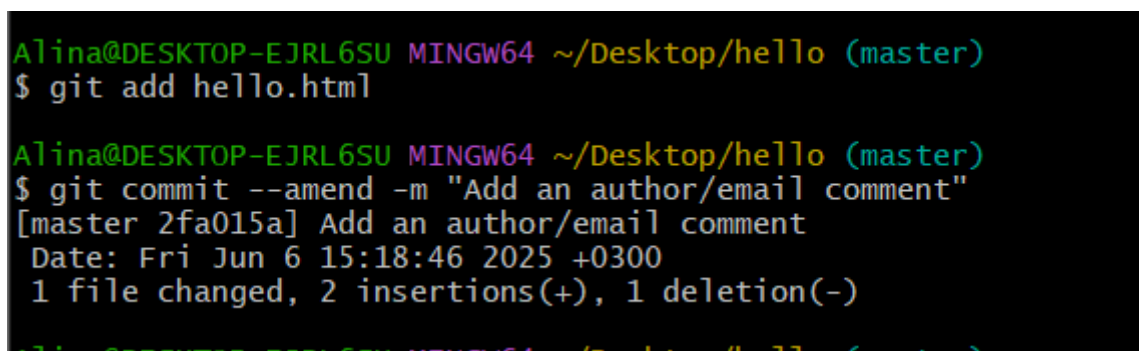
Далее необходимо добавить электронную почту в комментарий (рисунок 3).



```
1 <!--Author: Alina Bitkova (filarova69@gmail.com)-->
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 65 - Добавление электронной почты

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

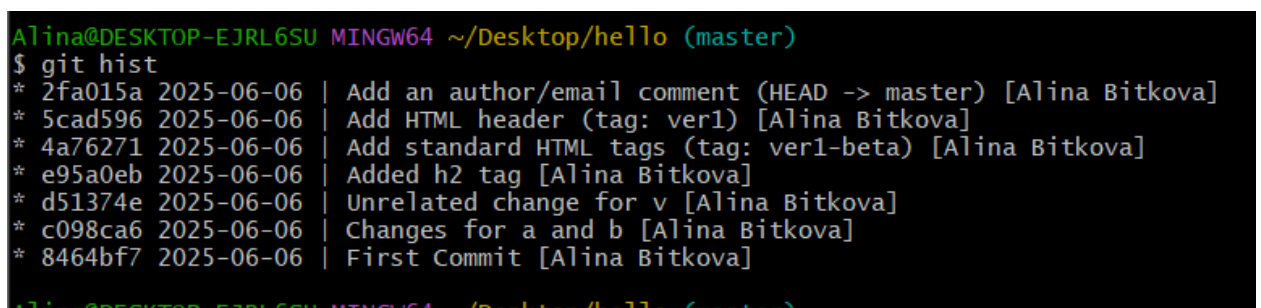


```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit --amend -m "Add an author/email comment"
[master 2fa015a] Add an author/email comment
Date: Fri Jun 6 15:18:46 2025 +0300
1 file changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 66 - Индексация и изменение коммита

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



```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist
* 2fa015a 2025-06-06 | Add an author/email comment (HEAD -> master) [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]
```

Рисунок 67 - 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).

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ mkdir lib

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git mv hello.html lib

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/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 68 - Перемещение файла

Далее надо осуществить коммит данного перемещения.

```


Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -m "Moved hello.html to lib"
[master 1f76223] Moved hello.html to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)

```

Рисунок 69 – Коммит перемещения

### [Подробнее о структуре](#)

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



```

Welcome  index.html X
C: > Users > rusta > Desktop > hello > index.html > html
1  <html>
2    <body>
3      <iframe src="lib/hello.html" width="200" height="200"></iframe>
4    </body>
5  </html>

```

Рисунок 70 - Содержимое файла index.html

Далее нужно проиндексировать и закоммитить файл (рисунок 9).



```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add index.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -m "Added index.html"
[master 3d1286b] Added index.html
1 file changed, 5 insertions(+)
create mode 100644 index.html
```

Рисунок 71 - Индексация и коммит

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

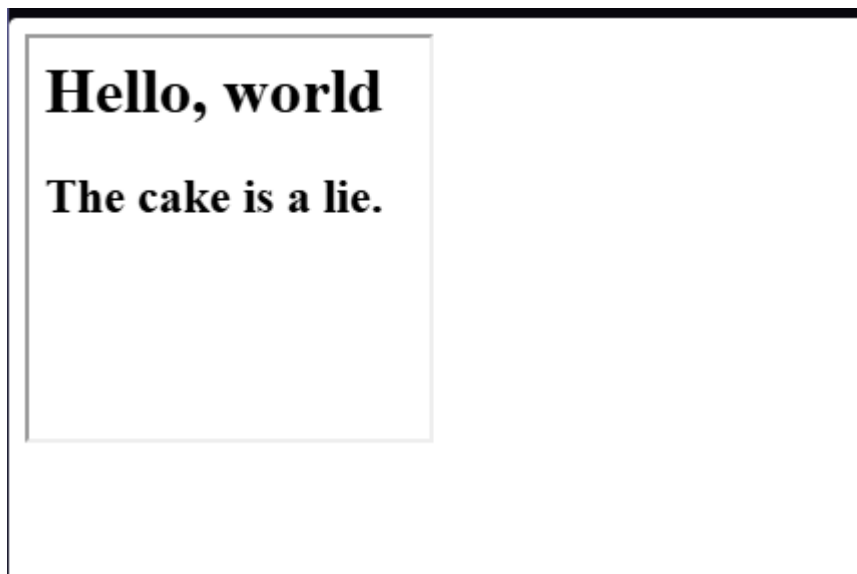


Рисунок 72 - Файл index.html, открытый в браузере

## Каталог .git

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ ls -C .git
COMMIT_EDITMSG HEAD ORIG_HEAD config description hooks/ index info/ logs/ objects/ refs/
```

Рисунок 73 - 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.

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ ls -C .git/objects
02/ 12/ 1f/ 22/ 2f/ 39/ 4a/ 5f/ 6f/ 73/ 84/ 8f/ a4/ bc/ c8/ dd/ e9/ fc/ pack/
05/ 1b/ 21/ 26/ 35/ 3d/ 5c/ 68/ 72/ 79/ 8d/ 9b/ b5/ c0/ d5/ e6/ ef/ info/
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
```

Рисунок 74 - Содержание каталога objects

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ ls -C .git/objects
02/ 12/ 1f/ 22/ 2f/ 39/ 4a/ 5f/ 6f/ 73/ 84/ 8f/ a4/ bc/ c8/ dd/ e9/ fc/ pack/
05/ 1b/ 21/ 26/ 35/ 3d/ 5c/ 68/ 72/ 79/ 8d/ 9b/ b5/ c0/ d5/ e6/ ef/ info/

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ ls -C .git/objects/8d
5990924e7abfeafa423bc9b1e535582263686b
```

Рисунок 75 - Просмотр каталога 8d

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
```

Рисунок 76 - Просмотр файла конфигурации

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ ls .git/refs
heads/  tags/
```

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ ls .git/refs/heads
master

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ ls .git/refs/tags
ver1  ver1-beta

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ cat .git/refs/tags/ver1
5cad5961606d0597f0d76cbf73b2b6c4be7cc47e
```

Рисунок 77 - Просмотр файлов и веток

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ cat .git/HEAD
ref: refs/heads/master

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
```

Рисунок 78 - Содержимое файла HEAD

### Работа с объектами git

Для начала необходимо посмотреть последний коммит (рисунок 17).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist --max-count=1
* 3d1286b 2025-06-06 | Added index.html (HEAD -> master) [Alina Bitkova]
```

Рисунок 79 - 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.

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git cat-file -t 3d1286b
commit

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git cat-file -p 3d1286b
tree 5ffa7f4dac389de33401a4bf46a008ba6b558470
parent 1f76223ea8aab9eb408509272275c6677e4a0f8a
author Alina Bitkova <filarova69@gmail.com> 1749213272 +0300
committer Alina Bitkova <filarova69@gmail.com> 1749213272 +0300

Added index.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)

```

Рисунок 80 - Просмотр объекта коммита

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git type 3d1286b
commit

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git dump 3d1286b
tree 5ffa7f4dac389de33401a4bf46a008ba6b558470
parent 1f76223ea8aab9eb408509272275c6677e4a0f8a
author Alina Bitkova <filarova69@gmail.com> 1749213272 +0300
committer Alina Bitkova <filarova69@gmail.com> 1749213272 +0300

Added index.html

```

Рисунок 81 - Использование алиасов

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git cat-file -p 5ffa7f4
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391    a.html
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391    b.html
100644 blob 8d5990924e7abfeafa423bc9b1e535582263686b    index.html
040000 tree 1228c5f9a3c6a5855bce24e8ea0e9c2309484aa3    lib
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391    v.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)

```

Рисунок 82 - Просмотр дерева каталогов

Затем нужно просмотреть каталог lib (рисунок 21).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git cat-file -p 1228c5f9a
100644 blob dd1dab474e752f170bb5945afec4b3161ec872b0    hello.html
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
```

Рисунок 83 - Просмотр каталога lib

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git cat-file -p dd1dab47
<!--Author: Alina Bitkova (filarova69@gmail.com)-->
<html>
  <head>

  </head>
  <body>
    <h1>Hello, world</h1>
    <h2>The cake is a lie.</h2>
  </body>
</html>
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
```

Рисунок 84 - 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.

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist
* 3d1286b 2025-06-06 | Added index.html (HEAD -> master) [Alina Bitkova]
* 1f76223 2025-06-06 | Moved hello.html to lib [Alina Bitkova]
* 2fa015a 2025-06-06 | Add an author/email comment [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git cat-file -p 8464bf7
tree 0207049fcff52f60055434766127a023b1bc978e
author Alina Bitkova <filarova69@gmail.com> 1749202227 +0300
committer Alina Bitkova <filarova69@gmail.com> 1749202227 +0300

First Commit

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git cat-file -p 0207049
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391    hello.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git cat-file -p e69de29
```

Рисунок 85 - Просмотр содержимого файла при первом коммите

\_УП\_4

## Работа с Git

### Создание ветки

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git checkout -b style
Switched to a new branch 'style'

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git status
On branch style
nothing to commit, working tree clean
```

Рисунок 86 - Создание новой ветки style

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ touch lib/style.css

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
```

Рисунок 87 - Создание файла стилей

```
C: > Users > rusta > Desktop > hello > lib > # style.css > h1
1  h1 {
2      color: red;
3  }
```

Рисунок 88 - Код style.css

После этого надо произвести индексацию и коммит (рисунок 4).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git add lib/style.css

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git commit -m "Added css stylesheet"
[style 5d62cd0] Added css stylesheet
1 file changed, 3 insertions(+)
create mode 100644 lib/style.css
```

Рисунок 89 - Индексация и коммит нового файла

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

```
C: > Users > rusta > Desktop > hello > lib > <> hello.html > html > head > link
1  <!--Author: Alina Bitkova (filarova69@gmail.com)-->
2  <html>
3      <head>
4          <link rel="stylesheet" type="text/css" media="all" href="style.css">
5      </head>
6      <body>
7          <h1>Hello, world</h1>
8          <h2>The cake is a lie.</h2>
9      </body>
10 </html>
```

Рисунок 90 - Изменения в файле hello.html

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git add lib/hello.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git commit -m "Hello.html uses style.css"
[style 90f11f8] Hello.html uses style.css
1 file changed, 1 insertion(+), 1 deletion(-)

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)

```

Рисунок 91 - Индексация и коммит

Далее аналогичные действия нужно осуществить с файлом index.html, как это показано на рисунках 7-8.

```

C: > Users > rusta > Desktop > hello > index.html > html > head > link
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>

```

Рисунок 92 - Изменения в файле index.html

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git add index.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git commit -m "Update index.html"
[style 720b266] Update index.html
1 file changed, 3 insertions(+)

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)

```

Рисунок 93 - Индексация и коммит

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.



```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git hist --all
* 720b266 2025-06-06 | Update index.html (HEAD -> style) [Alina Bitkova]
* 90f11f8 2025-06-06 | Hello.html uses style.css [Alina Bitkova]
* 5d62cd0 2025-06-06 | Added css stylesheet [Alina Bitkova]
* 3d1286b 2025-06-06 | Added index.html (master) [Alina Bitkova]
* 1f76223 2025-06-06 | Moved hello.html to lib [Alina Bitkova]
* 2fa015a 2025-06-06 | Add an author/email comment [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]
```

Рисунок 94 - Просмотр истории

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).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git checkout master
Switched to branch 'master'

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ cat lib/hello.html
<!--Author: Alina Bitkova (filarova69@gmail.com)-->
<html>
  <head>

  </head>
  <body>
    <h1>Hello, world</h1>
    <h2>The cake is a lie.</h2>
  </body>
</html>
```

Рисунок 95 - Переключение на ветку master

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

```

</html>
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git checkout style
Switched to branch 'style'

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ cat lib/hello.html
<!--Author: Alina Bitkova (filarova69@gmail.com)-->
<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>
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ |

```

Рисунок 96 - Переключение на ветку style

#### Изменения в ветке master

Необходимо переключиться на ветку master (рисунок 12) и добавить файл README (рисунки 13-14).

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git checkout master
Switched to branch 'master'

```

Рисунок 97 - Переключение на ветку master

```

C: > Users > rusta > Desktop > hello > ⓘ README
1 This is the Hello World example from the Git tutorial

```

Рисунок 98 - Содержимое файла README

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add README

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -m "Added README"
[master 7f810c2] Added README
1 file changed, 1 insertion(+)
create mode 100644 README

```

Рисунок 99 - Индексация и коммит

## Просмотр отличающихся веток

На рисунке 15 можно увидеть дерево коммитов.

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist --all
* 7f810c2 2025-06-06 | Added README (HEAD -> master) [Alina Bitkova]
| * 720b266 2025-06-06 | Update index.html (style) [Alina Bitkova]
| * 90f11f8 2025-06-06 | Hello.html uses style.css [Alina Bitkova]
| * 5d62cd0 2025-06-06 | Added css stylesheet [Alina Bitkova]
|/
* 3d1286b 2025-06-06 | Added index.html [Alina Bitkova]
* 1f76223 2025-06-06 | Moved hello.html to lib [Alina Bitkova]
* 2fa015a 2025-06-06 | Add an author/email comment [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]
```

Рисунок 100 - Дерево коммитов

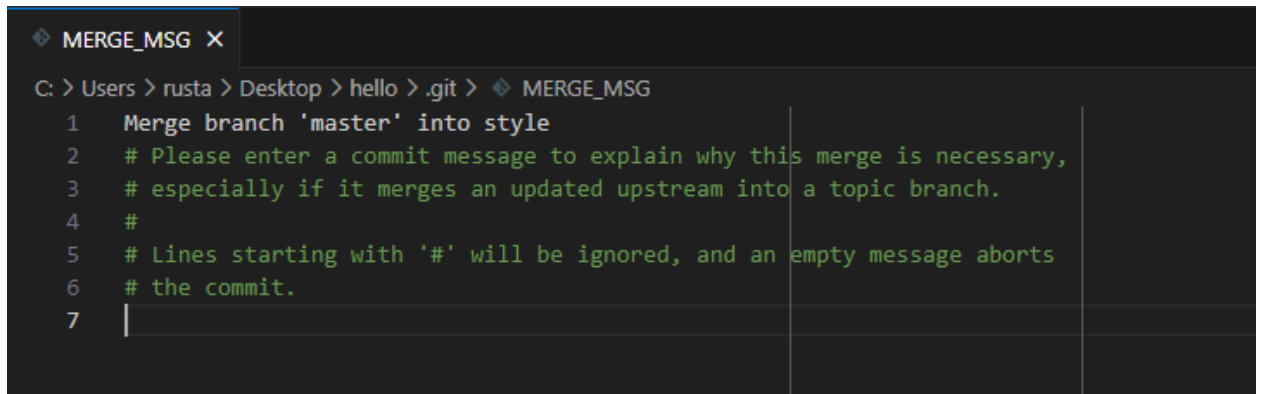
## Слияние

Слияние переносит изменения из двух веток в одну. Для слияния нужно перейти на ветку style и с помощью команды `git merge master` совместить ветки (рисунки 16-18).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git checkout style
Switched to branch 'style'
```

Рисунок 101 - Переключение на ветку style

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git merge master
hint: Waiting for your editor to close the file... |
```



```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git merge master
Merge made by the 'ort' strategy.
 README | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README
```

Рисунок 102 - Слияние с веткой master

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git hist --all
* a45d1fe 2025-06-06 | Merge branch 'master' into style (HEAD -> style) [Alina Bitkova]
|
| * 7f810c2 2025-06-06 | Added README (master) [Alina Bitkova]
| * 720b266 2025-06-06 | Update index.html [Alina Bitkova]
| * 90f11f8 2025-06-06 | Hello.html uses style.css [Alina Bitkova]
| * 5d62cd0 2025-06-06 | Added css stylesheet [Alina Bitkova]
|/
* 3d1286b 2025-06-06 | Added index.html [Alina Bitkova]
* 1f76223 2025-06-06 | Moved hello.html to lib [Alina Bitkova]
* 2fa015a 2025-06-06 | Add an author/email comment [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]
```

Рисунок 103 - Просмотр истории

## Создание конфликта

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git checkout master
Switched to branch 'master'
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
```

Рисунок 104 - Переход в ветку master

```
C: > Users > rusta > Desktop > hello > lib > <> hello.html > html > body > h1
1  <!--Author: Alina Bitkova (filarova69@gmail.com)-->
2  <html>
3      <head>
4          <!--no style-->
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 105 - Внесение изменений

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add lib/hello.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -m 'Life is great!'
[master 42dd5a9] Life is great!
1 file changed, 2 insertions(+), 2 deletions(-)
```

Рисунок 106 - Индексация и коммит

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist --all
* 42dd5a9 2025-06-06 | Life is great! (HEAD -> master) [Alina Bitkova]
| * a45dlfe 2025-06-06 | Merge branch 'master' into style (style) [Alina Bitkova]
| | \
| | /
| | /
| * 7f810c2 2025-06-06 | Added README [Alina Bitkova]
| * 720b266 2025-06-06 | Update index.html [Alina Bitkova]
| * 90f11f8 2025-06-06 | Hello.html uses style.css [Alina Bitkova]
| * 5d62cd0 2025-06-06 | Added css stylesheet [Alina Bitkova]
| /
* 3d1286b 2025-06-06 | Added index.html [Alina Bitkova]
* 1f76223 2025-06-06 | Moved hello.html to lib [Alina Bitkova]
* 2fa015a 2025-06-06 | Add an author/email comment [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]

```

Рисунок 107 - Конфликт изменений

### Разрешение конфликтов

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git checkout style
Switched to branch 'style'

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git merge master
Auto-merging lib/hello.html
CONFLICT (content): Merge conflict in lib/hello.html
Automatic merge failed; fix conflicts and then commit the result.

```

Рисунок 108 - Ошибка при слиянии

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

```

C: > Users > rusta > Desktop > hello > lib > <> hello.html > <> html > <> body
1  <!--Author: Alina Bitkova (filarova69@gmail.com)-->
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 109 - Просмотр файла hello.html при наличии конфликта

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

```

C: > Users > rusta > Desktop > hello > lib > <> hello.html > <html> > <head> > <link
1  <!--Author: Alina Bitkova (filarova69@gmail.com)-->
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 110 - Решение конфликта вручную

Затем следует произвести индексацию и коммит (рисунок 26).

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style|MERGING)
$ git add lib/hello.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style|MERGING)
$ git commit -m "Merged master fixed conflict."
[style 01c9ccb] Merged master fixed conflict.

```

Рисунок 111 - Индексация и коммит

\_уп\_5

## Работа с Git

### Сброс ветки style

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git checkout style
Already on 'style'

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git hist
* 01c9ccb 2025-06-06 | Merged master fixed conflict. (HEAD -> style) [Alina Bitkova]
| \
| * 42dd5a9 2025-06-06 | Life is great! (master) [Alina Bitkova]
* | a45d1fe 2025-06-06 | Merge branch 'master' into style [Alina Bitkova]
| \
| * 7f810c2 2025-06-06 | Added README [Alina Bitkova]
* | 720b266 2025-06-06 | Update index.html [Alina Bitkova]
* | 90f11f8 2025-06-06 | Hello.html uses style.css [Alina Bitkova]
* | 5d62cd0 2025-06-06 | Added css stylesheet [Alina Bitkova]
| /
* 3d1286b 2025-06-06 | Added index.html [Alina Bitkova]
* 1f76223 2025-06-06 | Moved hello.html to lib [Alina Bitkova]
* 2fa015a 2025-06-06 | Add an author/email comment [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]

```

Рисунок 112 - Просмотр истории

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git reset --hard 3d1286b
HEAD is now at 3d1286b Added index.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git hist --all
* 42dd5a9 2025-06-06 | Life is great! (master) [Alina Bitkova]
* 7f810c2 2025-06-06 | Added README [Alina Bitkova]
* 3d1286b 2025-06-06 | Added index.html (HEAD -> style) [Alina Bitkova]
* 1f76223 2025-06-06 | Moved hello.html to lib [Alina Bitkova]
* 2fa015a 2025-06-06 | Add an author/email comment [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]

```

Рисунок 113 - Сброс ветки style

### Сброс ветки master

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



```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git checkout master
Switched to branch 'master'

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist
* 42dd5a9 2025-06-06 | Life is great! (HEAD -> master) [Alina Bitkova]
* 7f810c2 2025-06-06 | Added README [Alina Bitkova]
* 3d1286b 2025-06-06 | Added index.html (style) [Alina Bitkova]
* 1f76223 2025-06-06 | Moved hello.html to lib [Alina Bitkova]
* 2fa015a 2025-06-06 | Add an author/email comment [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]

```

Рисунок 114 - Переключение на master и просмотр истории

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git reset --hard 42dd5a9
HEAD is now at 42dd5a9 Life is great!

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist --all
* 42dd5a9 2025-06-06 | Life is great! (HEAD -> master) [Alina Bitkova]
* 7f810c2 2025-06-06 | Added README [Alina Bitkova]
* 3d1286b 2025-06-06 | Added index.html (style) [Alina Bitkova]
* 1f76223 2025-06-06 | Moved hello.html to lib [Alina Bitkova]
* 2fa015a 2025-06-06 | Add an author/email comment [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]

```

Рисунок 115 - Сброс ветки master

## Перебазирование

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git checkout style
Switched to branch 'style'

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git rebase master
Successfully rebased and updated refs/heads/style.

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git hist
* 42dd5a9 2025-06-06 | Life is great! (HEAD -> style, master) [Alina Bitkova]
* 7f810c2 2025-06-06 | Added README [Alina Bitkova]
* 3d1286b 2025-06-06 | Added index.html [Alina Bitkova]
* 1f76223 2025-06-06 | Moved hello.html to lib [Alina Bitkova]
* 2fa015a 2025-06-06 | Add an author/email comment [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]

```

Рисунок 116 - Перебазирование веток

#### Слияние в ветку master

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (style)
$ git checkout master
Switched to branch 'master'

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git merge style
Already up to date.

```

Рисунок 117 - Слияние веток

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git hist
* 42dd5a9 2025-06-06 | Life is great! (HEAD -> master, style) [Alina Bitkova]
* 7f810c2 2025-06-06 | Added README [Alina Bitkova]
* 3d1286b 2025-06-06 | Added index.html [Alina Bitkova]
* 1f76223 2025-06-06 | Moved hello.html to lib [Alina Bitkova]
* 2fa015a 2025-06-06 | Add an author/email comment [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]

```

Рисунок 118 - 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.

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ git clone hello cloned_hello
Cloning into 'cloned_hello'...
done.

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ ls
Decocraft-2.6.3.7_1.12.2-2n.jar      hello/
'Genshin Impact.lnk'*              ptrlib-1_0_5.jar
'HoYoPlay.lnk'*                     r2modman.lnk*
'Life Is Strange.lnk'*              shaders/
'Opera GX.lnk'*                     'vlad kadoni.r2z'
'Prilozhenie 1.1 Git (1) э.docx'     'win10 new look'/
'Prilozhenie 1.2 Git э.docx'         zapret-discord-youtube-main/
'Prilozhenie 1.3 Git э.docx'         'Доп согл на повышение цены.pdf'
'Prilozhenie 1.4 Git (1) э.docx'     'Доп соглашение на мат капитал.pdf'
Roblox.lnk*                         'Евдокимова Дарья. Реферат 2.docx'
'Schedule I.lnk'*                   'Евдокимова Дарья. Реферат.docx'
VSCodeUserSetup-x64-1.100.3.exe*    'Игры По Сети.lnk'*
Yandex.lnk*                         'Лаунчер Life Makeover.lnk'*
Zona.lnk*                           'Этот компьютер.lnk'*
cloned_hello/                       курсоры/
desktop.ini                          обои/
```

Рисунок 119 - Переход в рабочий каталог и его клонирование

## Просмотр клонированного репозитория

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ cd cloned_hello

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ ls
README  a.html  b.html  index.html  lib/  v.html
```

Рисунок 120 - Просмотр содержимого клонированного репозитория

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git hist --all
* 42dd5a9 2025-06-06 | Life is great! (HEAD -> master, origin/style, origin/master, origin/HEAD) [Alina Bitkova]
* 7f810c2 2025-06-06 | Added README [Alina Bitkova]
* 3d1286b 2025-06-06 | Added index.html [Alina Bitkova]
* 1f76223 2025-06-06 | Moved hello.html to lib [Alina Bitkova]
* 2fa015a 2025-06-06 | Add an author/email comment [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]
```

Рисунок 121 - Просмотр логов клонированного каталога

## Origin

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git remote
origin

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git remote show origin
* remote origin
  Fetch URL: C:/Users/rusta/Desktop/hello
  Push URL: C:/Users/rusta/Desktop/hello
  HEAD branch: master
  Remote branches:
    master tracked
    style 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)
```

Рисунок 122 - Просмотр данных об origin

## Удаленные ветки

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git branch
* master

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/style
```

Рисунок 123 - Просмотр удаленных веток

## Работа с Git

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ cd ../hello
```

Рисунок 124 - Переход в оригинальный репозиторий

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

```
C: > Users > rusta > Desktop > hello > ⓘ README
1 This is the Hello World example from the Git tutorial
2 [changed in original]
```

Рисунок 125 - Изменения в файле README

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add README

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -m "Changed README in original repo"
[master c064f4f] Changed README in original repo
1 file changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 126 - Индексация и коммит новых изменений

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ cd ../cloned_hello

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 345 bytes | 2.00 KiB/s, done.
From C:/Users/rusta/Desktop/hello
 42dd5a9..c064f4f master    -> origin/master
```

Рисунок 127 - Извлечение изменений

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git hist --all
* c064f4f 2025-06-06 | Changed README in original repo (origin/master, origin/HEAD) [Alina Bitkova]
* 42dd5a9 2025-06-06 | Life is great! (HEAD -> master, origin/style) [Alina Bitkova]
* 7f810c2 2025-06-06 | Added README [Alina Bitkova]
* 3d1286b 2025-06-06 | Added index.html [Alina Bitkova]
* 1f76223 2025-06-06 | Moved hello.html to lib [Alina Bitkova]
* 2fa015a 2025-06-06 | Add an author/email comment [Alina Bitkova]
* 5cad596 2025-06-06 | Add HTML header (tag: ver1) [Alina Bitkova]
* 4a76271 2025-06-06 | Add standard HTML tags (tag: ver1-beta) [Alina Bitkova]
* e95a0eb 2025-06-06 | Added h2 tag [Alina Bitkova]
* d51374e 2025-06-06 | Unrelated change for v [Alina Bitkova]
* c098ca6 2025-06-06 | Changes for a and b [Alina Bitkova]
* 8464bf7 2025-06-06 | First Commit [Alina Bitkova]
```

Рисунок 128 - Просмотр истории

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ cat README
This is the Hello world example from the Git tutorial
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
```

Рисунок 129 - Вывод содержимого файла README

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git merge origin/master
Updating 42dd5a9..c064f4f
Fast-forward
 README | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 130 - Слияние изменений

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ cat README
This is the Hello world example from the Git tutorial
(changed in original)
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
```

Рисунок 131 - Вывод содержимого файла README

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git pull
Already up to date.
```

Рисунок 132 - Команда git pull

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git branch --track style origin/style
branch 'style' set up to track 'origin/style'.

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git branch -a
* master
  style
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/style

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git hist --max-count=2
* c064f4f 2025-06-06 | Changed README in original repo (HEAD -> master, origin/master, origin/HEAD) [Alina Bitkova]
* 42dd5a9 2025-06-06 | Life is great! (origin/style, style) [Alina Bitkova]

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ |
```

Рисунок 133 - Добавление локальной ветки

Далее необходимо создать чистый репозиторий (рисунок 11).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ cd ..

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ git clone --bare hello hello.git
Cloning into bare repository 'hello.git'...
done.

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ ls hello.git
HEAD config description hooks/ info/ objects/ packed-refs refs/
```

Рисунок 134 - Создание чистого репозитория

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ cd hello

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git remote add shared ../hello.git
```

Рисунок 135 - Добавление удаленного репозитория

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



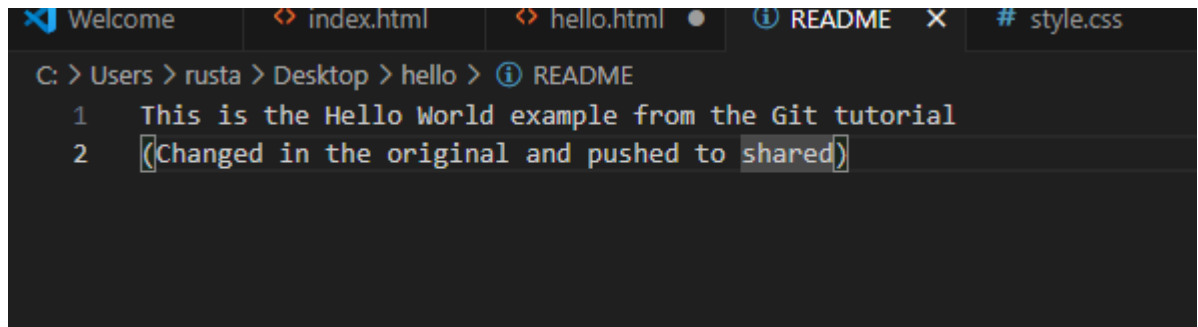


Рисунок 136 - Внесение изменений в файл

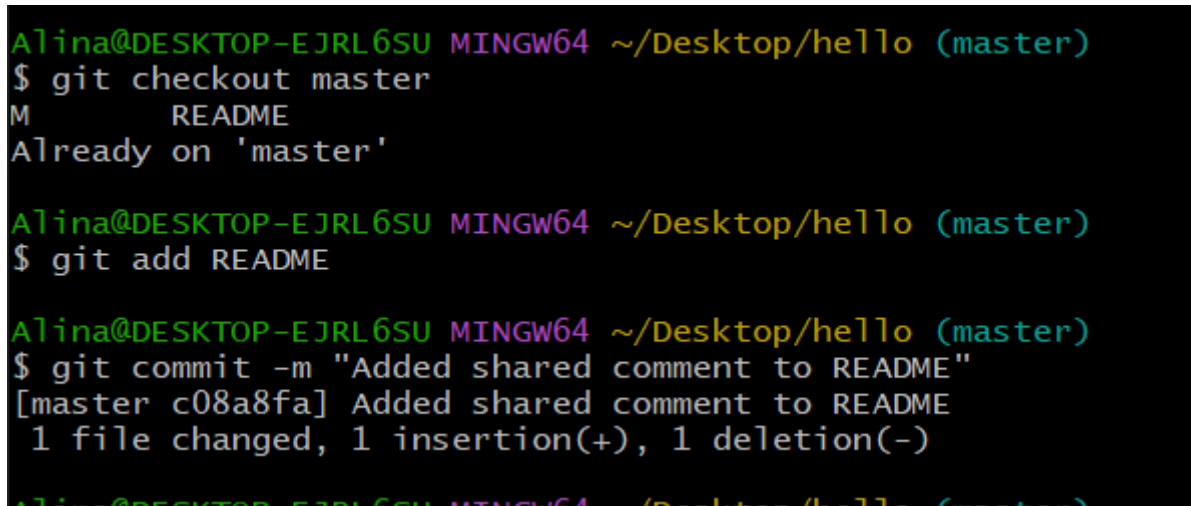


Рисунок 137 - Индексация и коммит

Далее надо отправить изменения в общий репозиторий, используя команду `git push shared master` (рисунок 15).

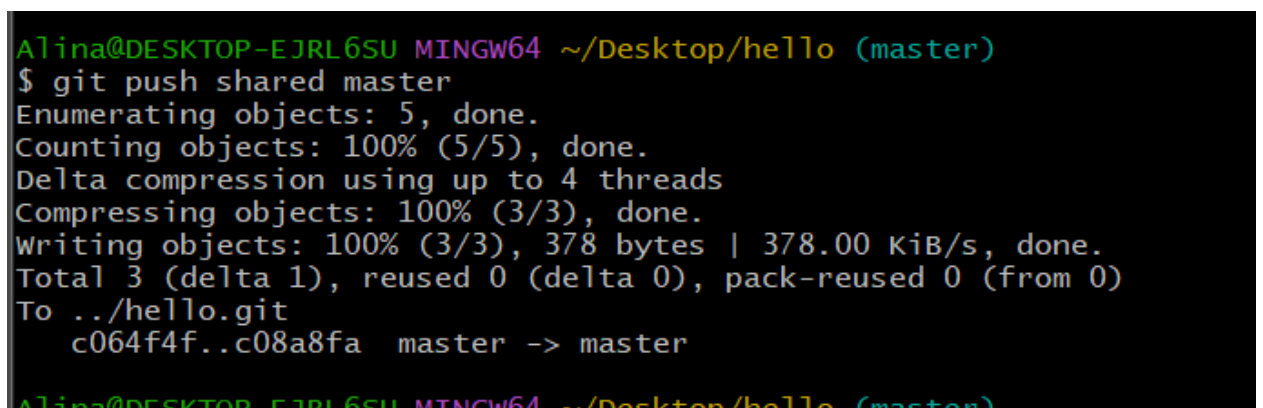


Рисунок 138 - Отправка изменений в общий репозиторий

Для извлечения общих изменений нужно перейти в клонированный каталог и выполнить перечень команд (рисунок 16).



```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ cd ../cloned_hello

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ |
```

Рисунок 139 - Переход в клонированный репозиторий

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git remote add shared ../hello.git

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git branch --track shared master
branch 'shared' set up to track 'master'.

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git pull shared master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 358 bytes | 1024 bytes/s, done.
From ../hello
 * branch                master      -> FETCH_HEAD
 * [new branch]          master      -> shared/master
Updating c064f4f..c08a8fa
Fast-forward
 README | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ cat README
This is the Hello World example from the Git tutorial
(Changed in the original and pushed to shared)
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ |
```

Рисунок 140 - Команды, извлекающие общие изменения

Для настройки git сервера нужно выполнить команду, показанную на рисунке 18. Затем в другом окне можно проверить работу сервера, сделав копию проекта hello (рисунок 18).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/cloned_hello (master)
$ git daemon --verbose --export-all --base-path=.
[16792] Ready to rumble
```

Рисунок 141 - Настройка сервера

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ git clone ~/Desktop/hello.git network_hello
Cloning into 'network_hello'...
done.

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop (master)
$ cd network_hello

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello (master)
$ ls
README  a.html  b.html  index.html  lib/  v.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello (master)
```

Рисунок 142 - Клонирование проекта

\_уп\_7

## Работа с Git

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello (master)
$ 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 | 2.39 MiB/s, done.
Resolving deltas: 100% (837/837), done.
```

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello (master)
$ cd ticgit

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello/ticgit (master)
$ git remote
origin

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello/ticgit (master)
```

Рисунок 143 – Клонирование репозитория и просмотр удаленных репозиторий

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)
```

Рисунок 144 - Просмотр удаленных репозиторий с ключом `-v`

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello/ticgit (master)
$ git remote add pb https://github.com/paulboone/ticgit

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello/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 145 - Добавление удаленного репозитория

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello/ticgit (master)
$ git fetch pb
remote: Enumerating objects: 43, done.
remote: Counting objects: 100% (22/22), done.
remote: Total 43 (delta 22), reused 22 (delta 22), pack-reused 21 (from 1)
Unpacking objects: 100% (43/43), 5.99 KiB | 3.00 KiB/s, done.
From https://github.com/paulboone/ticgit
 * [new branch]      master    -> pb/master
 * [new branch]      ticgit     -> pb/ticgit
```

Рисунок 146 - Использование имени вместо пути

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello/ticgit (master)
$ git fetch ~/Desktop/cloned_hello
remote: Enumerating objects: 39, done.
remote: Counting objects: 100% (39/39), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 39 (delta 13), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (39/39), 3.47 KiB | 1024 bytes/s, done.
From C:/Users/rusta/Desktop/cloned_hello
 * branch            HEAD      -> FETCH_HEAD
```

Рисунок 147 - Получение данных из удаленных проектов

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

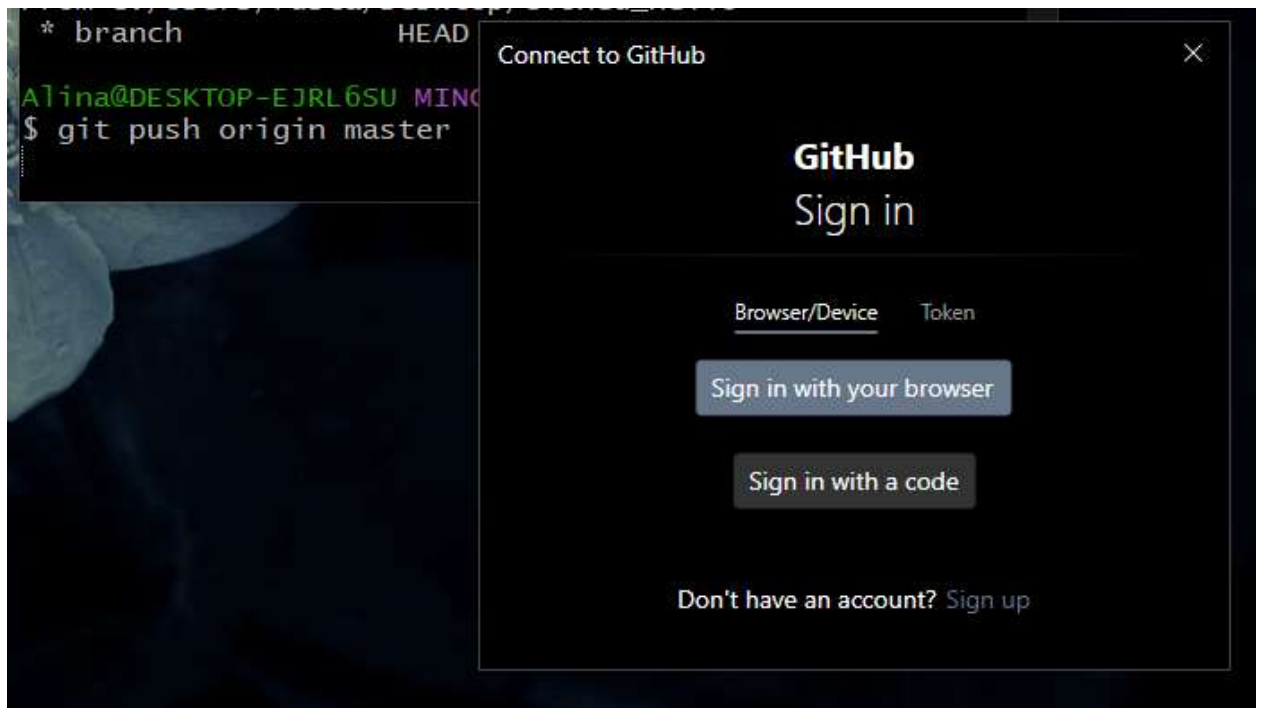


Рисунок 148 - Отправка изменений в удаленный репозиторий

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello/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)
```

Рисунок 149 - Информация об удаленном репозитории

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello/ticgit (master)
$ git remote rename pb paul
Renaming remote references: 100% (4/4), done.

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello/ticgit (master)
$ git remote
origin
paul
```

Рисунок 150 - Переименование удаленного репозитория

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello/ticgit (master)
$ git remote remove paul

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/network_hello/ticgit (master)
$ git remote
origin
```

Рисунок 151 - Удаление удаленного репозитория

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git tag
ver1
ver1-beta
```

Рисунок 152 - Просмотр тегов

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git tag -a v2.1 -m "My version 2.1"

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git tag
v2.1
ver1
ver1-beta
```

Рисунок 153 - Создание аннотированной метки

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git show v2.1
tag v2.1
Tagger: Alina Bitkova <filarova69@gmail.com>
Date:   Fri Jun 6 17:53:08 2025 +0300

My version 2.1

commit c08a8fadb096bd62717fe29fa589364843c6560c (HEAD -> master, tag: v2.1, shared/master)
Author: Alina Bitkova <filarova69@gmail.com>
Date:   Fri Jun 6 17:13:37 2025 +0300

    Added shared comment to README

diff --git a/README b/README
index f06bb8a..33b357b 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

```

Рисунок 154 - 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).

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git tag v2.1-lw

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git tag
v2.1
v2.1-lw
ver1
ver1-beta

```

Рисунок 155 - Создание легковесной метки

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git show v2.1-lw
commit c08a8fadf096bd62717fe29fa589364843c6560c (HEAD -> master, tag: v2.1-lw, tag: v2.1, sha
red/master)
Author: Alina Bitkova <filarova69@gmail.com>
Date:   Fri Jun 6 17:13:37 2025 +0300

    Added shared comment to README

diff --git a/README b/README
index f06bb8a..33b357b 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

```

Рисунок 156 - Просмотр данных тега

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git log --pretty=oneline
c08a8fadf096bd62717fe29fa589364843c6560c (HEAD -> master, tag: v2.1-lw, tag: v2.1, shared/mas
ter) Added shared comment to README
c064f4f7989b4db2dd5d6d761b081014eb7256e8 Changed README in original repo
42dd5a99dc98afe728e5aa150f6fccfbbf3e0626 (style) Life is great!
7f810c25a802deba18be4a2cdf54c93dafd290c9 Added README
3d1286b3d32434f7321ab7131cff6258b00d13d4 Added index.html
1f76223ea8aab9eb408509272275c6677e4a0f8a Moved hello.html to lib
2fa015a01f47f742206719effb0d52d056d65b7b Add an author/email comment
5cad5961606d0597f0d76cbf73b2b6c4be7cc47e (tag: ver1) Add HTML header
4a7627113c015ac76ed0f016330367ca41e21a85 (tag: ver1-beta) Add standard HTML tags
e95a0eb3d42b50af2d58decc6c411653125ef0ce Added h2 tag
d51374e11ee75b5707be00013d1231c20c068055 Unrelated change for v
c098ca6fa449d593f4af6addb6f9953d5e08d503 Changes for a and b
8464bf7e14798c1d8e54f59b93068bd080c877e3 First Commit

```

Рисунок 157 - Просмотр истории

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git tag -a ver3 c064f4f

```

Рисунок 158 - Создание тега определенному коммиту

```

MERGE_MSG TAG_EDITMSG
C: > Users > rusta > Desktop > 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 159 - Ввод сообщения в текстовом редакторе

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git show ver3
tag ver3
Tagger: Alina Bitkova <filarova69@gmail.com>
Date:   Fri Jun 6 17:58:39 2025 +0300

Tag message1

commit c064f4f7989b4db2dd5d6d761b081014eb7256e8 (tag: ver3)
Author: Alina Bitkova <filarova69@gmail.com>
Date:   Fri Jun 6 16:48:52 2025 +0300

    Changed README in original repo

diff --git a/README b/README
index 79e1713..f06bb8a 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

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ |
```

Рисунок 160 - Просмотр данных тега

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git push shared v2.1
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 167 bytes | 167.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To ../hello.git
 * [new tag]          v2.1 -> v2.1
```

Рисунок 161 - Отправка тега на удаленный сервер

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



```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git push shared --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 167 bytes | 167.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To ../hello.git
* [new tag]          v2.1-lw -> v2.1-lw
* [new tag]          ver3 -> ver3
```

Рисунок 162 - Отправка всех тегов на сервер

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git tag -d v2.1-lw
Deleted tag 'v2.1-lw' (was c08a8fa)
```

Рисунок 163 - Удаление тега

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git push shared :refs/tags/v2.1-lw
To ../hello.git
- [deleted]          v2.1-lw
```

Рисунок 164 - 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).

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/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 c08a8fa Added shared comment to README

```

Рисунок 165 - Переключение на метку

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello ((v2.1))
$ git checkout -b version2 v2.1
Switched to a new branch 'version2'

```

Рисунок 166 - Создание новой ветки

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

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git config --global alias.co checkout

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git config --global alias.br branch

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git config --global alias.ci commit

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git config --global alias.st status

```

Рисунок 167 - Задание алиасов

```

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git config --global alias.unstage 'reset HEAD --'

```

Рисунок 168 - Создание псевдонима исключения файла из индекса

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git unstage fileA

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git reset HEAD -- fileA
```

Рисунок 169 - Использование созданного псевдонима

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git config --global alias.last 'log -1 HEAD'
```

Рисунок 170 - Создание алиаса для просмотра последнего коммита

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git last
commit c08a8fadb096bd62717fe29fa589364843c6560c (HEAD -> version2, tag: v2.1, shared/master,
master)
Author: Alina Bitkova <filarova69@gmail.com>
Date:   Fri Jun 6 17:13:37 2025 +0300

    Added shared comment to README
```

Рисунок 171 - Результат работы созданного алиаса

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git config --global alias.visual '!gitk'

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
```

Рисунок 172 - Создание псевдонима внешней команды

\_УП\_8

## Работа с Git

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git add README test.rb LICENSE

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git commit -m "Initial commit of my project"
[version2 1092b5c] Initial commit of my project
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 LICENSE
create mode 100644 test.rb
```

Рисунок 1 - Индексация и коммит 3 файлов

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git branch testing
```

Рисунок 2 - Создание ветки testing

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (version2)
$ git checkout testing
Switched to branch 'testing'
```

Рисунок 3 - Переключение на ветку testing

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (testing)
$ git add test.rb

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (testing)
$ git commit -m 'New changes'
[testing 9c8cdfc] New changes
1 file changed, 1 insertion(+)
```

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).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (testing)
$ git checkout master
Switched to branch 'master'
```

Рисунок 5 - Переключение на ветку master

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add test.rb

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -a -m 'another changes'
[master 0a04ff4] another changes
1 file changed, 1 insertion(+)
create mode 100644 test.rb
```

Рисунок 6 - Еще индексация и коммит test.rb

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git checkout -b 'iss53'
Switched to a new branch 'iss53'
```

Рисунок 7 - Создание и переключение на ветку iss53

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (iss53)
$ git add index.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (iss53)
$ git commit -m 'Added html'
[iss53 7555055] Added html
1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 8 - Индексация и коммит файла index.html

Далее нужно переключить ветку на master (рисунок 9).

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).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (hotfix)
$ git add index.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (hotfix)
$ git commit -a -m 'Fixed the broken email address'
[hotfix 244e420] Fixed the broken email address
1 file changed, 1 insertion(+), 1 deletion(-)
```

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (hotfix)
$ git checkout master
Switched to branch 'master'

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git merge hotfix
Updating 0a04ff4..244e420
Fast-forward
 index.html | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

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

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git branch -d hotfix
Deleted branch hotfix (was 244e420).
```

Рисунок 12 - Удаление ветки hotfix

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git add index.html

Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git commit -a -m 'Finished the new footer'
[master 6a5b31a] Finished the new footer
1 file changed, 1 insertion(+), 1 deletion(-)
```

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

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git merge iss53
Already up to date.
```

Рисунок 14 - Слияние веток

После этого ветку iss53 нужно удалить (рисунок 15).

```
Alina@DESKTOP-EJRL6SU MINGW64 ~/Desktop/hello (master)
$ git branch -d iss53
Deleted branch iss53 (was 7555055).
```

Рисунок 15 - Удаление ветки iss53

\_УП\_9

## 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).

```
Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ git branch
* master
  style
  testing
  version2

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$
```

Рисунок 173 - Список существующих веток

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

```
Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ git branch -v
* master      5d3f273 Merge branch 'iss53'
  style       fa2a9ac Life is great!
  testing     47c6a30 Another changes
  version2    5220d80 Added shared comment to README

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ |
```

Рисунок 174 - 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).

```
Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ git branch --merged
* master
  style
  testing
  version2

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ |
```

Рисунок 175 - Список веток слитых с текущей

```
Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ git branch --no-merged

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ |
```

Рисунок 176 - Список веток не слитых с текущей

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

```
Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ git branch --d testing
Deleted branch testing (was 47c6a30).

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ |
```

Рисунок 177 - Удаление ветки

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

```
Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ git remote add origin https://github.com/Kamodov21/git-1.9.git

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/Kamodov21/git-1.9.git
  Push URL: https://github.com/Kamodov21/git-1.9.git
  HEAD branch: (unknown)

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$
```

Рисунок 178 - Просмотр удаленных веток

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

```
Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (main)
$ git push -u origin main
fatal: User cancelled dialog.
error: unable to read askpass response from 'D:/Git/mingw64/bin/git-askpass.exe'
Username for 'https://github.com':
error: unable to read askpass response from 'D:/Git/mingw64/bin/git-askpass.exe'
Password for 'https://github.com':
remote: No anonymous write access.
fatal: Authentication failed for 'https://github.com/Kamodov21/git-1.9.git/'
```

Рисунок 179 - Отправка изменений

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

```
Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (main)
$ git fetch origin
```

Рисунок 180 - Выполнение команды `git fetch`

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



```

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ git checkout -b serv shared/master
branch 'serv' set up to track 'shared/master'.
Switched to a new branch 'serv'

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (serv)
$

```

Рисунок 181 - Создание ветки на основе удаленной ветки

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

```

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (serv)
$ git push origin :serv
fatal: User cancelled dialog.
error: unable to read askpass response from 'D:/Git/mingw64/bin/git-askpass.exe'
Username for 'https://github.com':

```

Рисунок 182 - 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.

```

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (serv)
$ git branch -a
  main
  master
* serv
  style
  version2
  remotes/shared/master

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (serv)
$ git checkout -b exp
Switched to a new branch 'exp'

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (exp)
$

```

Рисунок 183 - 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.

```

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (se)
$ git checkout -b cl
Switched to a new branch 'cl'

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (cl)
$

```

Рисунок 184 - Перемещение изменений с параметром onto

```
Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ git merge c1
Already up to date.

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ |
```

Рисунок 185 - Слияние веток master и c1

```
Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ git rebase master se
Current branch se is up to date.

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (se)
$ |
```

Рисунок 186 - Перемещение изменений

```
Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (se)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'shared/master'.

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ git merge se
Already up to date.

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ |
```

Рисунок 187 - Слияние веток master и se

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

```
Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ git branch -d c1
Deleted branch c1 (was 5220d80).

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$ git branch -d se
Deleted branch se (was 5220d80).

Admin@DESKTOP-1S5EFVL MINGW64 ~/desktop/hello (master)
$
```

Рисунок 188 - Удаление веток c1 и se