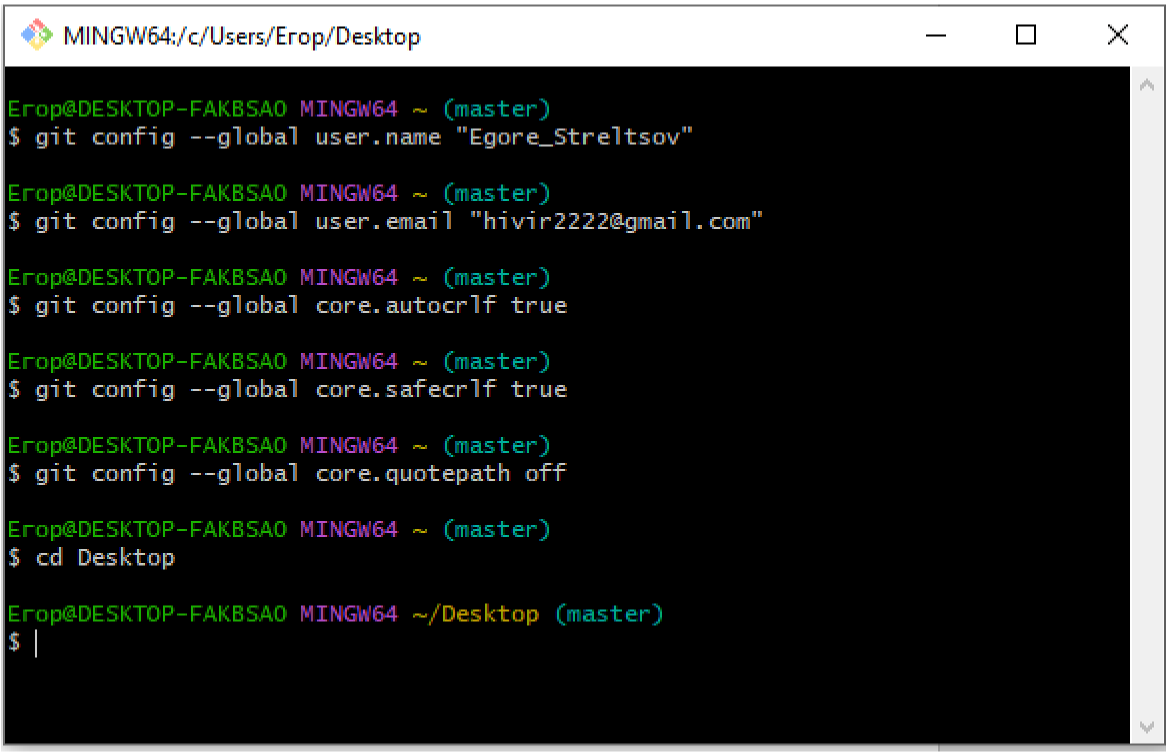
1.1

1. Настройка параметров Git

```
Егор@DESKTOP-FAKBSA0 MINGW64 ~ (master)
$ git config --global user.name "Egore_Streltsov"

Егор@DESKTOP-FAKBSA0 MINGW64 ~ (master)
$ git config --global user.email "hivir2222@gmail.com"

Егор@DESKTOP-FAKBSA0 MINGW64 ~ (master)
$ git config --global core.autocrlf true

Егор@DESKTOP-FAKBSA0 MINGW64 ~ (master)
$ git config --global core.safecrlf true

Егор@DESKTOP-FAKBSA0 MINGW64 ~ (master)
$ git config --global core.quotepath off
```

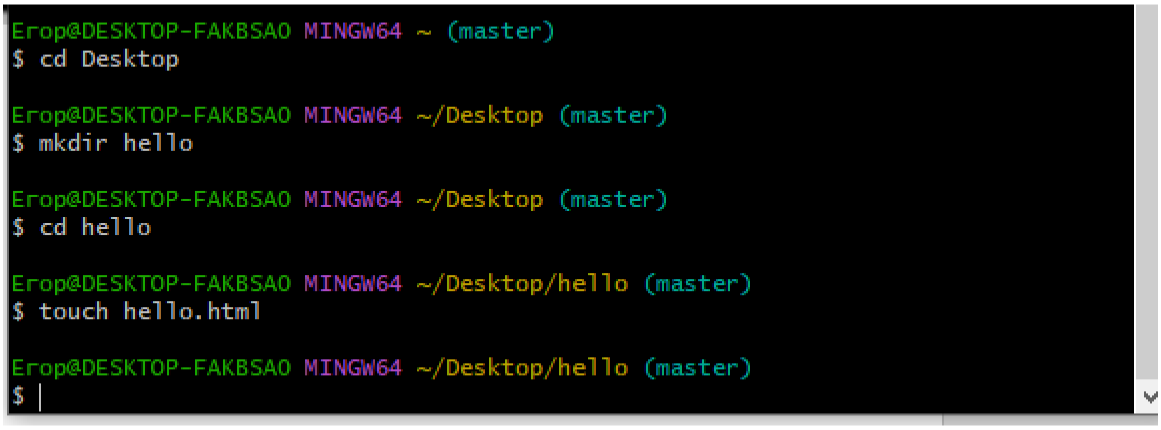


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~ (master)
$ git config --global user.name "Egore_Streltsov"

Егор@DESKTOP-FAKBSA0 MINGW64 ~ (master)
$ git config --global user.email "hivir2222@gmail.com"

Егор@DESKTOP-FAKBSA0 MINGW64 ~ (master)
$ git config --global core.autocrlf true

Егор@DESKTOP-FAKBSA0 MINGW64 ~ (master)
$ git config --global core.safecrlf true

Егор@DESKTOP-FAKBSA0 MINGW64 ~ (master)
$ git config --global core.quotepath off

Егор@DESKTOP-FAKBSA0 MINGW64 ~ (master)
$ cd Desktop

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop (master)
$ |
```

2. Создание проекта и файла hello.html

```
Егор@DESKTOP-FAKBSA0 MINGW64 ~ (master)
$ cd Desktop

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop (master)
```

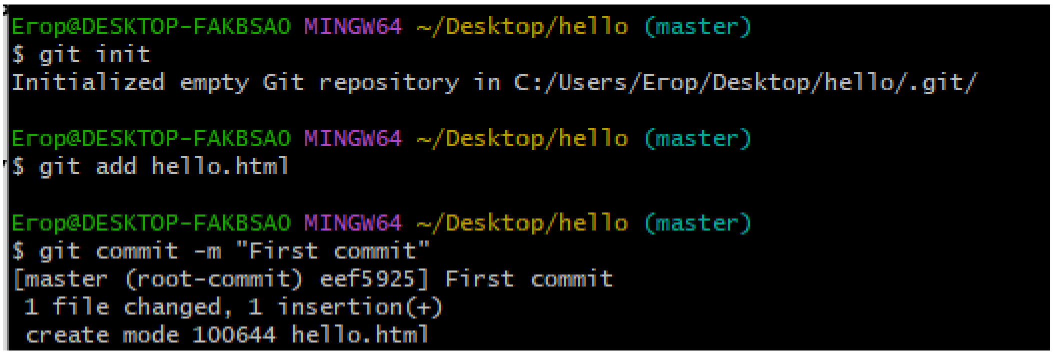
```
$ mkdir hello

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop (master)
$ cd hello

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ touch hello.html
```



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~ (master)
$ cd Desktop

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop (master)
$ mkdir hello

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop (master)
$ cd hello

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ touch hello.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ |
```

3. Инициализация Git, добавление страницы в репозиторий, добавление коммита

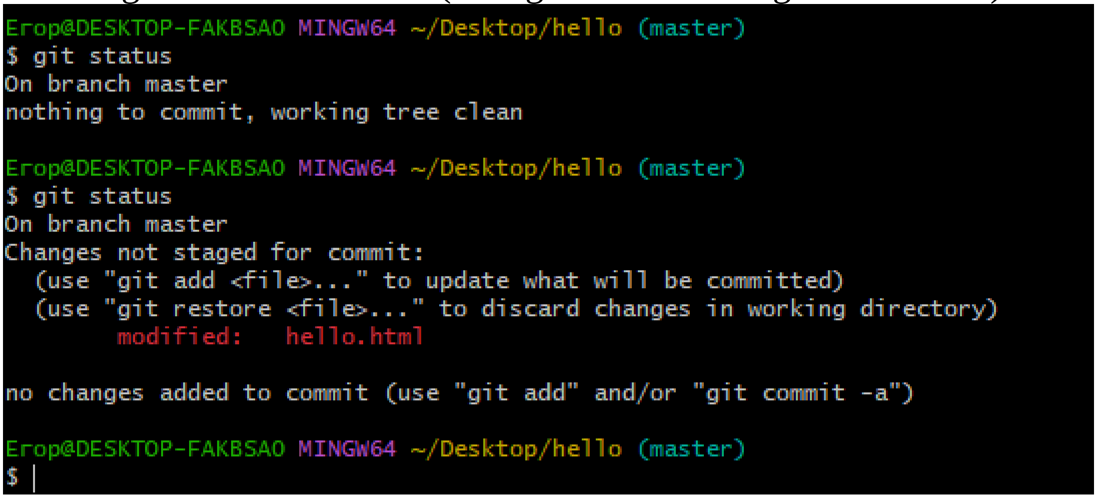
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git init
Initialized empty Git repository in C:/Users/Егор/Desktop/hello/.git/

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit -m "First commit"
[master (root-commit) eef5925] First commit
 1 file changed, 1 insertion(+)
 create mode 100644 hello.html
```



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git init
Initialized empty Git repository in C:/Users/Егор/Desktop/hello/.git/

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit -m "First commit"
[master (root-commit) eef5925] First commit
 1 file changed, 1 insertion(+)
 create mode 100644 hello.html
```

4. Проверка состояний до изменений и после внесения изменений

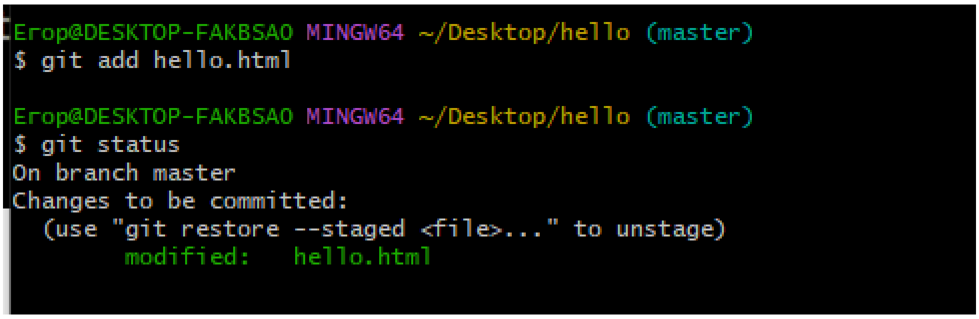
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
nothing to commit, working tree clean

Егор@DESKTOP-FAKBSA0 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")
```



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
nothing to commit, working tree clean

Егор@DESKTOP-FAKBSA0 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")

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ |
```

5. Индексация изменений

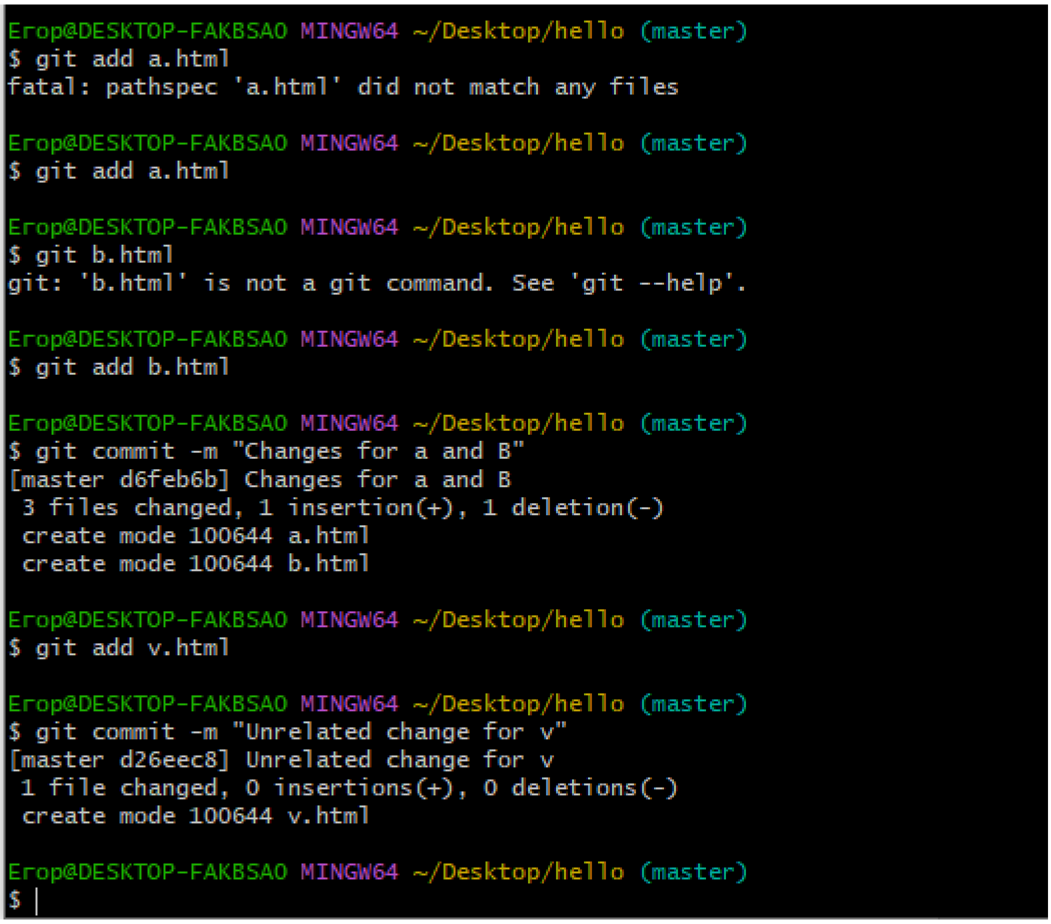
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

6. Индексация и коммит

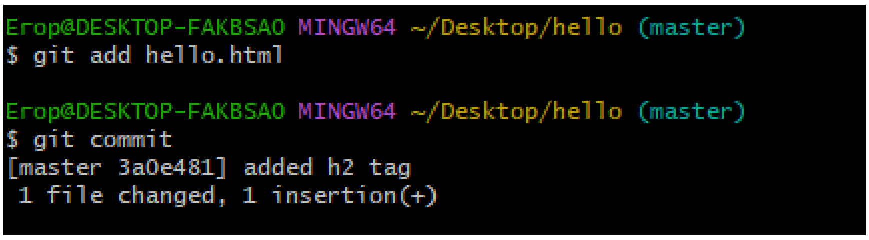
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add a.html
fatal: pathspec 'a.html' did not match any files

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add a.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git b.html
git: 'b.html' is not a git command. See 'git --help'.

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add b.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit -m "Changes for a and B"
[master d6feb6b] Changes for a and B
 3 files changed, 1 insertion(+), 1 deletion(-)
 create mode 100644 a.html
 create mode 100644 b.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add v.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit -m "Unrelated change for v"
[master d26eec8] Unrelated change for v
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 v.html
```



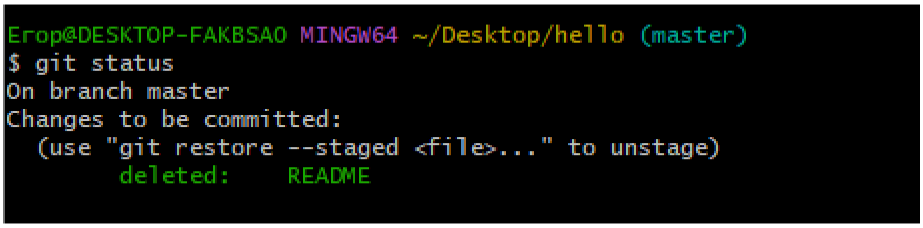
```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add a.html
fatal: pathspec 'a.html' did not match any files

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add a.html

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git b.html
git: 'b.html' is not a git command. See 'git --help'.

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add b.html

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit -m "Changes for a and B"
[master d6feb6b] Changes for a and B
 3 files changed, 1 insertion(+), 1 deletion(-)
 create mode 100644 a.html
 create mode 100644 b.html

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add v.html

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit -m "Unrelated change for v"
[master d26eec8] Unrelated change for v
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 v.html

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ |
```

7. Коммит изменений

```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit
[master 3a0e481] added h2 tag
 1 file changed, 1 insertion(+)
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    README
```



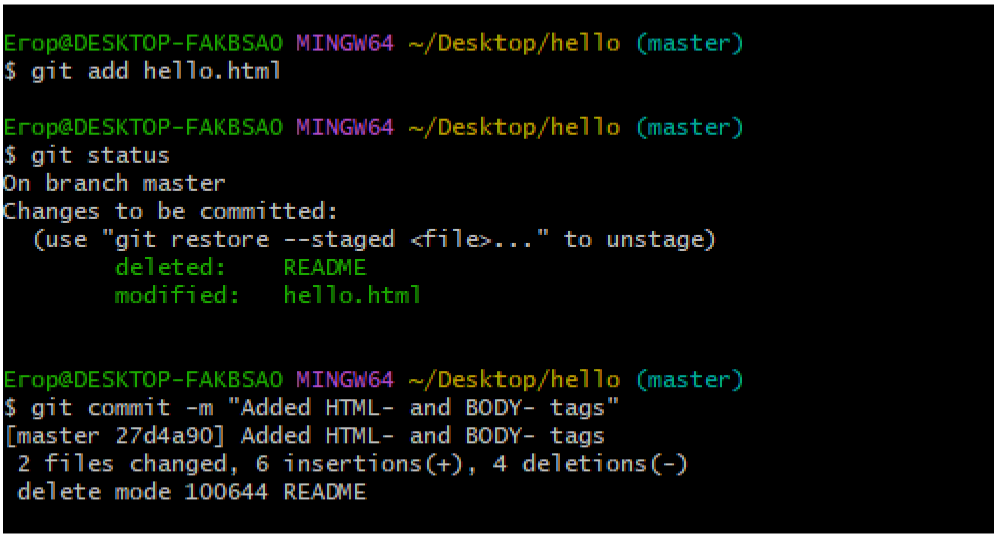
```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit
[master 3a0e481] added h2 tag
 1 file changed, 1 insertion(+)
```



```
Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    README
```

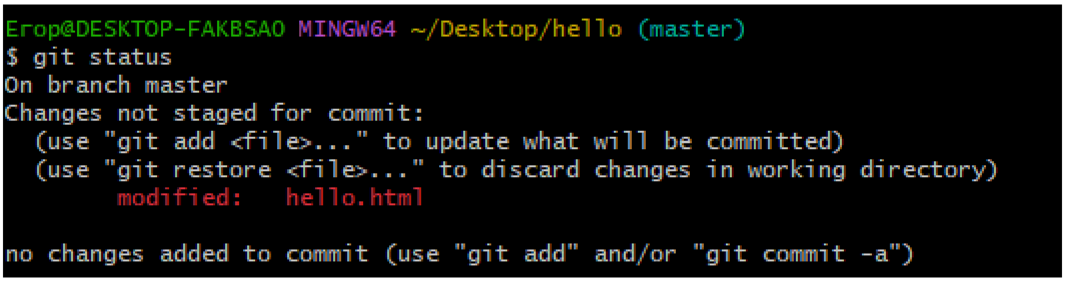
8. Изменения

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



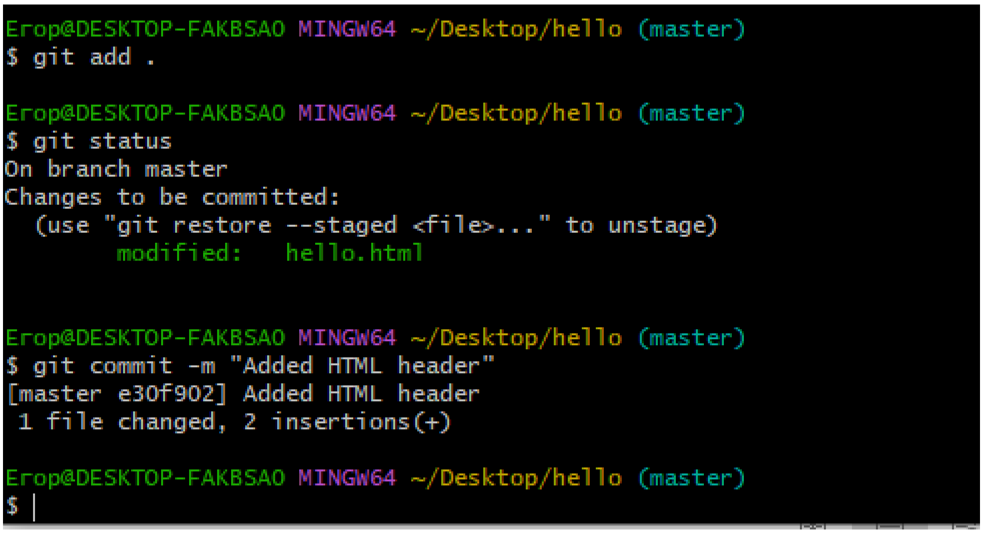
```
Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git add hello.html

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


Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git commit -m "Added HTML- and BODY- tags"
[master 27d4a90] Added HTML- and BODY- tags
 2 files changed, 6 insertions(+), 4 deletions(-)
 delete mode 100644 README
```

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



```
Erop@DESKTOP-FAKBSAO 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")
```

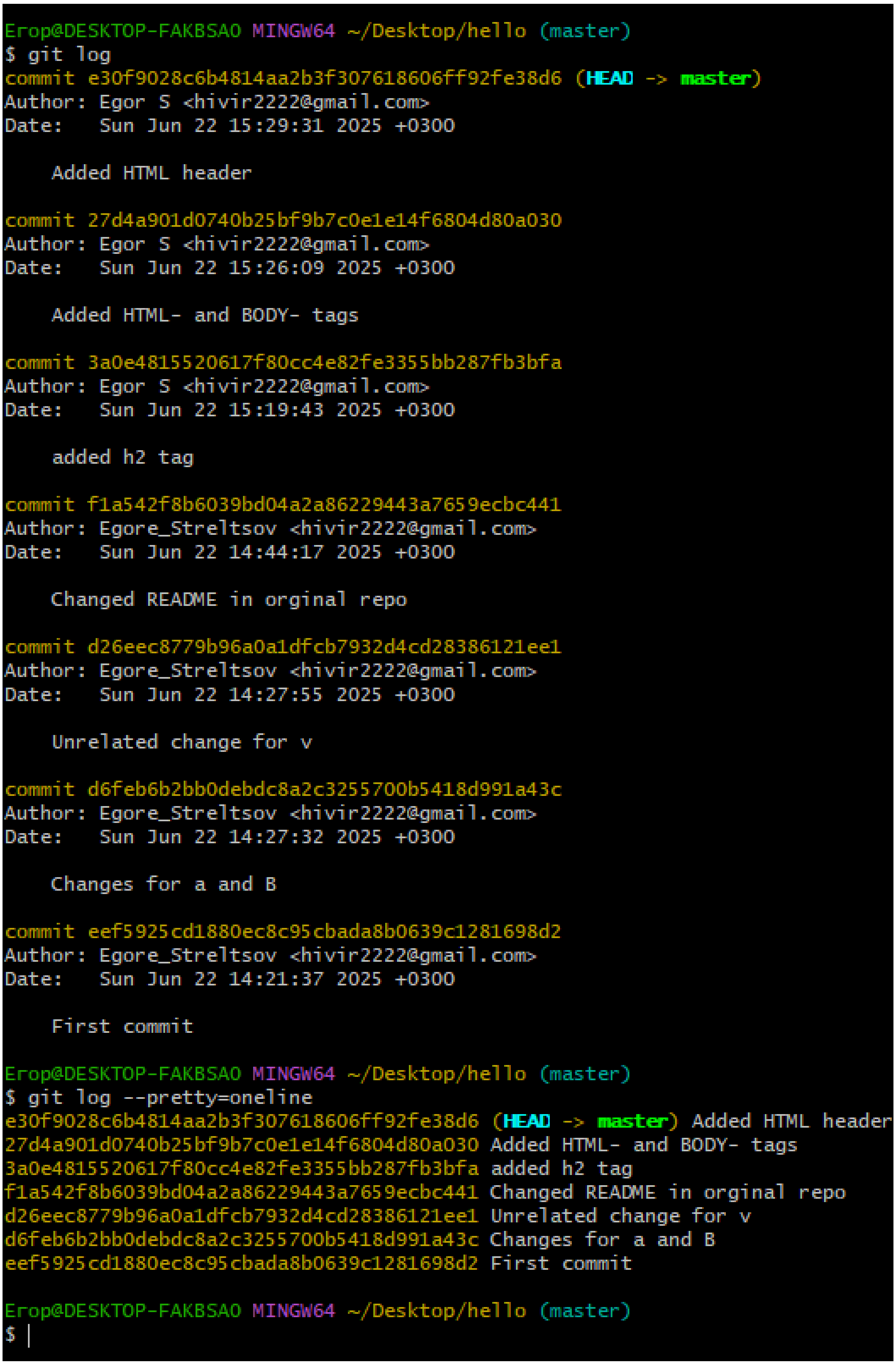
Добавление второго изменения в индекс, коммит второго изменения



```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add .

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html


Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit -m "Added HTML header"
[master e30f902] Added HTML header
 1 file changed, 2 insertions(+)

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ |
```

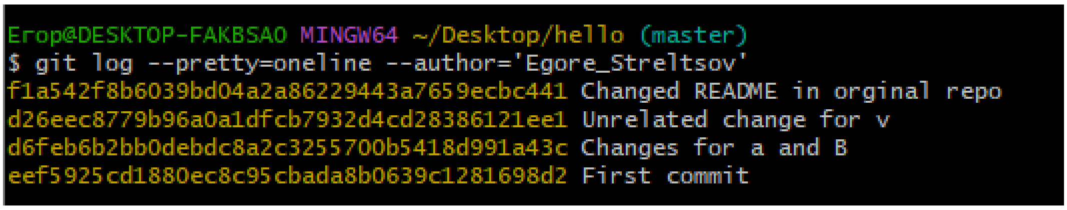
1.2 Git

Вывод списка произведенных изменений с полной информацией и в однострочном формате:



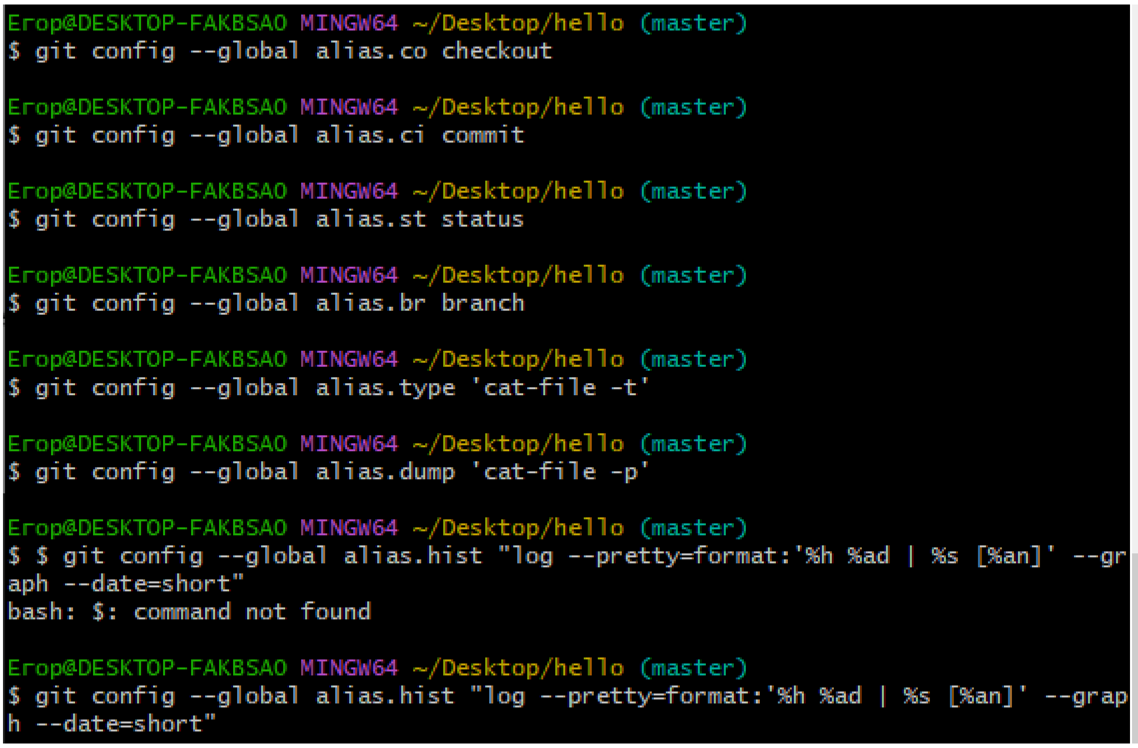
```
Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git log
commit e30f9028c6b4814aa2b3f307618606ff92fe38d6 (HEAD -> master)
Author: Egor S <hivir2222@gmail.com>
Date:   Sun Jun 22 15:29:31 2025 +0300

    Added HTML header

commit 27d4a901d0740b25bf9b7c0e1e14f6804d80a030
Author: Egor S <hivir2222@gmail.com>
Date:   Sun Jun 22 15:26:09 2025 +0300

    Added HTML- and BODY- tags

commit 3a0e4815520617f80cc4e82fe3355bb287fb3bfa
Author: Egor S <hivir2222@gmail.com>
Date:   Sun Jun 22 15:19:43 2025 +0300

    added h2 tag

commit f1a542f8b6039bd04a2a86229443a7659ecbc441
Author: Egore_Streltsov <hivir2222@gmail.com>
Date:   Sun Jun 22 14:44:17 2025 +0300

    Changed README in orginal repo

commit d26eec8779b96a0a1dfcb7932d4cd28386121ee1
Author: Egore_Streltsov <hivir2222@gmail.com>
Date:   Sun Jun 22 14:27:55 2025 +0300

    Unrelated change for v

commit d6feb6b2bb0debdc8a2c3255700b5418d991a43c
Author: Egore_Streltsov <hivir2222@gmail.com>
Date:   Sun Jun 22 14:27:32 2025 +0300

    Changes for a and B

commit eef5925cd1880ec8c95cbada8b0639c1281698d2
Author: Egore_Streltsov <hivir2222@gmail.com>
Date:   Sun Jun 22 14:21:37 2025 +0300

    First commit

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git log --pretty=oneline
e30f9028c6b4814aa2b3f307618606ff92fe38d6 (HEAD -> master) Added HTML header
27d4a901d0740b25bf9b7c0e1e14f6804d80a030 Added HTML- and BODY- tags
3a0e4815520617f80cc4e82fe3355bb287fb3bfa added h2 tag
f1a542f8b6039bd04a2a86229443a7659ecbc441 Changed README in orginal repo
d26eec8779b96a0a1dfcb7932d4cd28386121ee1 Unrelated change for v
d6feb6b2bb0debdc8a2c3255700b5418d991a43c Changes for a and B
eef5925cd1880ec8c95cbada8b0639c1281698d2 First commit

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ |
```

Вывод изменений с автором :



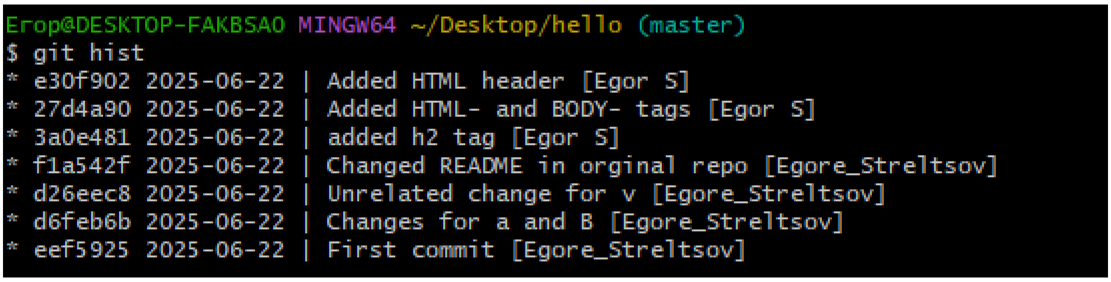
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git log --pretty=oneline --author='Egore_Streltsov'
f1a542f8b6039bd04a2a86229443a7659ecbc441 Changed README in orginal repo
d26eec8779b96a0a1dfcb7932d4cd28386121ee1 Unrelated change for v
d6feb6b2bb0debdc8a2c3255700b5418d991a43c Changes for a and B
eef5925cd1880ec8c95cbada8b0639c1281698d2 First commit
```

Настройка алиасов



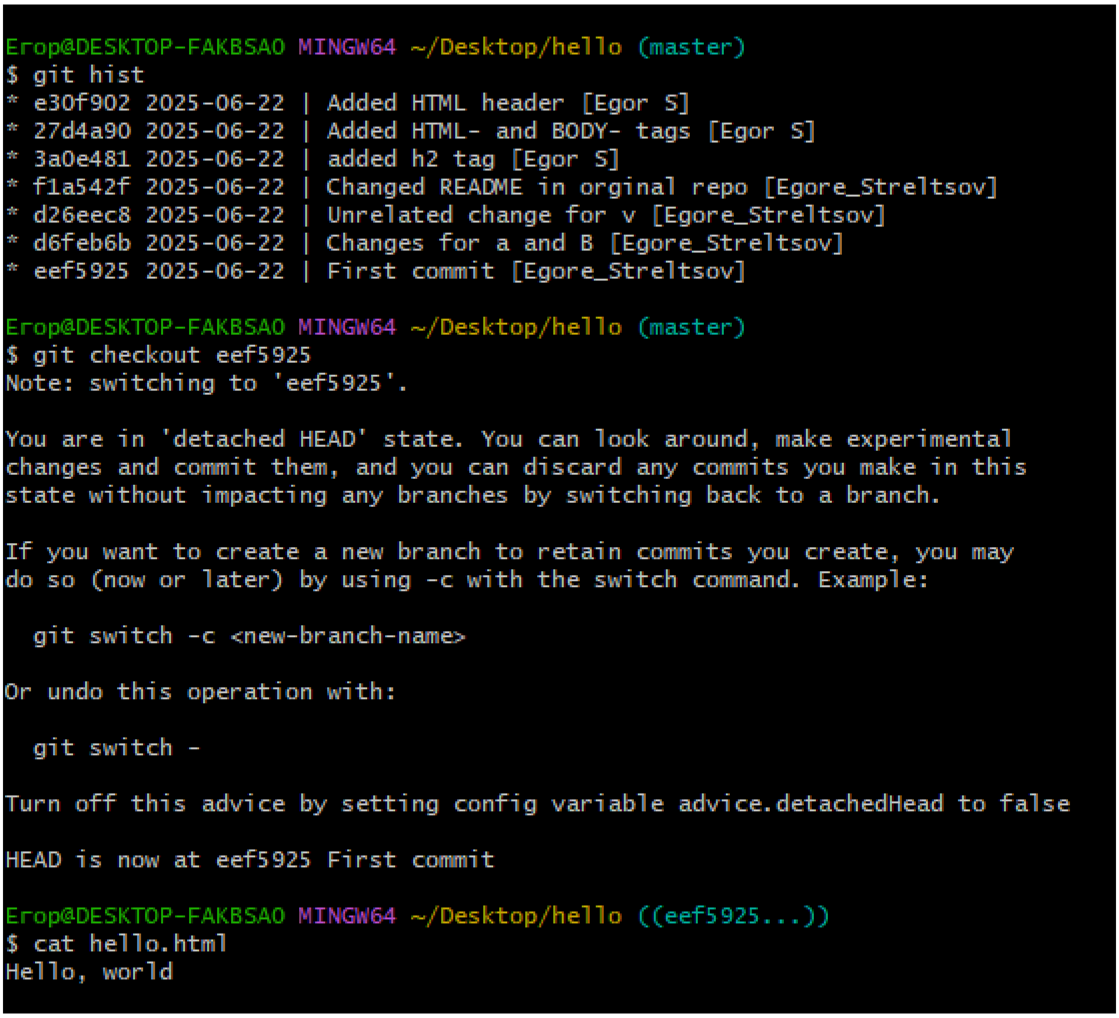
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git config --global alias.co checkout

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git config --global alias.ci commit

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git config --global alias.st status

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git config --global alias.br branch

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git config --global alias.type 'cat-file -t'

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git config --global alias.dump 'cat-file -p'

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ $ git config --global alias.hist "log --pretty=format:'%h %ad | %s [%an]' --gr
aph --date=short"
bash: $: command not found

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git config --global alias.hist "log --pretty=format:'%h %ad | %s [%an]' --grap
h --date=short"
```

Примеры выводов с использованием алиасов:



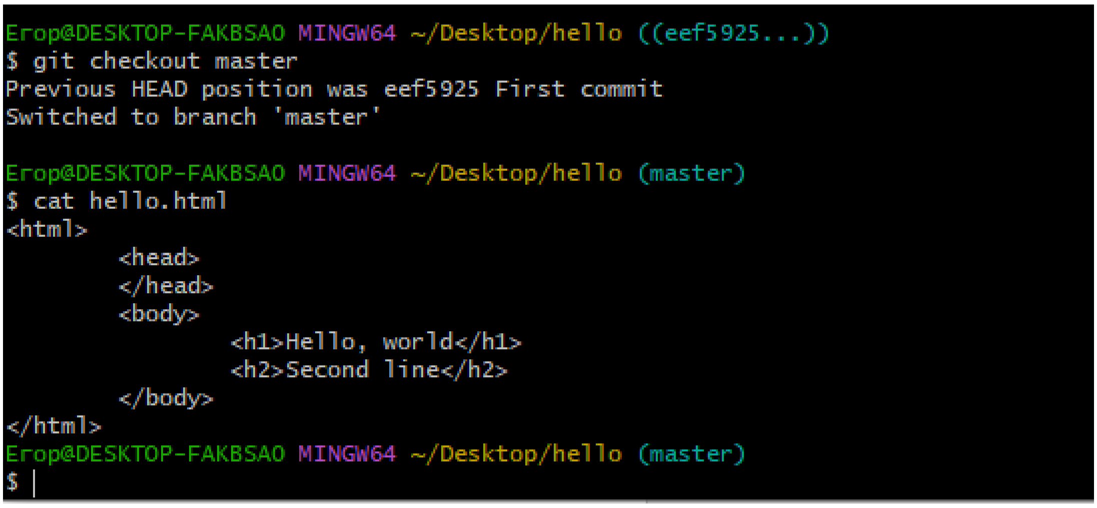
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git hist
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]
```

Далее нужно выполнить команду git checkout с номером нужного хэша



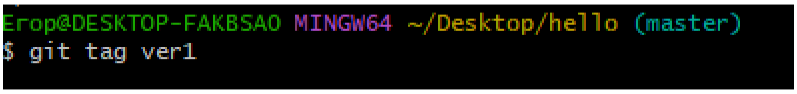
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git hist
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git checkout eef5925
Note: switching to 'eef5925'.

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 eef5925 First commit

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello ((eef5925...))
$ cat hello.html
Hello, world
```

Возвращение к последней версии в ветке master и вывод содержимого



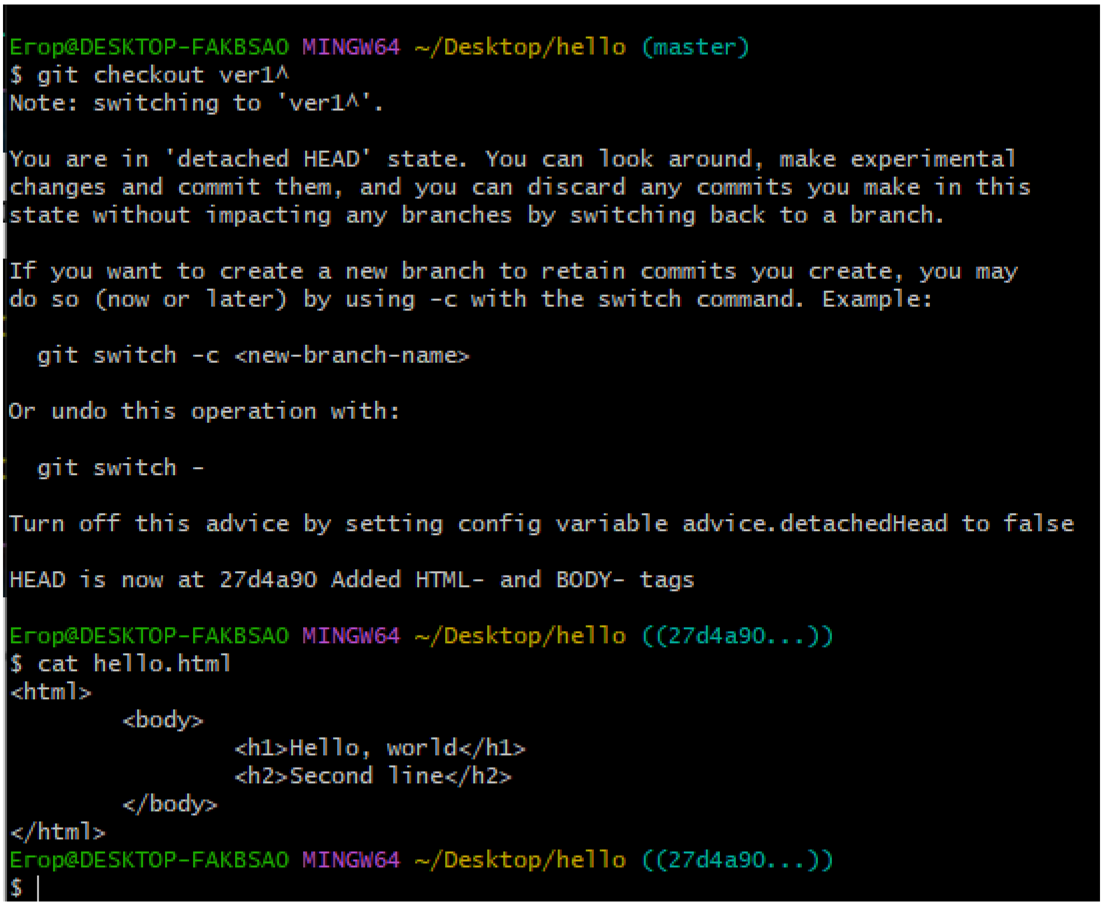
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello ((eef5925...))
$ git checkout master
Previous HEAD position was eef5925 First commit
Switched to branch 'master'

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ cat hello.html
<html>
        <head>
        </head>
        <body>
                <h1>Hello, world</h1>
                <h2>Second line</h2>
        </body>
</html>
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ |
```

Создание тегов версий



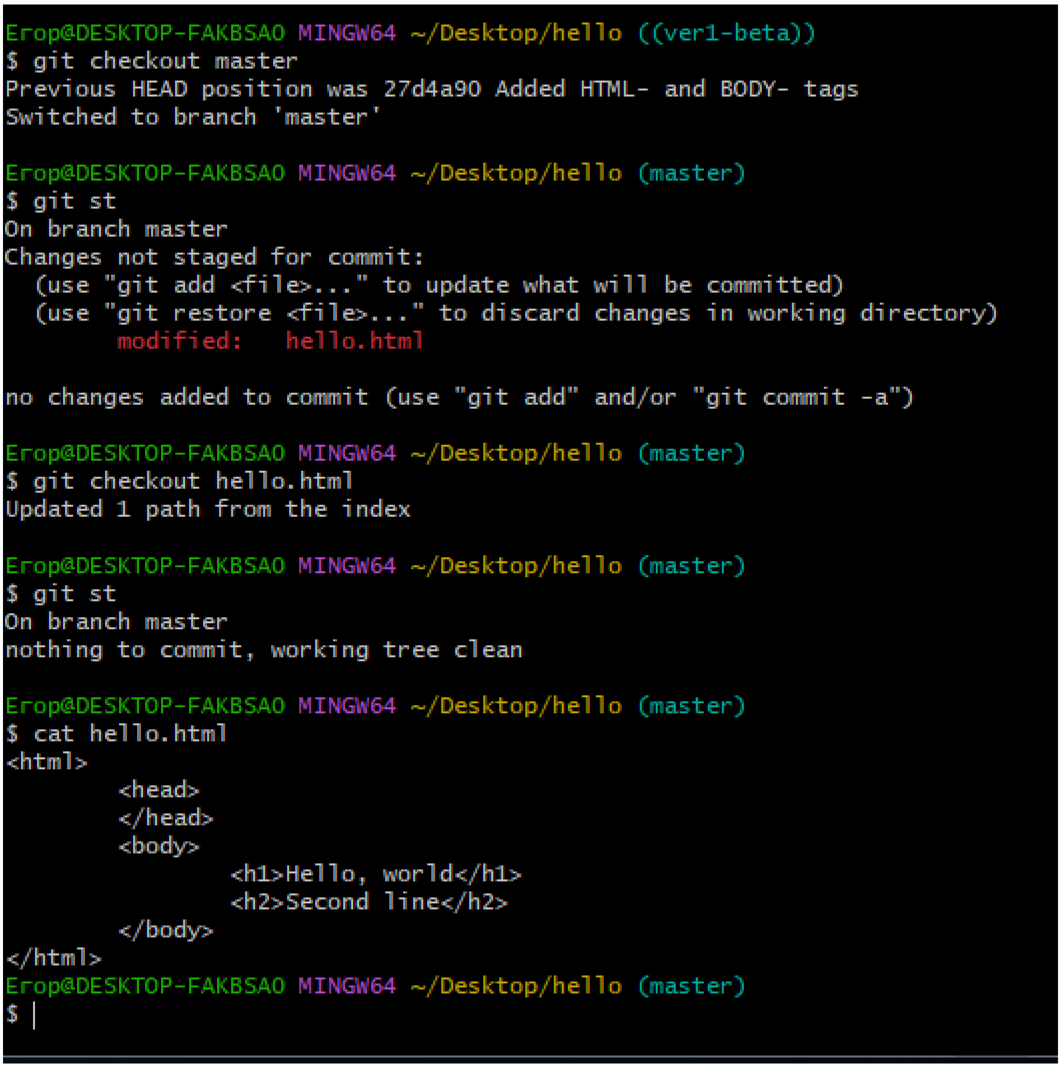
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git tag ver1
```

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



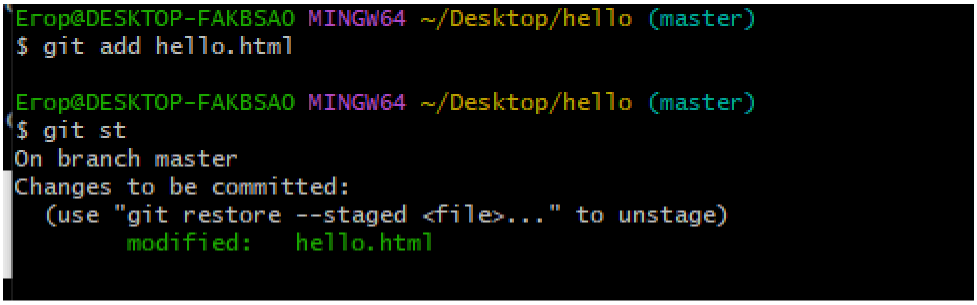
```
Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git checkout ver1^
Note: switching to 'ver1^'.

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 27d4a90 Added HTML- and BODY- tags

Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello ((27d4a90...))
$ cat hello.html
<html>
        <body>
                <h1>Hello, world</h1>
                <h2>Second line</h2>
        </body>
</html>
Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello ((27d4a90...))
$ |
```

Отмена локальных изменений до индексации



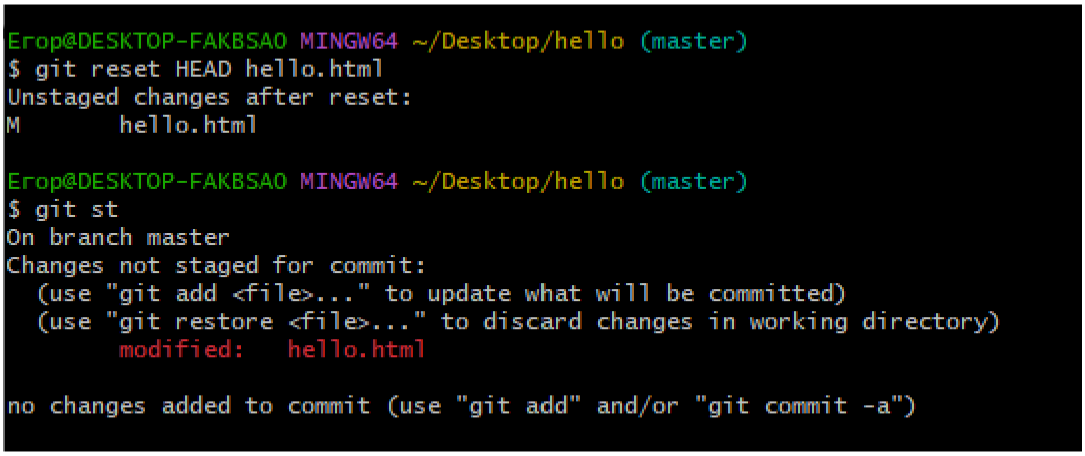
```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello ((ver1-beta))
$ git checkout master
Previous HEAD position was 27d4a90 Added HTML- and BODY- tags
Switched to branch 'master'

Erop@DESKTOP-FAKBSA0 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")

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git checkout hello.html
Updated 1 path from the index

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git st
On branch master
nothing to commit, working tree clean

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ cat hello.html
<html>
        <head>
        </head>
        <body>
                <h1>Hello, world</h1>
                <h2>Second line</h2>
        </body>
</html>
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ |
```

Отмена проиндексированных изменений



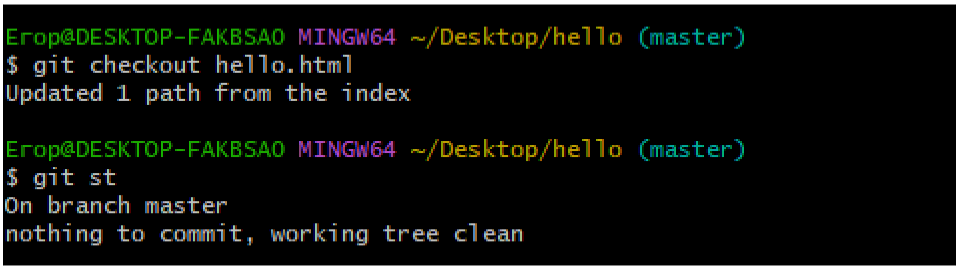
```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git st
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

Для отмены индексация изменения используется команда git reset HEAD hello.html



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

Егор@DESKTOP-FAKBSAO 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")
```

Команда reset сбрасывает буферную зону к HEAD и очищает ее от проиндексированных изменений. Но для удаления ненужного по-прежнему используется команда git checkout

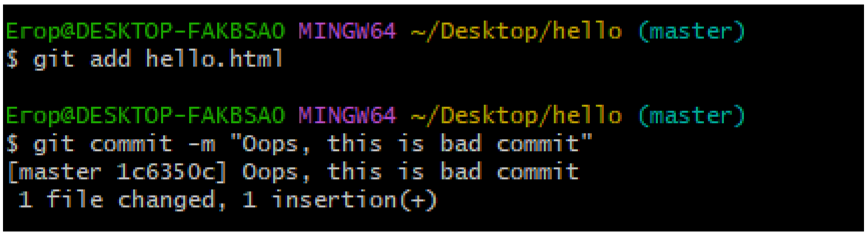


```
Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git checkout hello.html
Updated 1 path from the index

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

Отмена коммитов

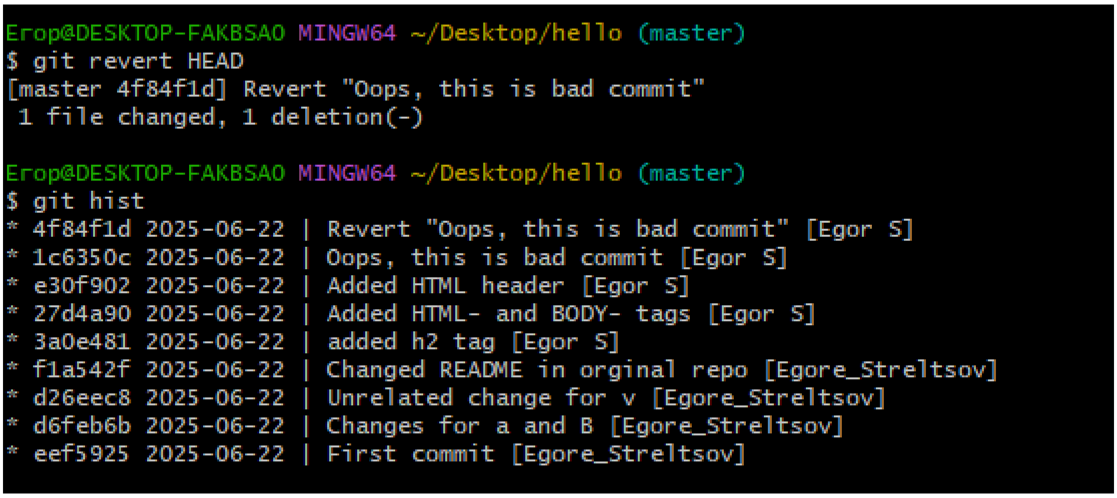
Для отмены коммита можно использовать способ создания нового коммита, отменяющего изменения.

Для начала надо внести изменение, проиндексировать его и записать коммит



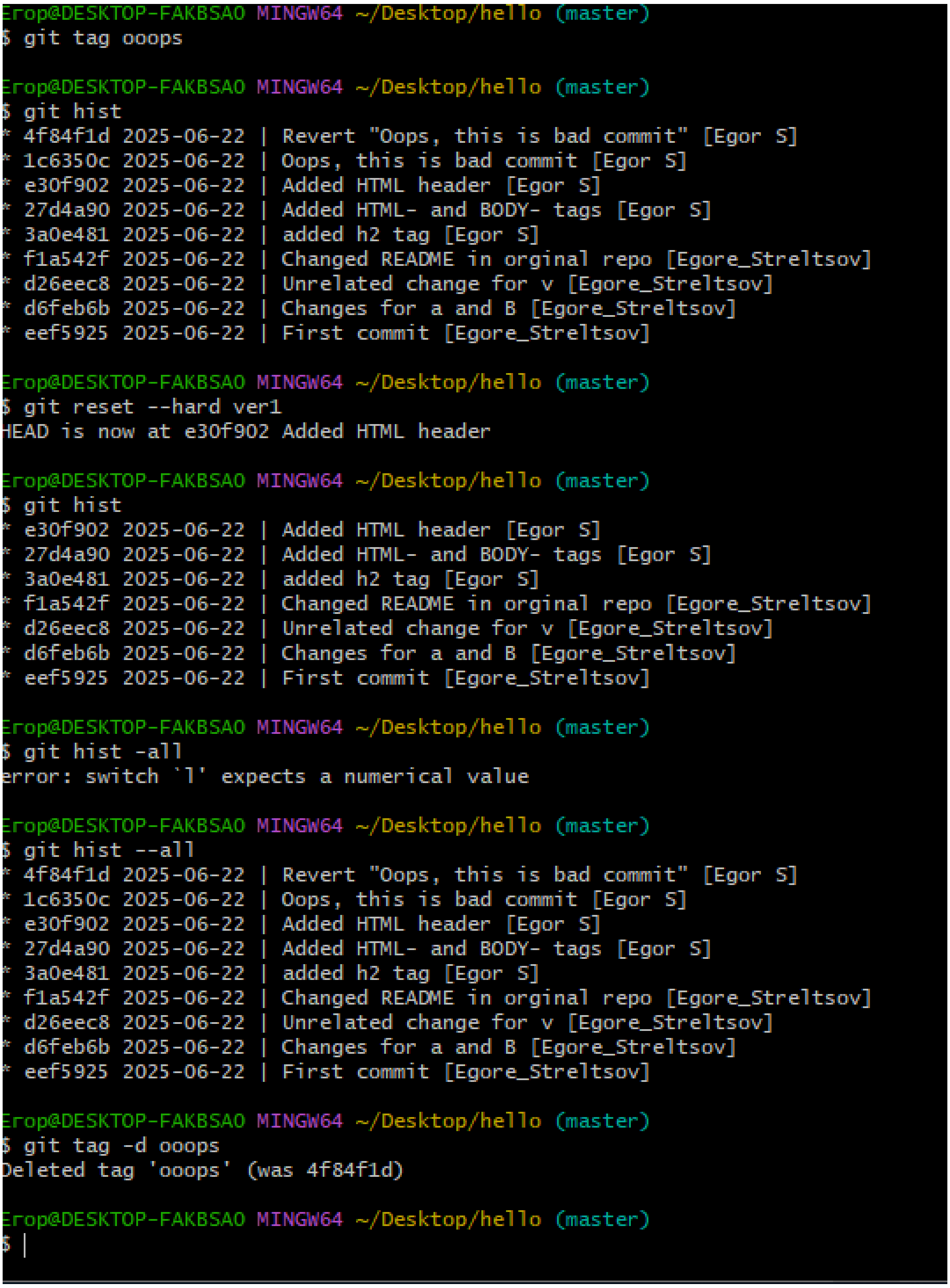
```
Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git commit -m "Oops, this is bad commit"
[master 1c6350c] Oops, this is bad commit
 1 file changed, 1 insertion(+)
```

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



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git revert HEAD
[master 4f84f1d] Revert "Oops, this is bad commit"
 1 file changed, 1 deletion(-)

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git hist
* 4f84f1d 2025-06-22 | Revert "Oops, this is bad commit" [Egor S]
* 1c6350c 2025-06-22 | Oops, this is bad commit [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]
```

Создание, а затем удаление тега “ooops”

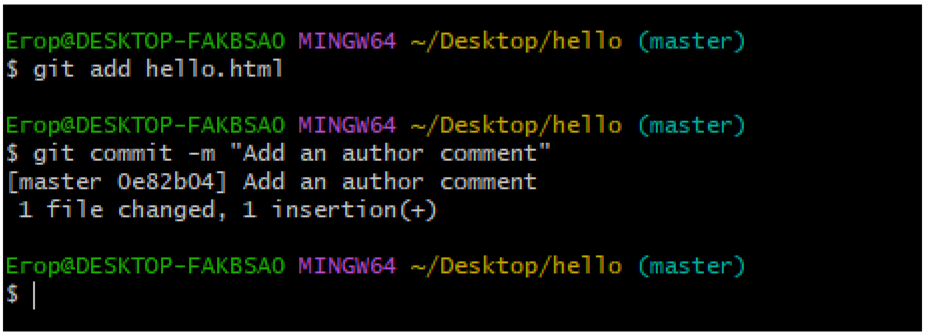


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git tag ooops

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git hist
* 4f84f1d 2025-06-22 | Revert "Oops, this is bad commit" [Egor S]
* 1c6350c 2025-06-22 | Oops, this is bad commit [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git reset --hard ver1
HEAD is now at e30f902 Added HTML header

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git hist
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git hist -all
error: switch `l' expects a numerical value

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git hist --all
* 4f84f1d 2025-06-22 | Revert "Oops, this is bad commit" [Egor S]
* 1c6350c 2025-06-22 | Oops, this is bad commit [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git tag -d ooops
Deleted tag 'ooops' (was 4f84f1d)

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ |
```

Git 1.3

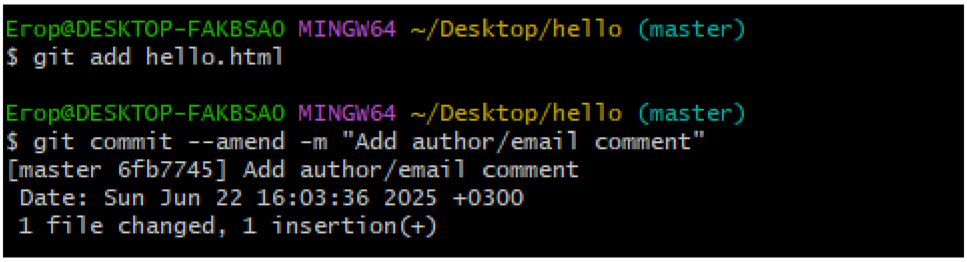
Внесение изменений в коммиты

Для начала будет создан коммит, в который позже будут внесены изменения. Добавление комментария в файл hello.html и его индексация и коммит.



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit -m "Add an author comment"
[master 0e82b04] Add an author comment
 1 file changed, 1 insertion(+)

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ |
```

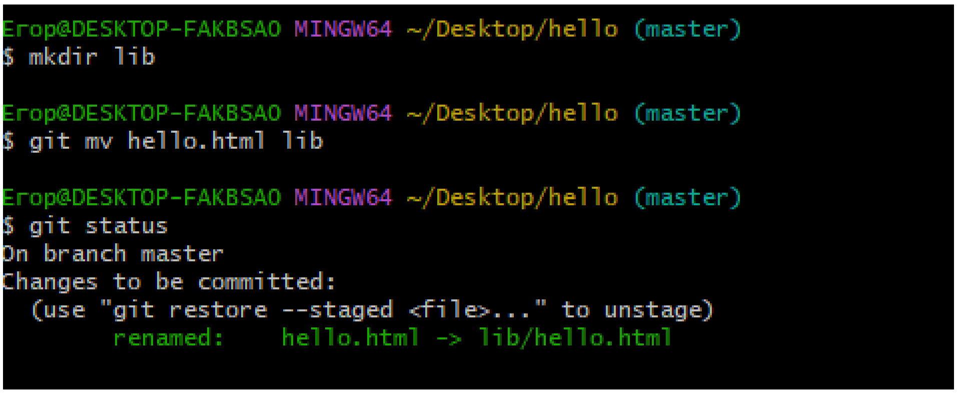
Добавление почты в коммент, и изменения коммита с указанием добавления почты.



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit --amend -m "Add author/email comment"
[master 6fb7745] Add author/email comment
 Date: Sun Jun 22 16:03:36 2025 +0300
 1 file changed, 1 insertion(+)
```

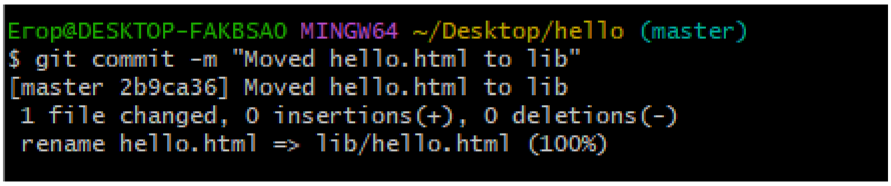
Перемещение файлов

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



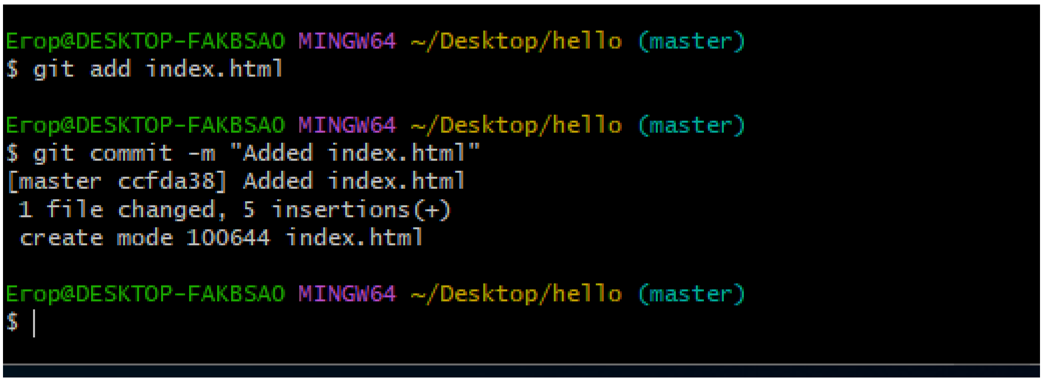
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ mkdir lib

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git mv hello.html lib

Егор@DESKTOP-FAKBSA0 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
```

Коммит данного действия

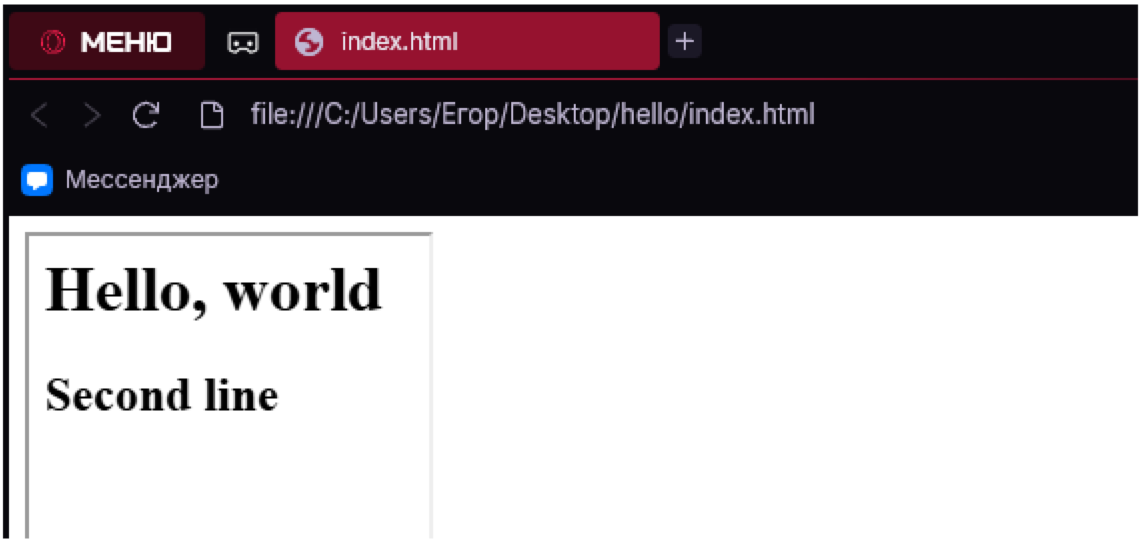


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit -m "Moved hello.html to lib"
[master 2b9ca36] Moved hello.html to lib
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename hello.html => lib/hello.html (100%)
```

Создание нового файла – index.html, с внесением в него изменений, индексация и коммит



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add index.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit -m "Added index.html"
[master ccfda38] Added index.html
 1 file changed, 5 insertions(+)
 create mode 100644 index.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ |
```

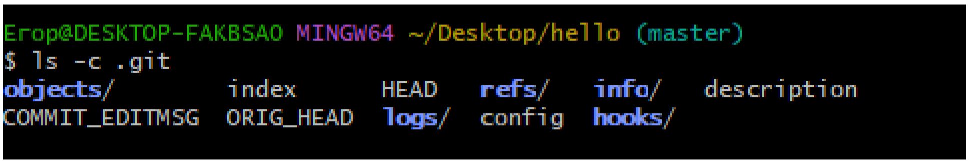


Hello, world

Second line

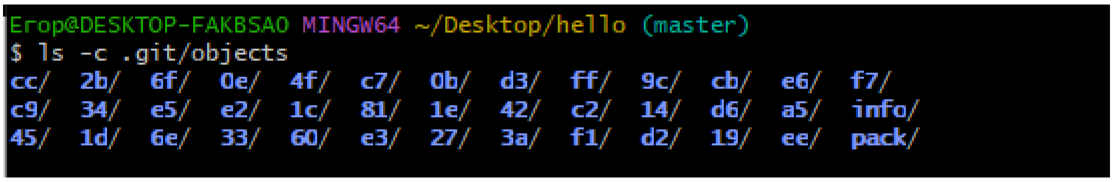
Каталог .git

Для просмотра каталога .git необходимо выполнить след. Команду



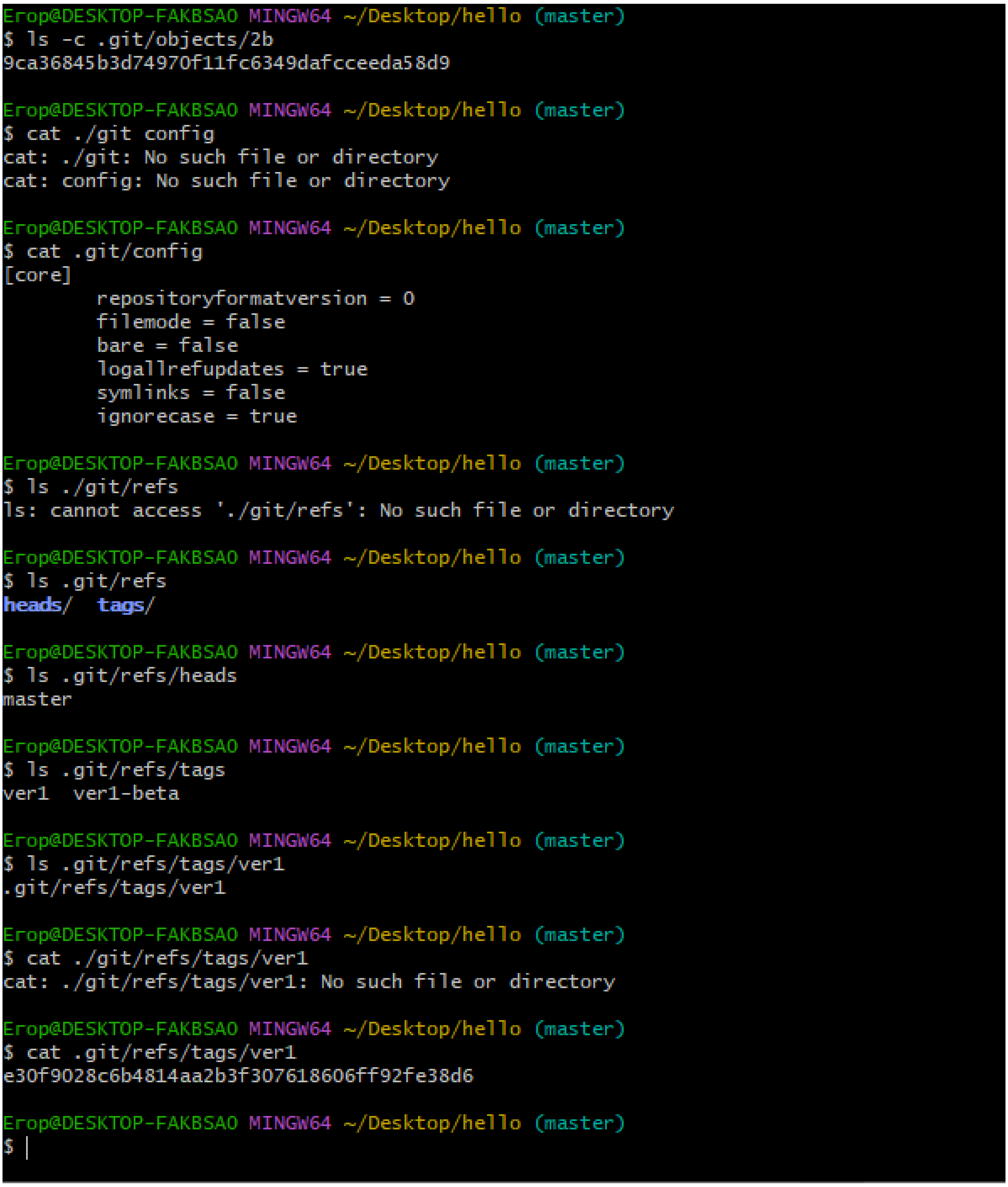
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ ls -c .git
objects/       index      HEAD      refs/   info/   description
COMMIT_EDITMSG ORIG_HEAD  logs/     config  hooks/
```

Просмотр каталога objects



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ ls -c .git/objects
cc/  2b/  6f/  0e/  4f/  c7/  0b/  d3/  ff/  9c/  cb/  e6/  f7/
c9/  34/  e5/  e2/  1c/  81/  1e/  42/  c2/  14/  d6/  a5/  info/
45/  1d/  6e/  33/  60/  e3/  27/  3a/  f1/  d2/  19/  ee/  pack/
```

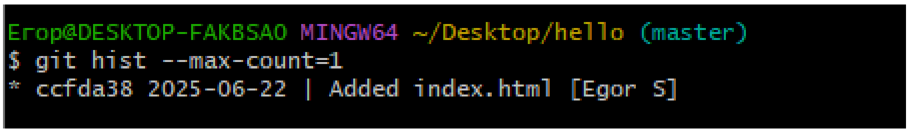
Просмотр различных файлов в подкаталоге tags, веток в каталоге heads.



```
Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ ls -c .git/objects/2b
9ca36845b3d74970f11fc6349dafcceeda58d9

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ cat ./git config
cat: ./git: No such file or directory
cat: config: No such file or directory

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

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ ls ./git/refs
ls: cannot access './git/refs': No such file or directory

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ ls .git/refs
heads/  tags/

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ ls .git/refs/heads
master

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

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ ls .git/refs/tags/ver1
.git/refs/tags/ver1

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ cat ./git/refs/tags/ver1
cat: ./git/refs/tags/ver1: No such file or directory

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ cat .git/refs/tags/ver1
e30f9028c6b4814aa2b3f307618606ff92fe38d6

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ |
```

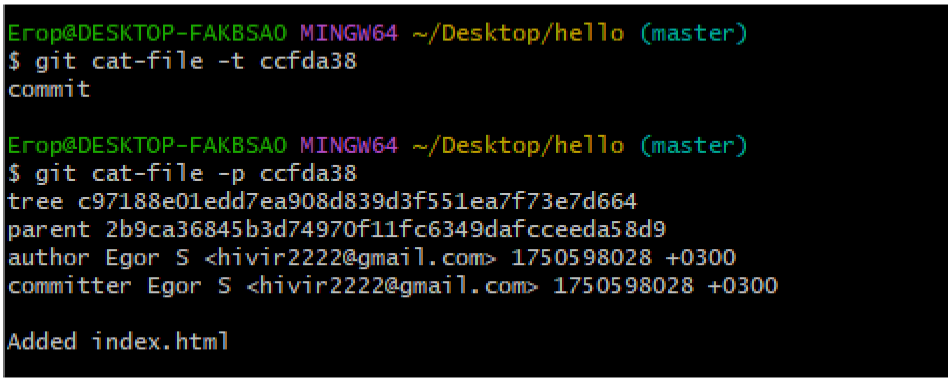
Работа с обьектами git

Просмотр последнего коммита

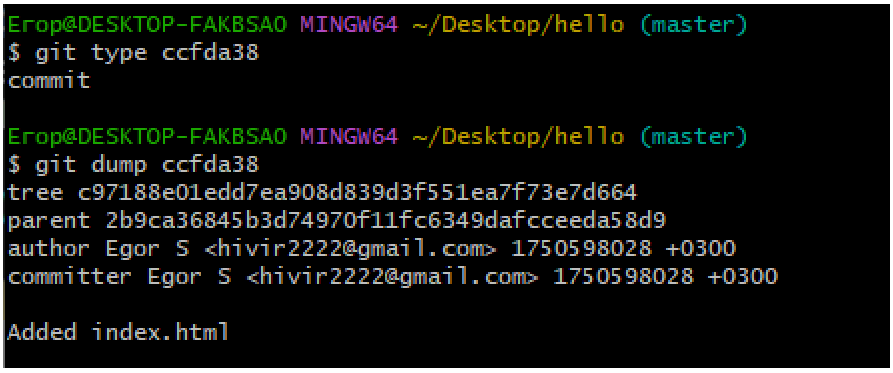


```
Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git hist --max-count=1
* ccfda38 2025-06-22 | Added index.html [Egor S]
```

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

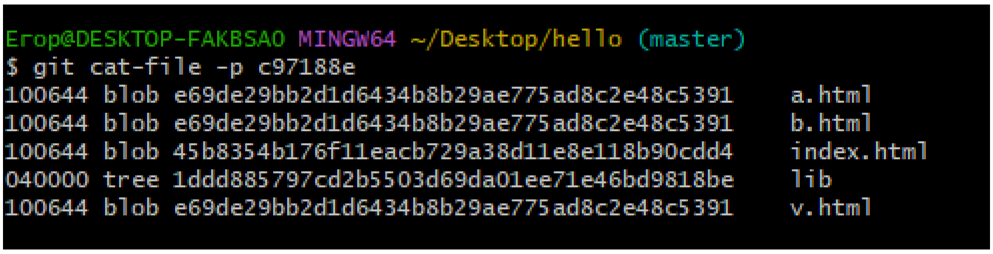


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git cat-file -t ccfda38
commit

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git cat-file -p ccfda38
tree c97188e01edd7ea908d839d3f551ea7f73e7d664
parent 2b9ca36845b3d74970f11fc6349dafcceeda58d9
author Egor S <hivir2222@gmail.com> 1750598028 +0300
committer Egor S <hivir2222@gmail.com> 1750598028 +0300

Added index.html
```



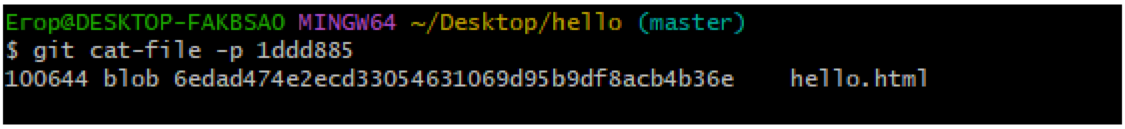
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git type ccfda38
commit

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git dump ccfda38
tree c97188e01edd7ea908d839d3f551ea7f73e7d664
parent 2b9ca36845b3d74970f11fc6349dafcceeda58d9
author Egor S <hivir2222@gmail.com> 1750598028 +0300
committer Egor S <hivir2222@gmail.com> 1750598028 +0300

Added index.html
```

Просмотр дерева каталогов с помощью его хэша



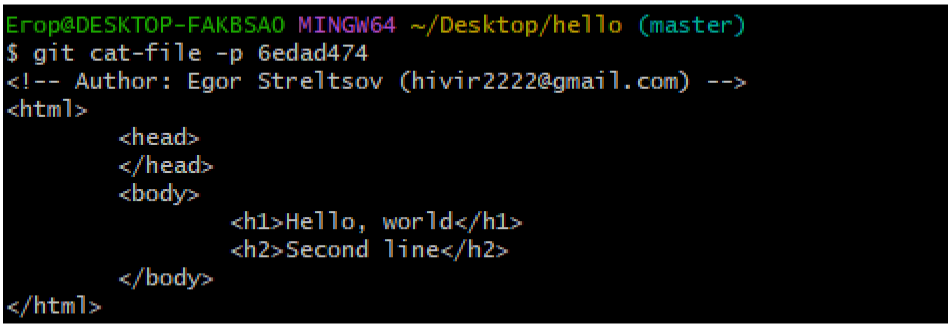
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git cat-file -p c97188e
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391    a.html
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391    b.html
100644 blob 45b8354b176f11eacb729a38d11e8e118b90cdd4    index.html
040000 tree 1ddd885797cd2b5503d69da01ee71e46bd9818be    lib
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391    v.html
```

Просмотр каталога lib



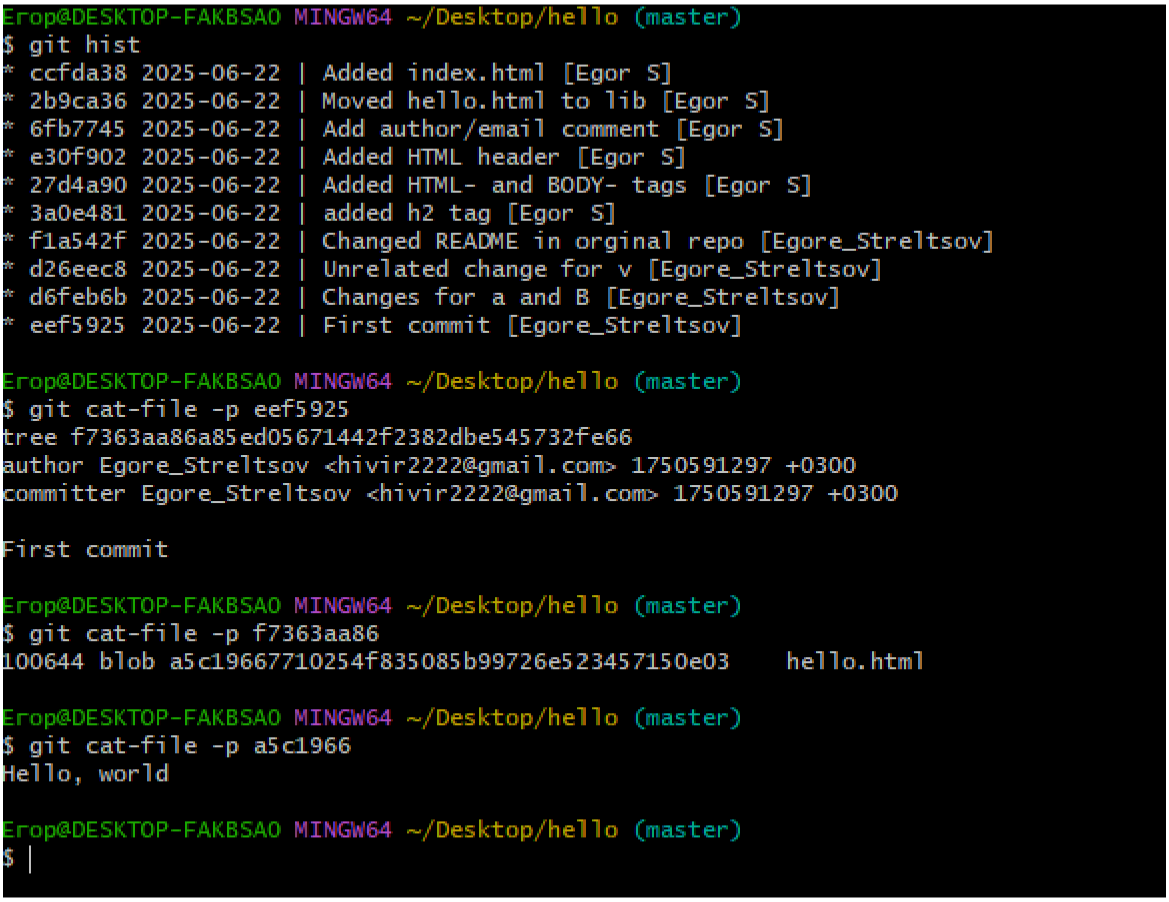
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git cat-file -p 1ddd885
100644 blob 6edad474e2ecd33054631069d95b9df8acb4b36e    hello.html
```

Просмотр содержимого файла hello.html



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git cat-file -p 6edad474
<!-- Author: Egor Streltsov (hivir2222@gmail.com) -->
<html>
        <head>
        </head>
        <body>
                <h1>Hello, world</h1>
                <h2>Second line</h2>
        </body>
</html>
```

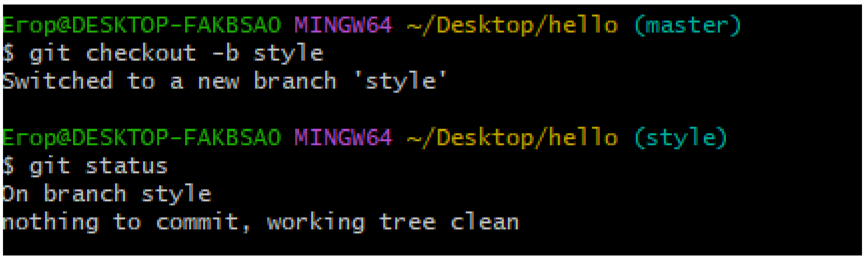
Аналогичным образом можно просмотреть содержимое файла, каким оно было в самом первом коммите. Для этого требуется использовать лишь нужные хэши.



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git hist
* ccfda38 2025-06-22 | Added index.html [Egor S]
* 2b9ca36 2025-06-22 | Moved hello.html to lib [Egor S]
* 6fb7745 2025-06-22 | Add author/email comment [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git cat-file -p eef5925
tree f7363aa86a85ed05671442f2382dbe545732fe66
author Egore_Streltsov <hivir2222@gmail.com> 1750591297 +0300
committer Egore_Streltsov <hivir2222@gmail.com> 1750591297 +0300

First commit

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git cat-file -p f7363aa86
100644 blob a5c19667710254f835085b99726e523457150e03    hello.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git cat-file -p a5c1966
Hello, world

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ |
```

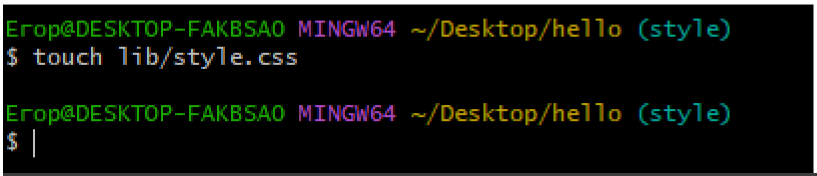
Git 1.4

Создание ветки style

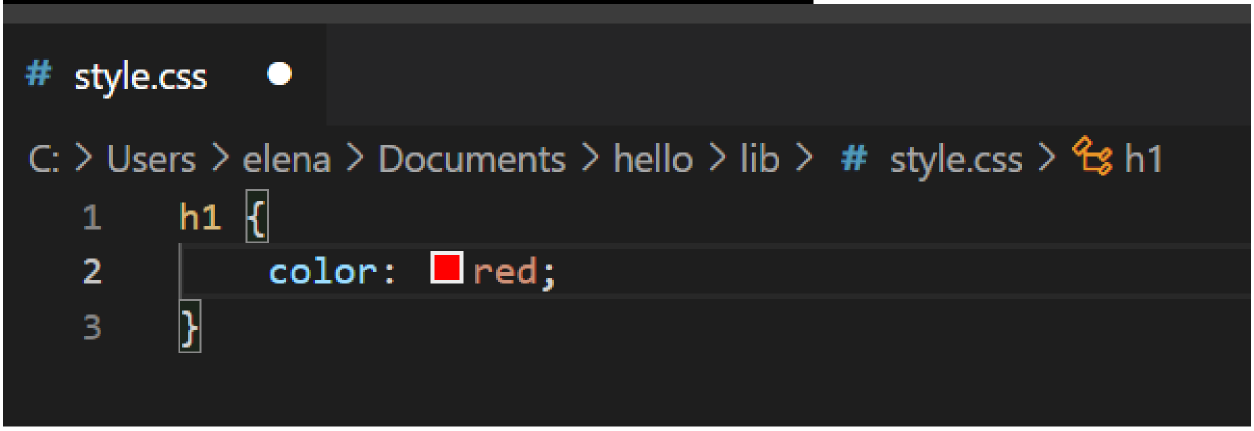


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git checkout -b style
Switched to a new branch 'style'

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ git status
On branch style
nothing to commit, working tree clean
```

Затем нужно создать файл стилей и внести в него код



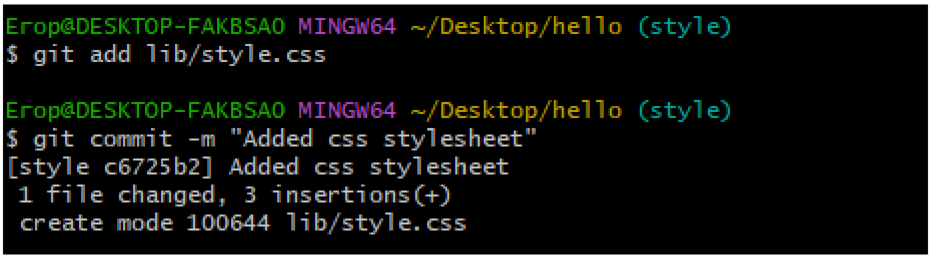
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ touch lib/style.css

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ |
```



# style.css ●
C: > Users > elena > Documents > hello > lib > # style.css > h1

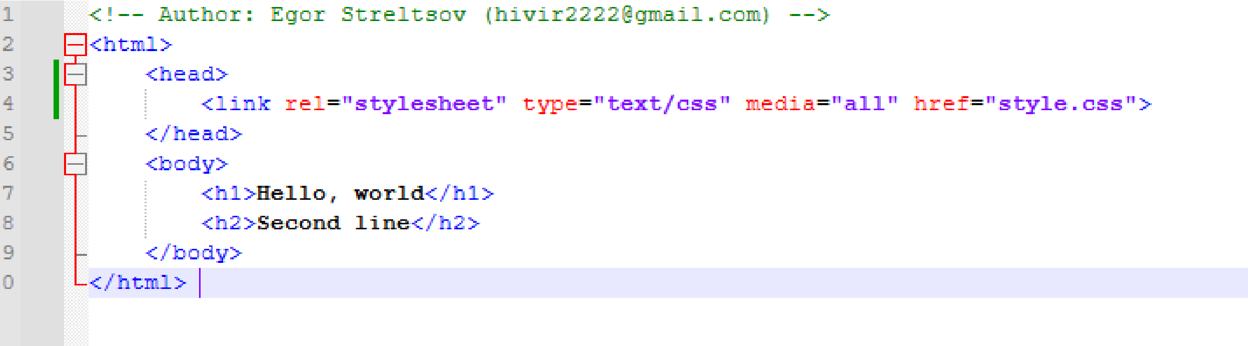
```
h1 {
    color: red;
}
```

После этого надо произвести индексацию и коммит

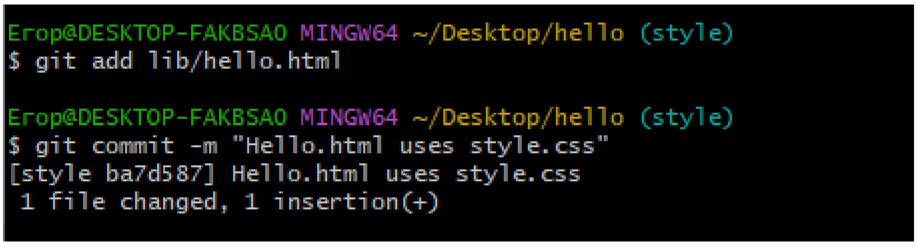


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ git add lib/style.css

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ git commit -m "Added css stylesheet"
[style c6725b2] Added css stylesheet
 1 file changed, 3 insertions(+)
 create mode 100644 lib/style.css
```

Далее требуется изменить основную страницу hello.html и закоммитить изменения

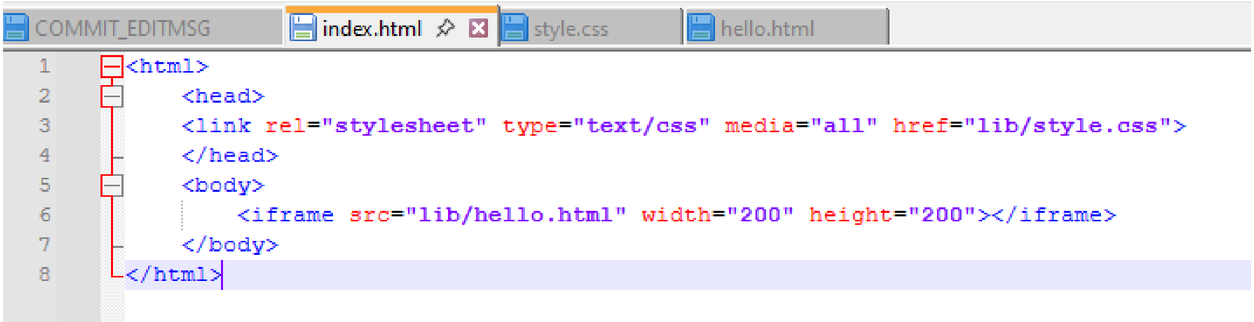


```
<!-- Author: Egor Streltsov (hivir2222@gmail.com) -->
<html>
    <head>
        <link rel="stylesheet" type="text/css" media="all" href="style.css">
    </head>
    <body>
        <h1>Hello, world</h1>
        <h2>Second line</h2>
    </body>
</html>
```

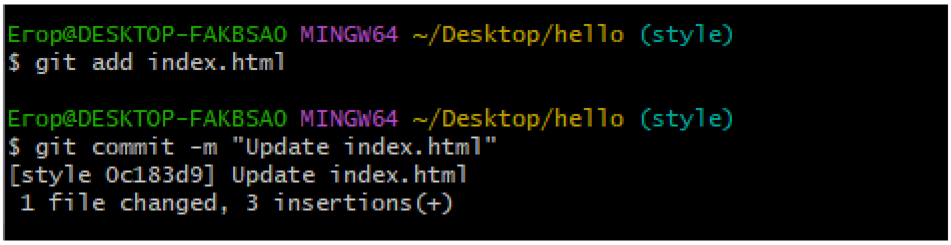


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ git add lib/hello.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ git commit -m "Hello.html uses style.css"
[style ba7d587] Hello.html uses style.css
 1 file changed, 1 insertion(+)
```

Далее аналогичные действия нужно осуществить с файлом index.html



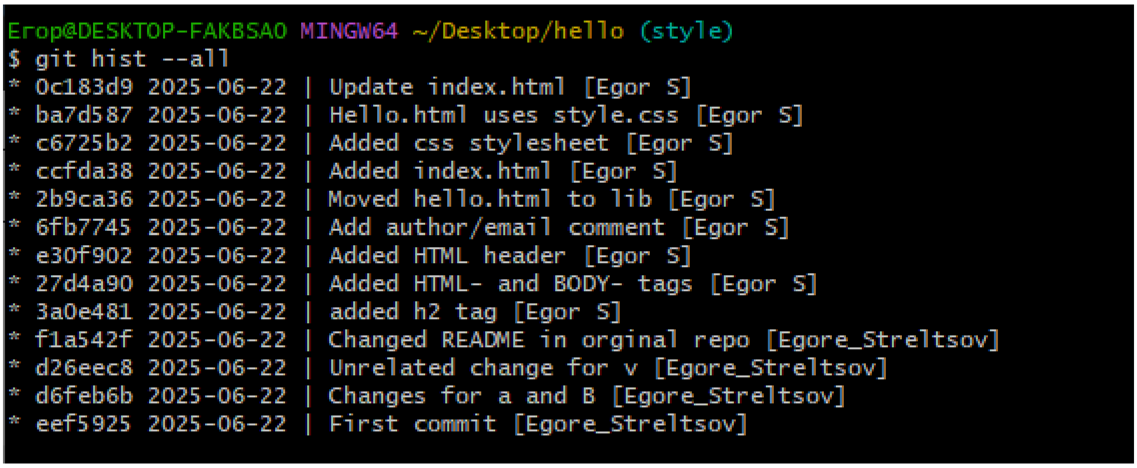
```
<html>
    <head>
    <link rel="stylesheet" type="text/css" media="all" href="lib/style.css">
    </head>
    <body>
        <iframe src="lib/hello.html" width="200" height="200"></iframe>
    </body>
</html>
```



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ git add index.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ git commit -m "Update index.html"
[style 0c183d9] Update index.html
 1 file changed, 3 insertions(+)
```

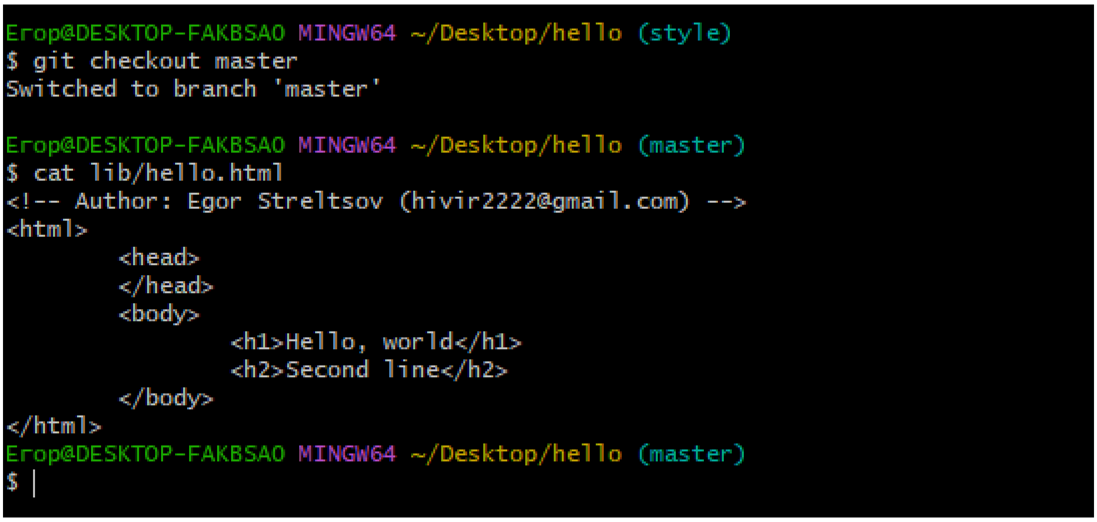
Навигация по веткам

При просмотре истории можно увидеть, что теперь в проекте 2 ветки.



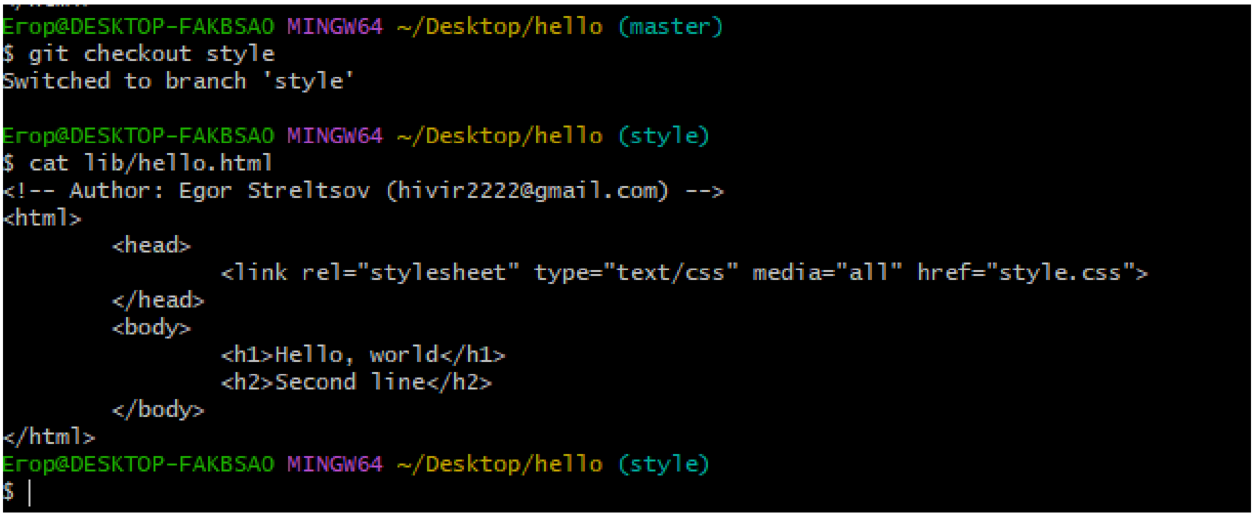
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ git hist --all
* 0c183d9 2025-06-22 | Update index.html [Egor S]
* ba7d587 2025-06-22 | Hello.html uses style.css [Egor S]
* c6725b2 2025-06-22 | Added css stylesheet [Egor S]
* ccfda38 2025-06-22 | Added index.html [Egor S]
* 2b9ca36 2025-06-22 | Moved hello.html to lib [Egor S]
* 6fb7745 2025-06-22 | Add author/email comment [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]
```

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



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

Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ cat lib/hello.html
<!-- Author: Egor Streltsov (hivir2222@gmail.com) -->
<html>
        <head>
        </head>
        <body>
                <h1>Hello, world</h1>
                <h2>Second line</h2>
        </body>
</html>
Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ |
```

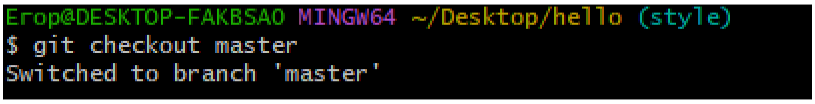
При переключении на ветку style файл hello.html будет иметь другое содержание



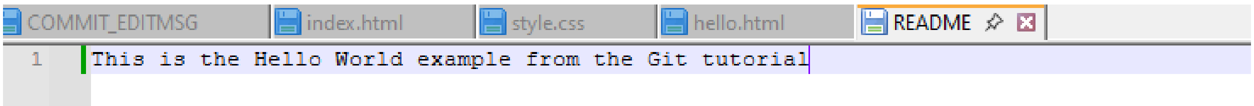
```
Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git checkout style
Switched to branch 'style'

Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (style)
$ cat lib/hello.html
<!-- Author: Egor Streltsov (hivir2222@gmail.com) -->
<html>
        <head>
                <link rel="stylesheet" type="text/css" media="all" href="style.css">
        </head>
        <body>
                <h1>Hello, world</h1>
                <h2>Second line</h2>
        </body>
</html>
Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (style)
$ |
```

Изменение в ветке master

Необходимо переключиться на ветку master и добавить файл README

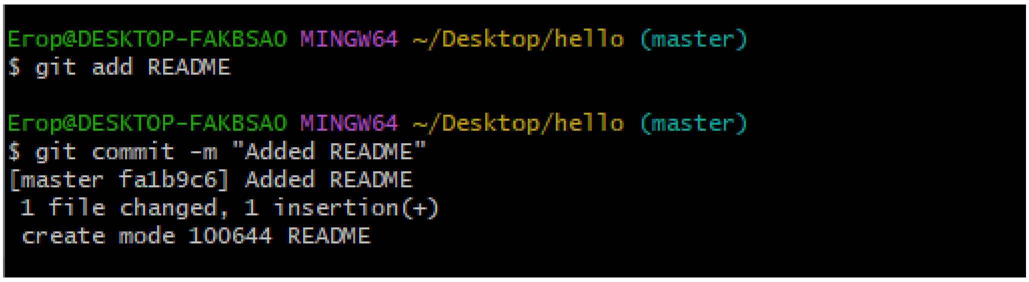


```
Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (style)
$ git checkout master
Switched to branch 'master'
```



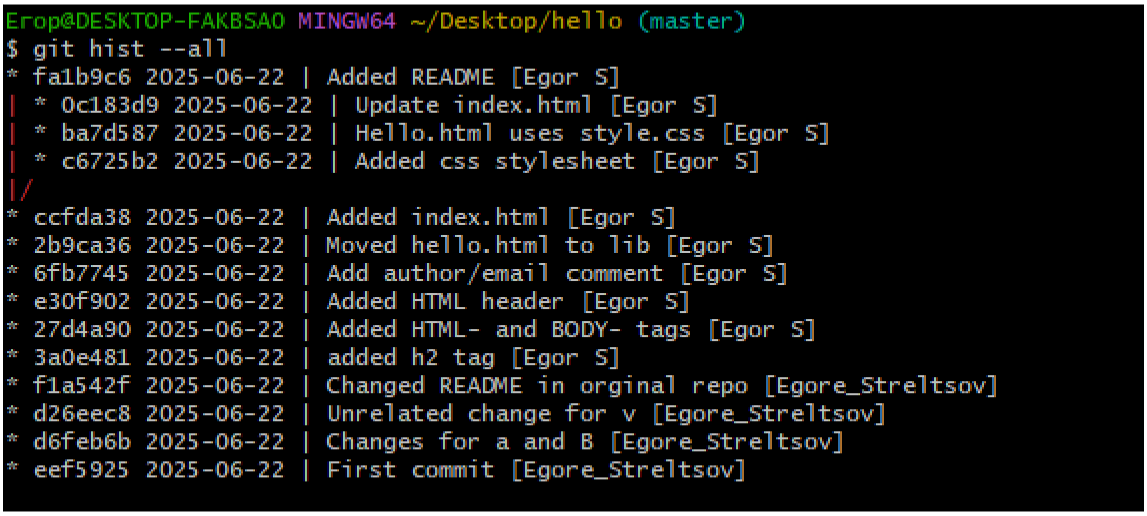
```
This is the Hello World example from the Git tutorial
```

Индексация и коммит



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

Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git commit -m "Added README"
[master fa1b9c6] Added README
 1 file changed, 1 insertion(+)
 create mode 100644 README
```

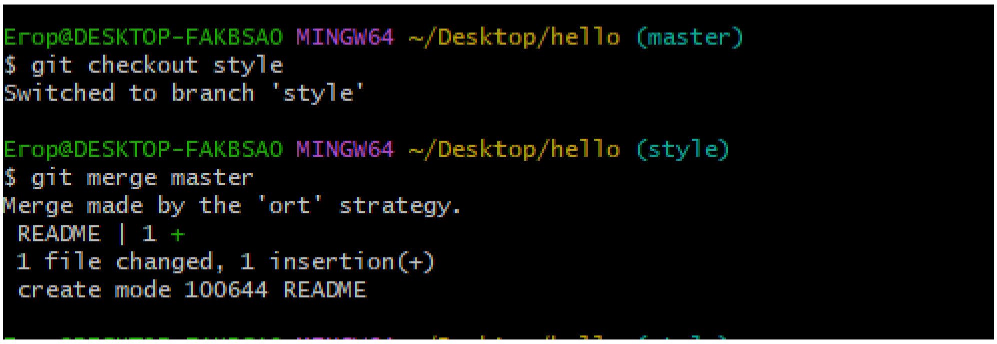
Просмотр отличающихся веток



```
Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git hist --all
* fa1b9c6 2025-06-22 | Added README [Egor S]
| * 0c183d9 2025-06-22 | Update index.html [Egor S]
| * ba7d587 2025-06-22 | Hello.html uses style.css [Egor S]
| * c6725b2 2025-06-22 | Added css stylesheet [Egor S]
|/
* ccfda38 2025-06-22 | Added index.html [Egor S]
* 2b9ca36 2025-06-22 | Moved hello.html to lib [Egor S]
* 6fb7745 2025-06-22 | Add author/email comment [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]
```

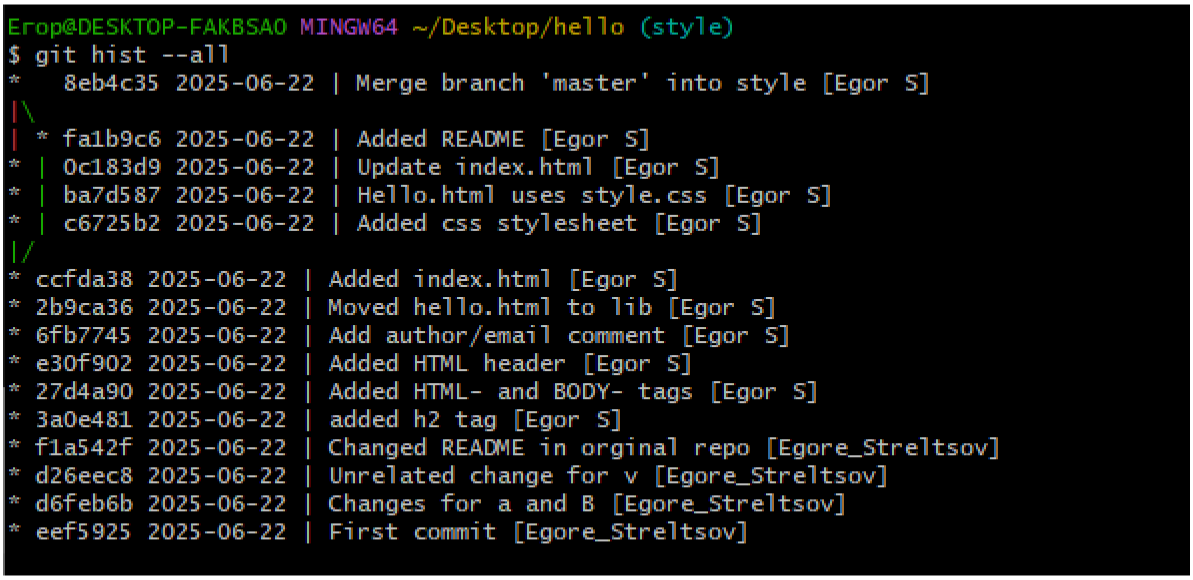
Слияние

Слияние переносит изменения из двух веток в одну. Для слияния нужно перейти на ветку style и с помощью команды git merge master совместить ветки



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

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

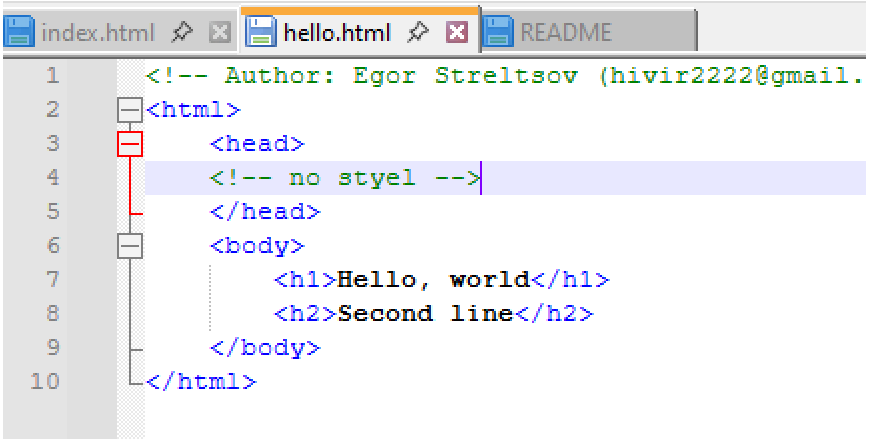
Просмотр истории



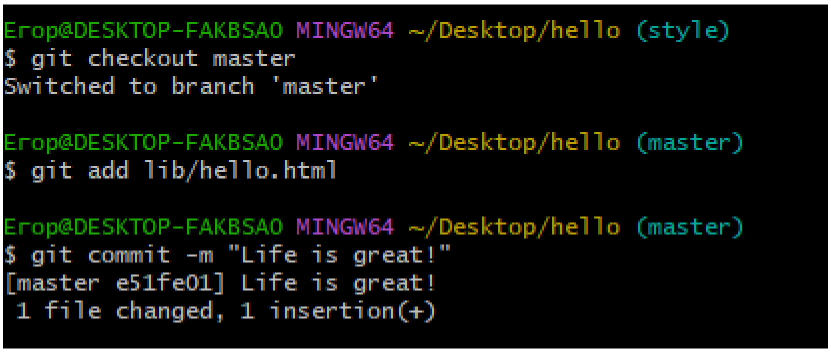
```
Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (style)
$ git hist --all
*   8eb4c35 2025-06-22 | Merge branch 'master' into style [Egor S]
|\
| * fa1b9c6 2025-06-22 | Added README [Egor S]
* | 0c183d9 2025-06-22 | Update index.html [Egor S]
* | ba7d587 2025-06-22 | Hello.html uses style.css [Egor S]
* | c6725b2 2025-06-22 | Added css stylesheet [Egor S]
|/
* ccfda38 2025-06-22 | Added index.html [Egor S]
* 2b9ca36 2025-06-22 | Moved hello.html to lib [Egor S]
* 6fb7745 2025-06-22 | Add author/email comment [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]
```

Создание конфликта

Для того, чтобы создать конфликт необходимо перейти в ветку master и внести изменения в файл hello.html



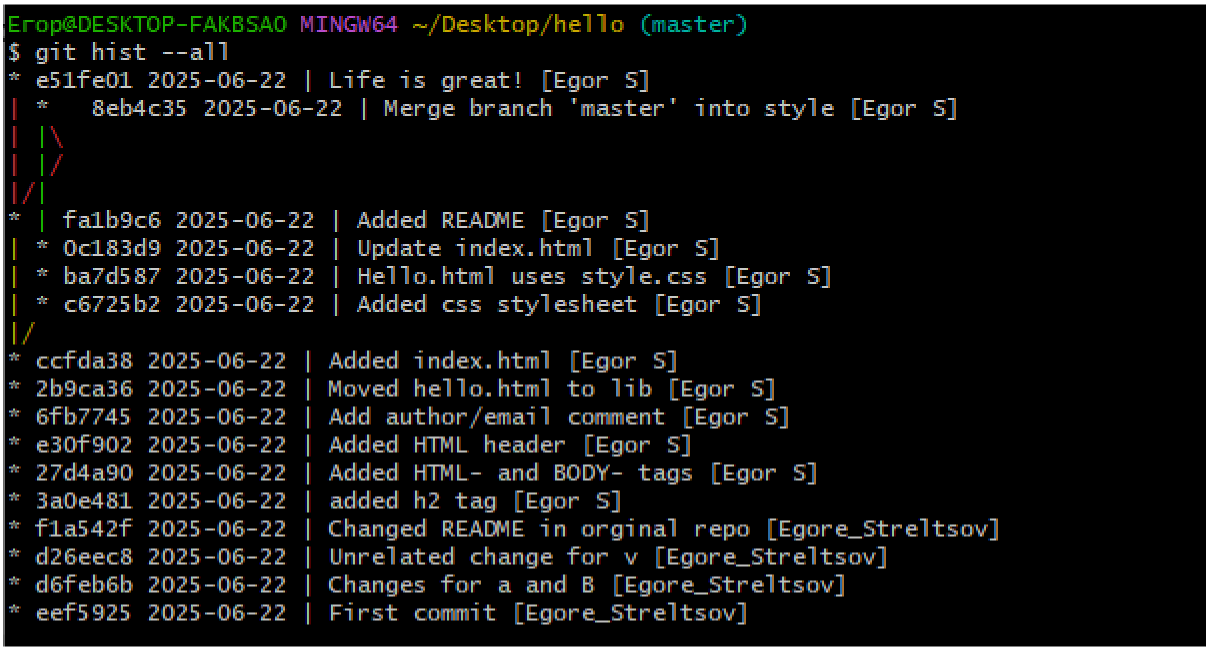
```
<!-- Author: Egor Streltsov (hivir2222@gmail.
<html>
    <head>
    <!-- no styel -->
    </head>
    <body>
        <h1>Hello, world</h1>
        <h2>Second line</h2>
    </body>
</html>
```



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

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git add lib/hello.html

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git commit -m "Life is great!"
[master e51fe01] Life is great!
 1 file changed, 1 insertion(+)
```

Индексация и коммит

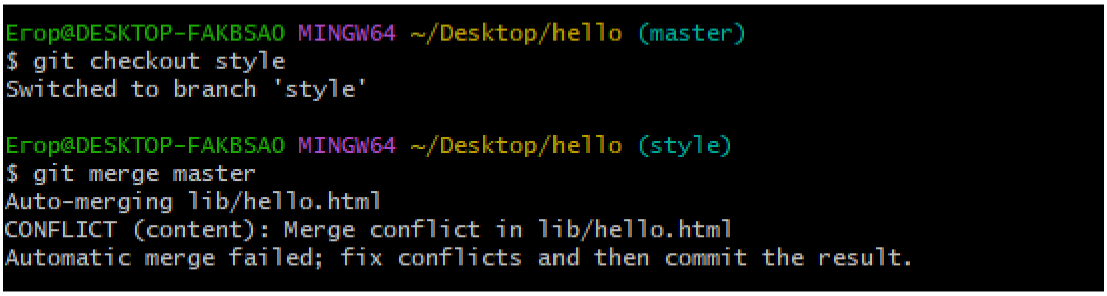
После выполнения предыдущих действий при просмотре веток можно будет увидеть конфликт в виде:



```
Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git hist --all
* e51fe01 2025-06-22 | Life is great! [Egor S]
| *   8eb4c35 2025-06-22 | Merge branch 'master' into style [Egor S]
| |\
| |/
|/|
* | fa1b9c6 2025-06-22 | Added README [Egor S]
| * 0c183d9 2025-06-22 | Update index.html [Egor S]
| * ba7d587 2025-06-22 | Hello.html uses style.css [Egor S]
| * c6725b2 2025-06-22 | Added css stylesheet [Egor S]
|/
* ccfda38 2025-06-22 | Added index.html [Egor S]
* 2b9ca36 2025-06-22 | Moved hello.html to lib [Egor S]
* 6fb7745 2025-06-22 | Add author/email comment [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]
```

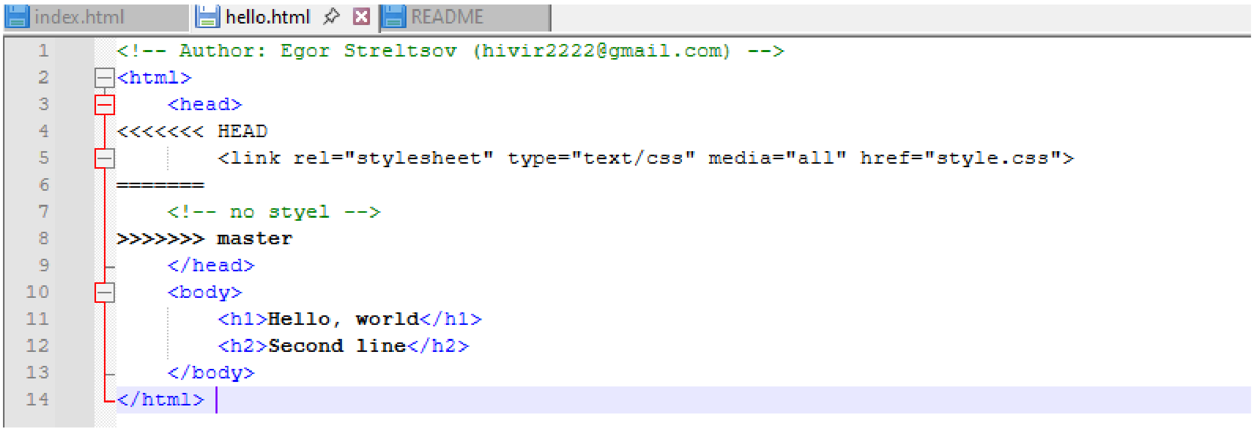
Разрешение конфликтов

При попытке объединить ветку style с master будет показана ошибка из-за конфликта



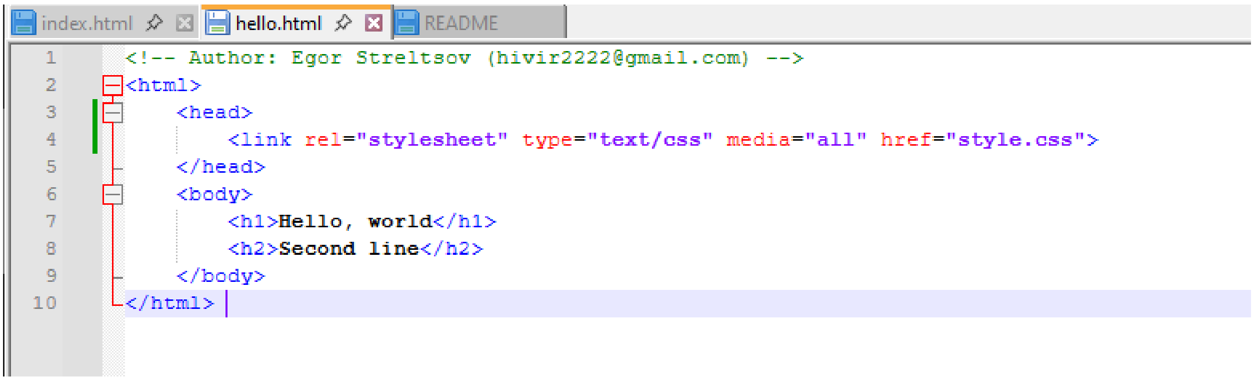
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git checkout style
Switched to branch 'style'

Егор@DESKTOP-FAKBSA0 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.
```

При открытии файла hello.html конфликт будет показан

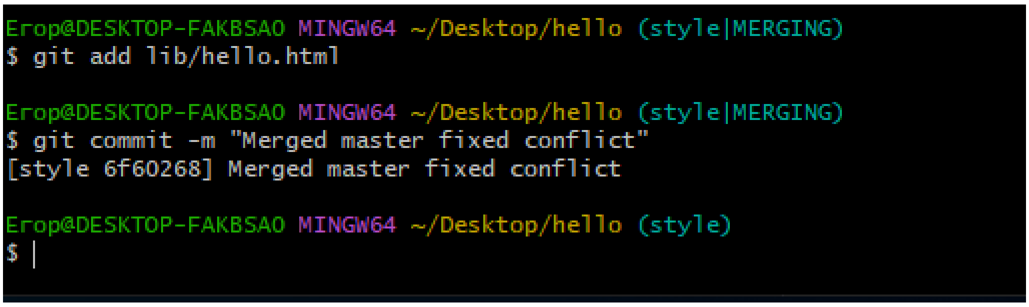


```
<!-- Author: Egor Streltsov (hivir2222@gmail.com) -->
<html>
    <head>
<<<<<<< HEAD
        <link rel="stylesheet" type="text/css" media="all" href="style.css">
=======
    <!-- no styel -->
>>>>>>> master
    </head>
    <body>
        <h1>Hello, world</h1>
        <h2>Second line</h2>
    </body>
</html>
```

Решение конфликта вручную



```
<!-- Author: Egor Streltsov (hivir2222@gmail.com) -->
<html>
    <head>
        <link rel="stylesheet" type="text/css" media="all" href="style.css">
    </head>
    <body>
        <h1>Hello, world</h1>
        <h2>Second line</h2>
    </body>
</html>
```

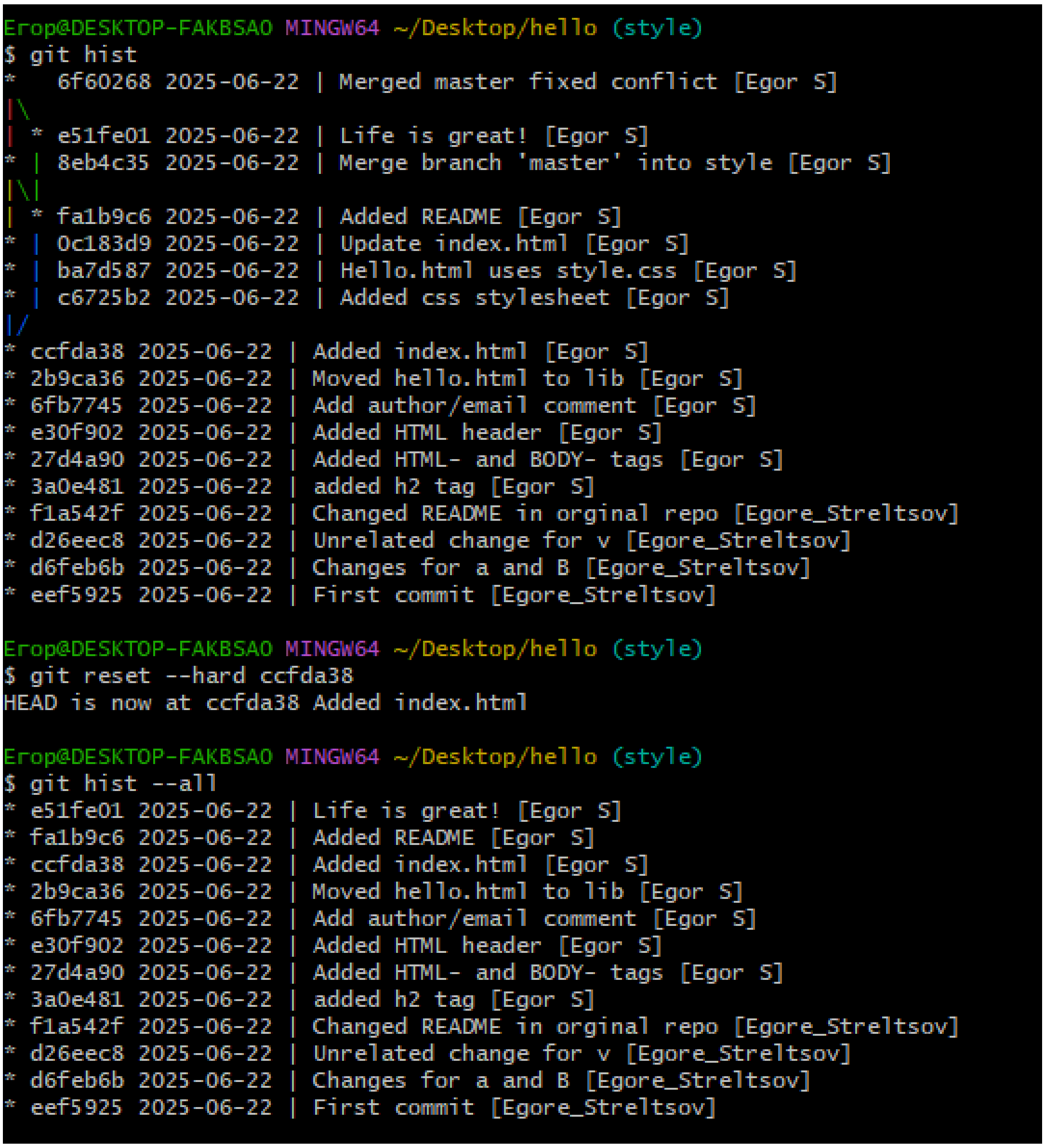


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style|MERGING)
$ git add lib/hello.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style|MERGING)
$ git commit -m "Merged master fixed conflict"
[style 6f60268] Merged master fixed conflict

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ 
```

Git 1.5

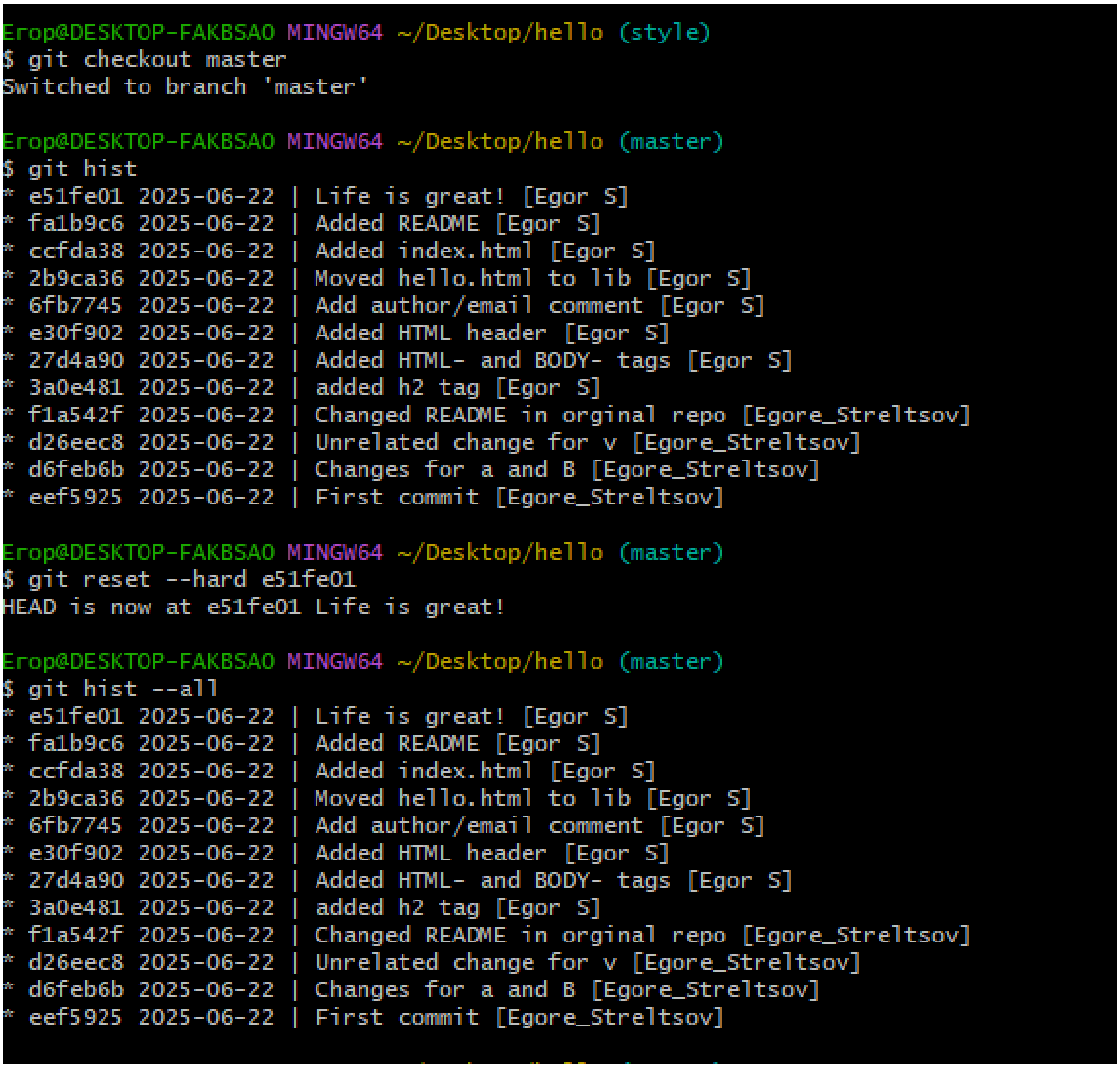
Сброс ветки style

Для сброса ветки необходимо применить команду reset --hard до требуемой точки



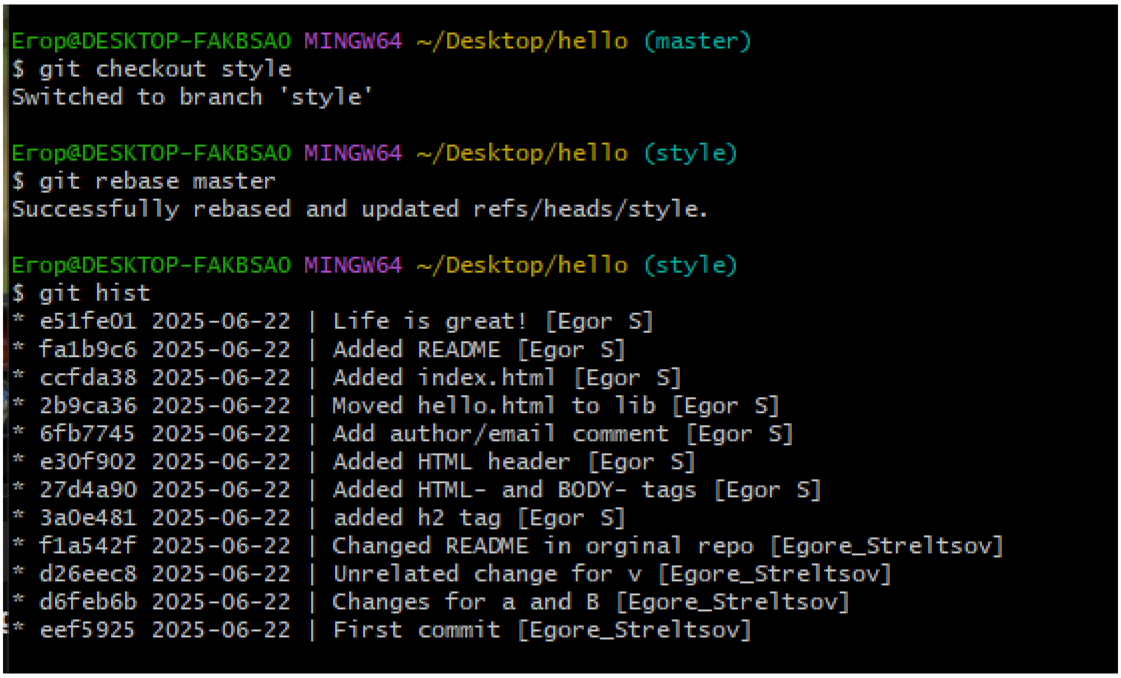
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ git hist
*   6f60268 2025-06-22 | Merged master fixed conflict [Egor S]
|\
| * e51fe01 2025-06-22 | Life is great! [Egor S]
* | 8eb4c35 2025-06-22 | Merge branch 'master' into style [Egor S]
|\|
| * fa1b9c6 2025-06-22 | Added README [Egor S]
* | 0c183d9 2025-06-22 | Update index.html [Egor S]
* | ba7d587 2025-06-22 | Hello.html uses style.css [Egor S]
* | c6725b2 2025-06-22 | Added css stylesheet [Egor S]
|/
* ccfda38 2025-06-22 | Added index.html [Egor S]
* 2b9ca36 2025-06-22 | Moved hello.html to lib [Egor S]
* 6fb7745 2025-06-22 | Add author/email comment [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ git reset --hard ccfda38
HEAD is now at ccfda38 Added index.html

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (style)
$ git hist --all
* e51fe01 2025-06-22 | Life is great! [Egor S]
* fa1b9c6 2025-06-22 | Added README [Egor S]
* ccfda38 2025-06-22 | Added index.html [Egor S]
* 2b9ca36 2025-06-22 | Moved hello.html to lib [Egor S]
* 6fb7745 2025-06-22 | Add author/email comment [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]
```

Сброс ветки master



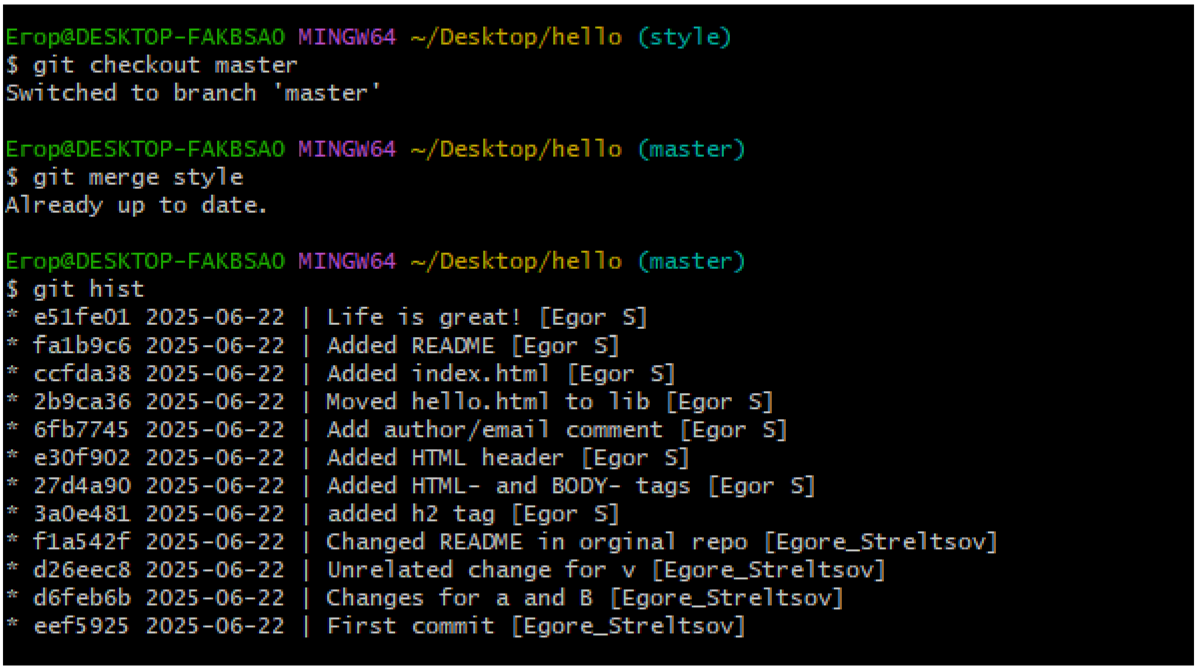
```
Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (style)
$ git checkout master
Switched to branch 'master'

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git hist
* e51fe01 2025-06-22 | Life is great! [Egor S]
* fa1b9c6 2025-06-22 | Added README [Egor S]
* ccfda38 2025-06-22 | Added index.html [Egor S]
* 2b9ca36 2025-06-22 | Moved hello.html to lib [Egor S]
* 6fb7745 2025-06-22 | Add author/email comment [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git reset --hard e51fe01
HEAD is now at e51fe01 Life is great!

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git hist --all
* e51fe01 2025-06-22 | Life is great! [Egor S]
* fa1b9c6 2025-06-22 | Added README [Egor S]
* ccfda38 2025-06-22 | Added index.html [Egor S]
* 2b9ca36 2025-06-22 | Moved hello.html to lib [Egor S]
* 6fb7745 2025-06-22 | Add author/email comment [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]
```

Перебазирование



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

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

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (style)
$ git hist
* e51fe01 2025-06-22 | Life is great! [Egor S]
* fa1b9c6 2025-06-22 | Added README [Egor S]
* ccfda38 2025-06-22 | Added index.html [Egor S]
* 2b9ca36 2025-06-22 | Moved hello.html to lib [Egor S]
* 6fb7745 2025-06-22 | Add author/email comment [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]
```

Слияние в ветку master

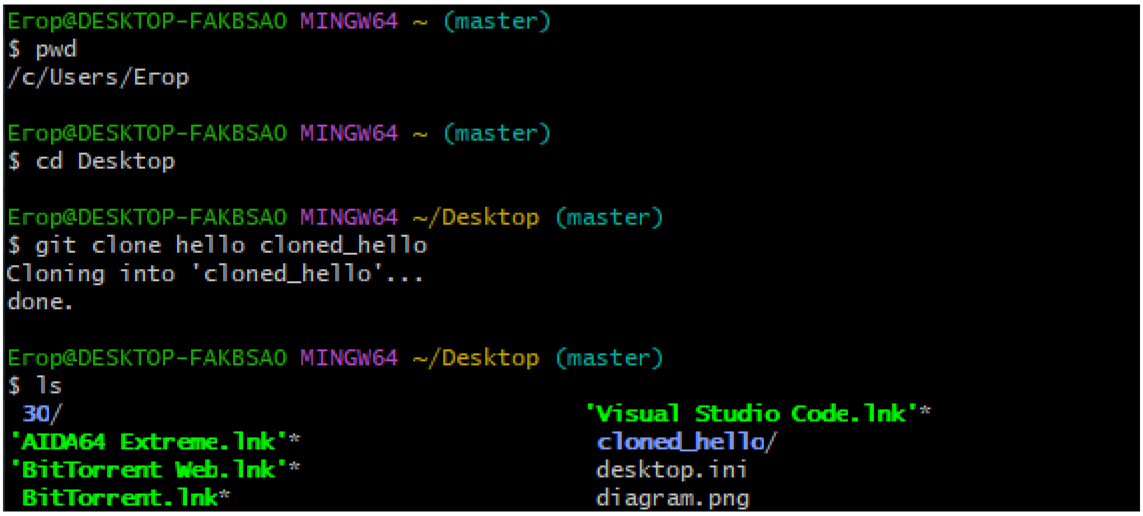


```
Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (style)
$ git checkout master
Switched to branch 'master'

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

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git hist
* e51fe01 2025-06-22 | Life is great! [Egor S]
* fa1b9c6 2025-06-22 | Added README [Egor S]
* ccfda38 2025-06-22 | Added index.html [Egor S]
* 2b9ca36 2025-06-22 | Moved hello.html to lib [Egor S]
* 6fb7745 2025-06-22 | Add author/email comment [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]
```

Клонирование репозиториев

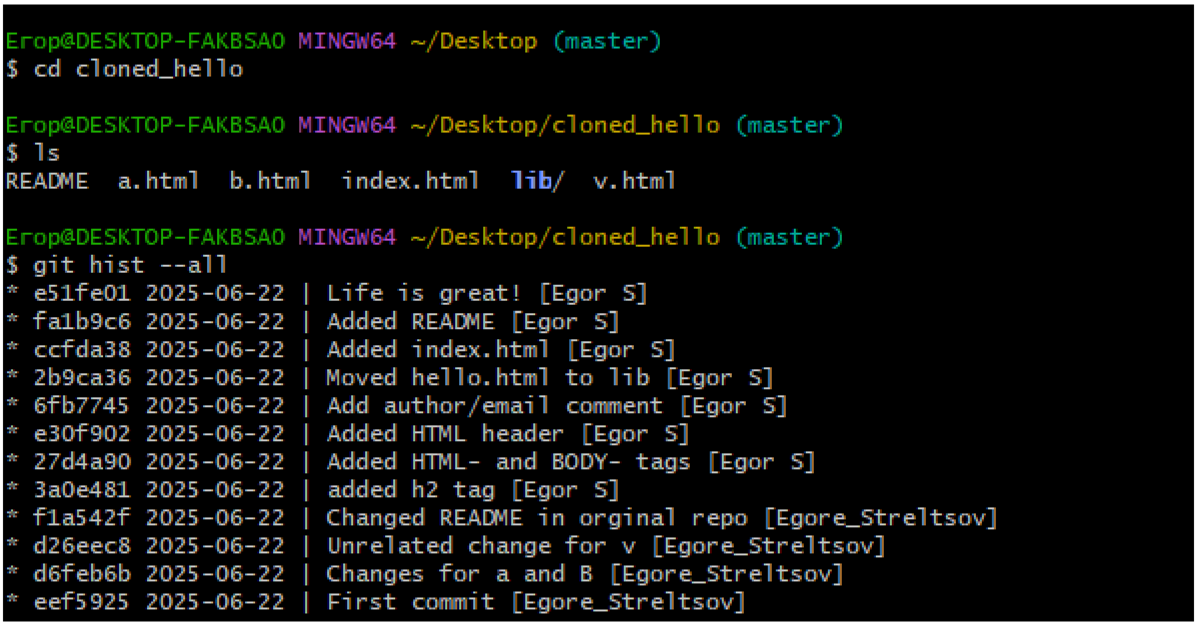
Далее требуется научиться делать копии репозиториев.

Для этого необходимо перейти в рабочий каталог и затем использовать команду git clone.



```
Егор@DESKTOP-FAKBSAO MINGW64 ~ (master)
$ pwd
/c/Users/Егор

Егор@DESKTOP-FAKBSAO MINGW64 ~ (master)
$ cd Desktop

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

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop (master)
$ ls
 30/                                     'Visual Studio Code.lnk'*
'AIDA64 Extreme.lnk'*                     cloned_hello/
'BitTorrent Web.lnk'*                     desktop.ini
 BitTorrent.lnk*                          diagram.png
```

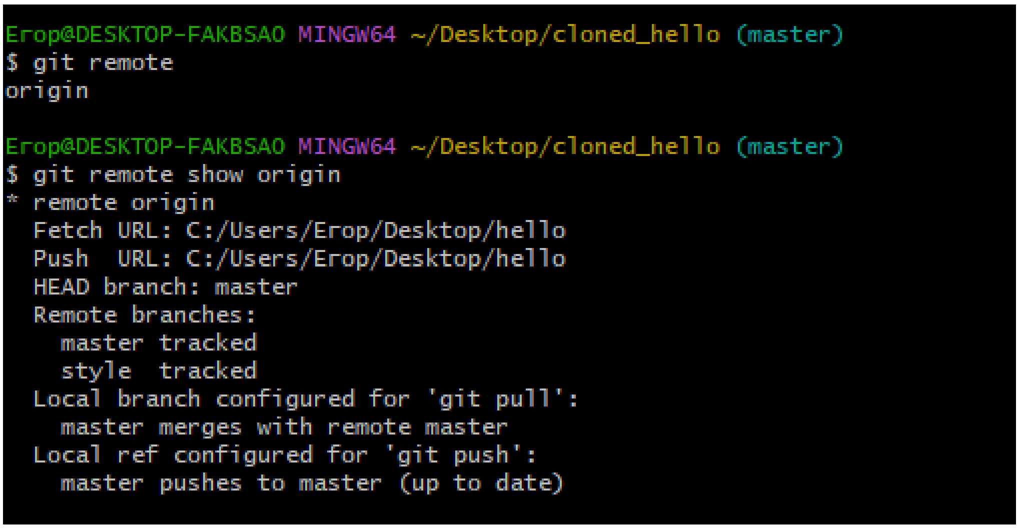
Просмотр клонированного репозитория



```
Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop (master)
$ cd cloned_hello

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

Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/cloned_hello (master)
$ git hist --all
* e51fe01 2025-06-22 | Life is great! [Egor S]
* fa1b9c6 2025-06-22 | Added README [Egor S]
* ccfda38 2025-06-22 | Added index.html [Egor S]
* 2b9ca36 2025-06-22 | Moved hello.html to lib [Egor S]
* 6fb7745 2025-06-22 | Add author/email comment [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]
```

Origin

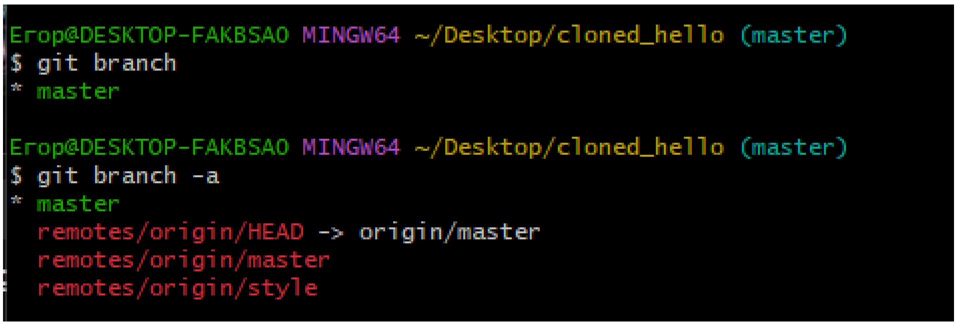
Origin – имя по умолчанию. Просмотр данных о нем возможен с помощью команд



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

Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/cloned_hello (master)
$ git remote show origin
* remote origin
  Fetch URL: C:/Users/Erop/Desktop/hello
  Push  URL: C:/Users/Erop/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)
```

Удаление ветки

Для просмотра удаленных веток используется команда git branch –a

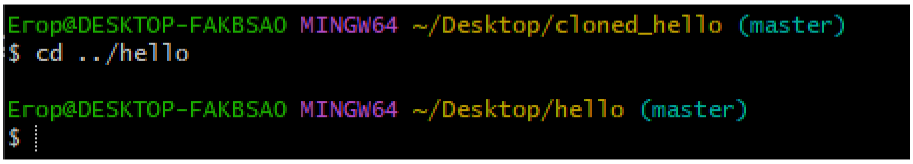


```
Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/cloned_hello (master)
$ git branch
* master

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

Git 1.6

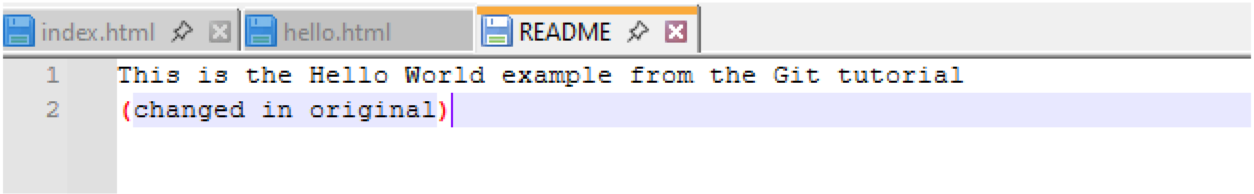
Работа с Git

Сначала необходимо внести изменения в оригинальный репозиторий. Для этого нужно перейти в данный репозиторий

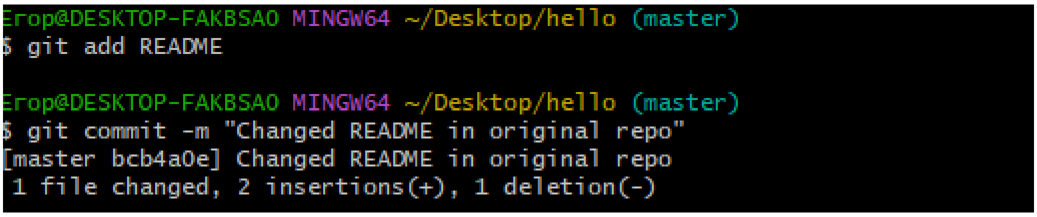


```
Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/cloned_hello (master)
$ cd ../hello

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ 
```

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

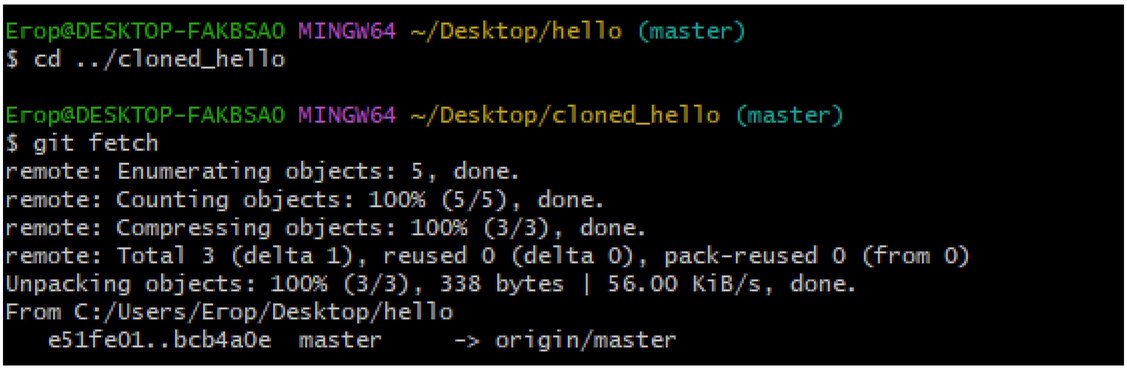


```
This is the Hello World example from the Git tutorial
(changed in original)
```



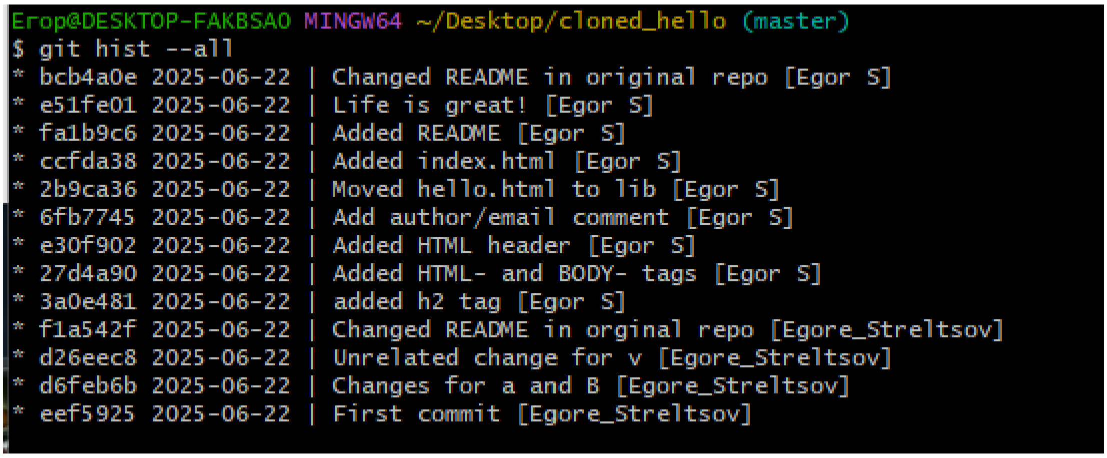
```
Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git add README

Егор@DESKTOP-FAKBSAO MINGW64 ~/Desktop/hello (master)
$ git commit -m "Changed README in original repo"
[master bcb4a0e] Changed README in original repo
 1 file changed, 2 insertions(+), 1 deletion(-)
```

Далее требуется перейти в клонированный репозиторий и извлечь изменения с помощью команды git fetch и просмотреть историю .



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

Егор@DESKTOP-FAKBSAO 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), 338 bytes | 56.00 KiB/s, done.
From C:/Users/Егор/Desktop/hello
   e51fe01..bcb4a0e  master     -> origin/master
```

Извлечение изменений, просмотр истории

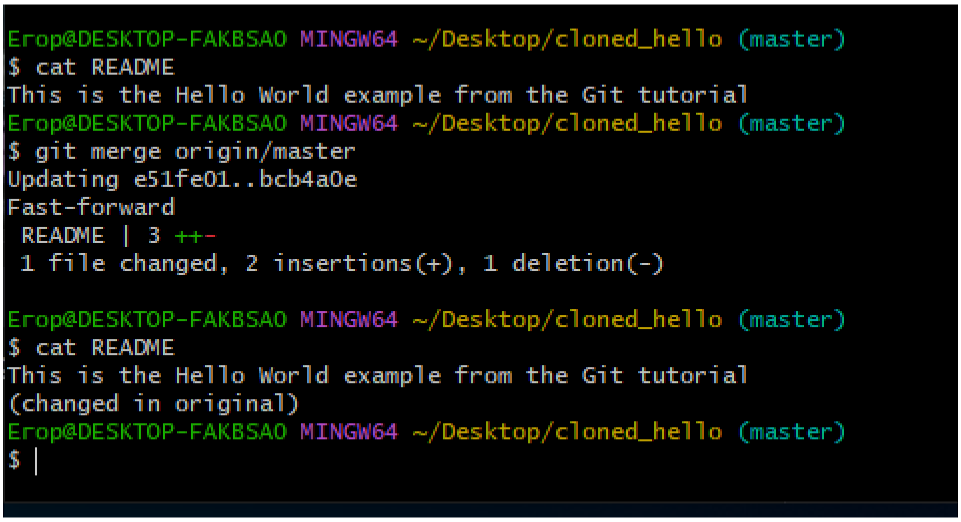


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git hist --all
* bcb4a0e 2025-06-22 | Changed README in original repo [Egor S]
* e51fe01 2025-06-22 | Life is great! [Egor S]
* fa1b9c6 2025-06-22 | Added README [Egor S]
* ccfda38 2025-06-22 | Added index.html [Egor S]
* 2b9ca36 2025-06-22 | Moved hello.html to lib [Egor S]
* 6fb7745 2025-06-22 | Add author/email comment [Egor S]
* e30f902 2025-06-22 | Added HTML header [Egor S]
* 27d4a90 2025-06-22 | Added HTML- and BODY- tags [Egor S]
* 3a0e481 2025-06-22 | added h2 tag [Egor S]
* f1a542f 2025-06-22 | Changed README in orginal repo [Egore_Streltsov]
* d26eec8 2025-06-22 | Unrelated change for v [Egore_Streltsov]
* d6feb6b 2025-06-22 | Changes for a and B [Egore_Streltsov]
* eef5925 2025-06-22 | First commit [Egore_Streltsov]
```

При попытке вывести содержимое файла README можно увидеть, что изменения не были внесены

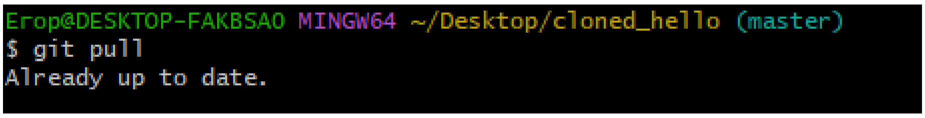
Далее нужно слить извлеченные изменения в ветку master

И после выполнения предыдущего действия при выводе README можно будет увидеть последние изменения



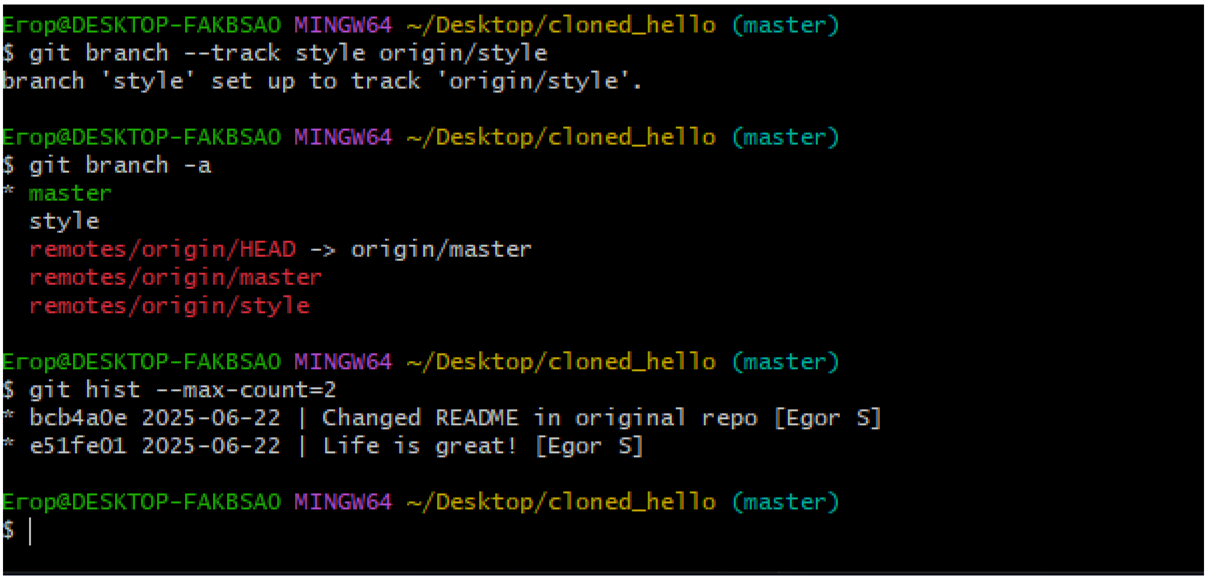
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ cat README
This is the Hello World example from the Git tutorial
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git merge origin/master
Updating e51fe01..bcb4a0e
Fast-forward
 README | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ cat README
This is the Hello World example from the Git tutorial
(changed in original)
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ |
```

Также существует команда, объединяющая функции git fetch и git merge



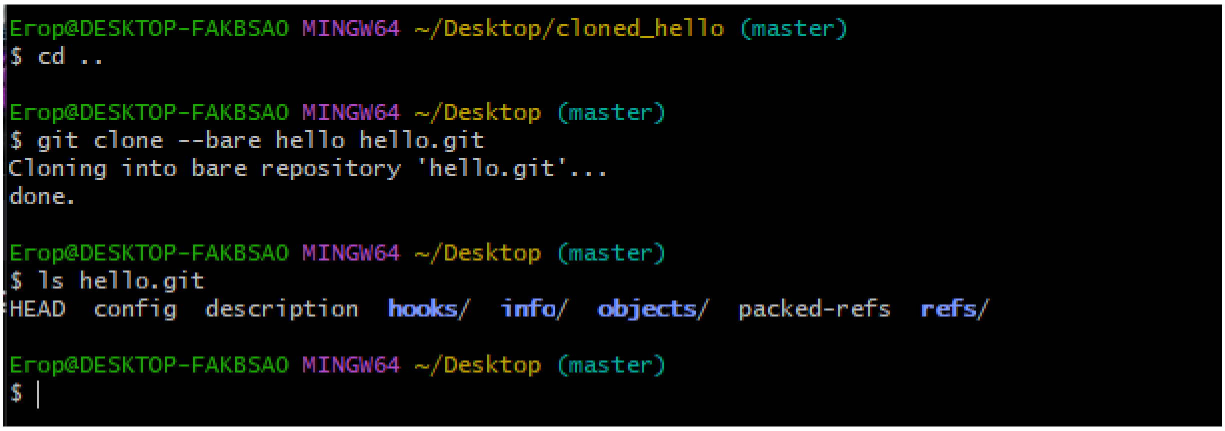
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git pull
Already up to date.
```

Далее требуется добавить локальную ветку, которая будет отслеживать удаленную ветку



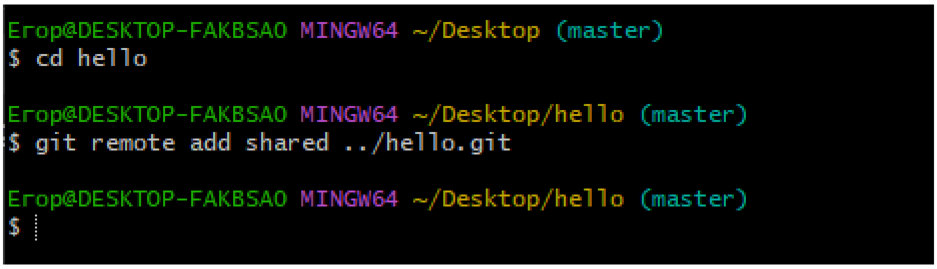
```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git branch --track style origin/style
branch 'style' set up to track 'origin/style'.

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git branch -a
* master
  style
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git hist --max-count=2
* bcb4a0e 2025-06-22 | Changed README in original repo [Egor S]
* e51fe01 2025-06-22 | Life is great! [Egor S]

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$
```

Далее необходимо создать чистый репозиторий



```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ cd ..

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop (master)
$ git clone --bare hello hello.git
Cloning into bare repository 'hello.git'...
done.

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop (master)
$ ls hello.git
HEAD  config  description  hooks/  info/  objects/  packed-refs  refs/

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop (master)
$
```

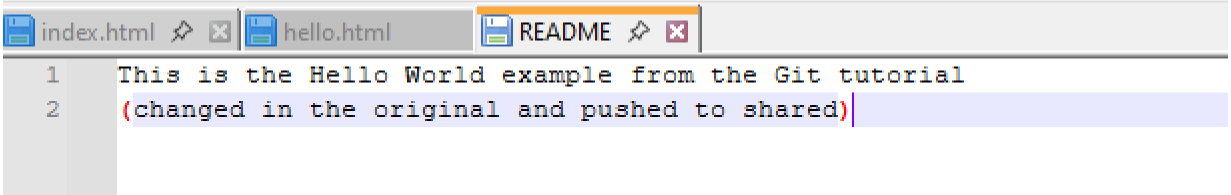
Для добавления удаленного репозитория используется команда git remote add shared



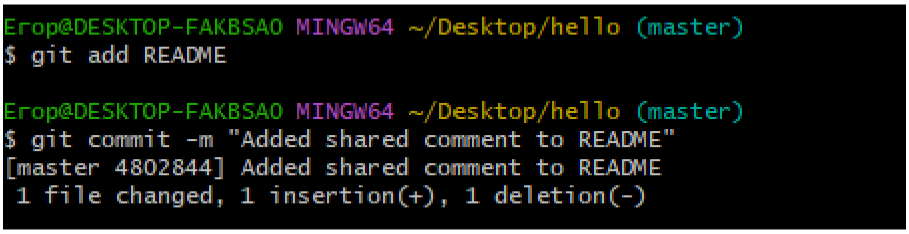
```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop (master)
$ cd hello

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git remote add shared ../hello.git

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$
```

Затем требуется научиться отправлять изменения в удаленный репозиторий.

Для этого сначала надо внести изменения, проиндексировать и произвести коммит.

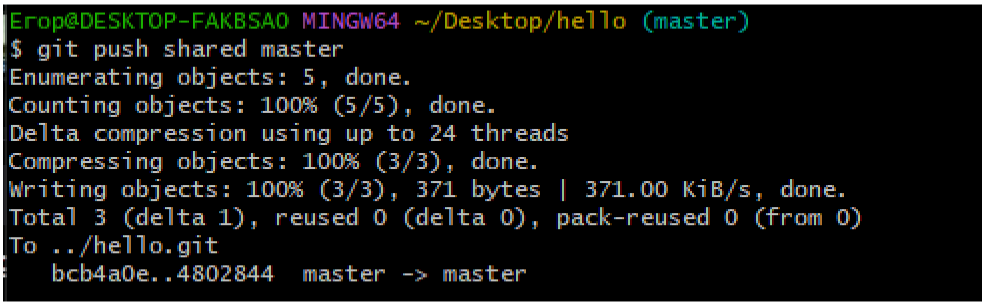


```
This is the Hello World example from the Git tutorial
(changed in the original and pushed to shared)
```



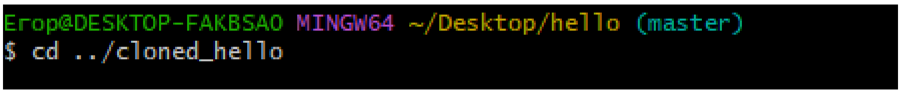
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git add README

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git commit -m "Added shared comment to README"
[master 4802844] Added shared comment to README
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Далее надо отправить изменения в общий репозиторий, используя команду git push shared master

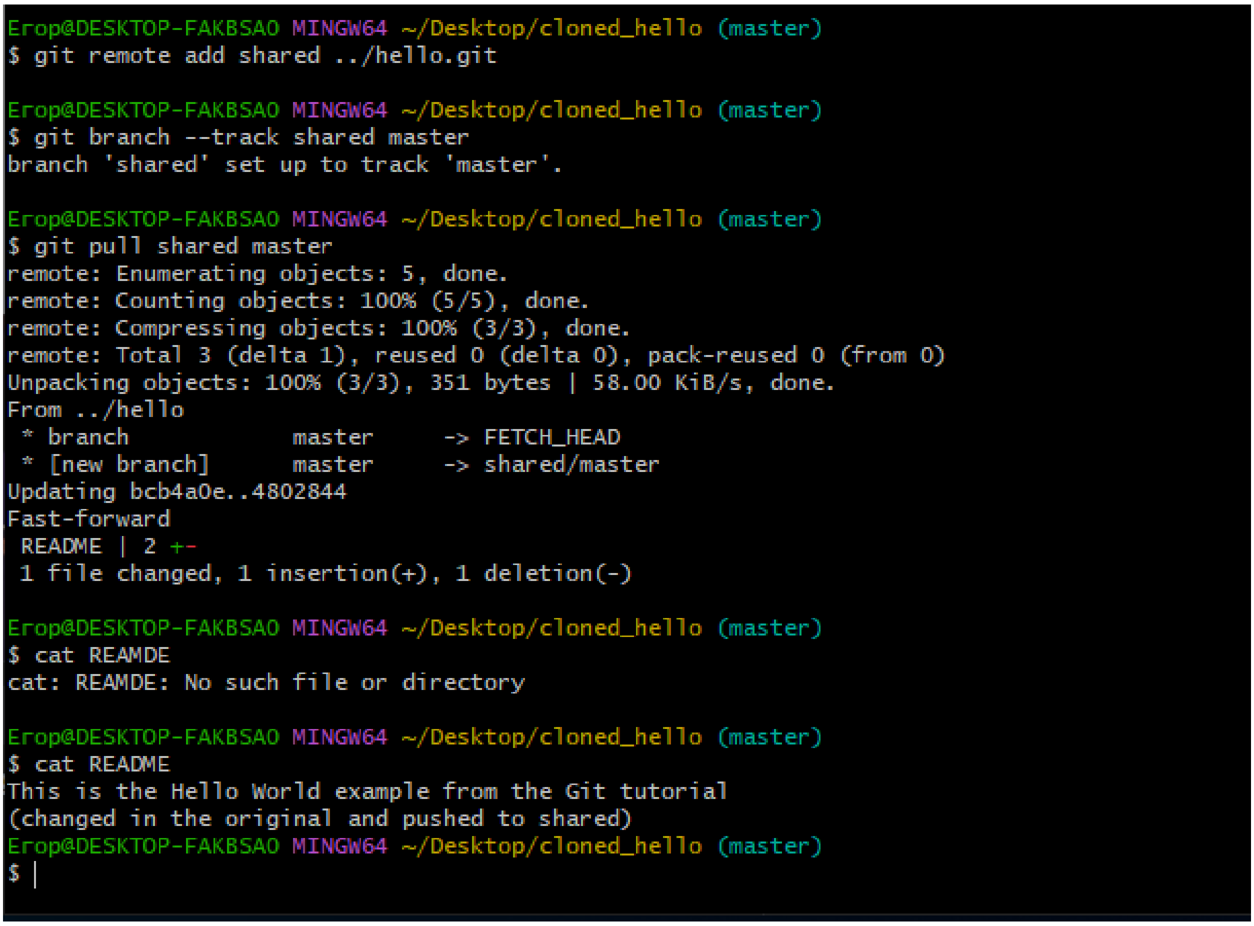


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ git push shared master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 24 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 371 bytes | 371.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
To ../hello.git
   bcb4a0e..4802844  master -> master
```

Для извлечения общих изменений нужно перейти в клонированный каталог и выполнить перечень команд

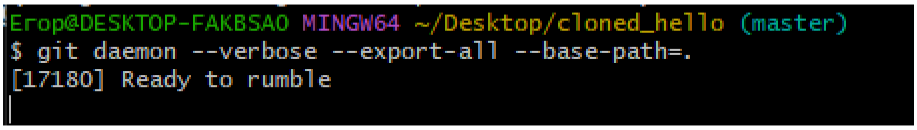


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello (master)
$ cd ../cloned_hello
```

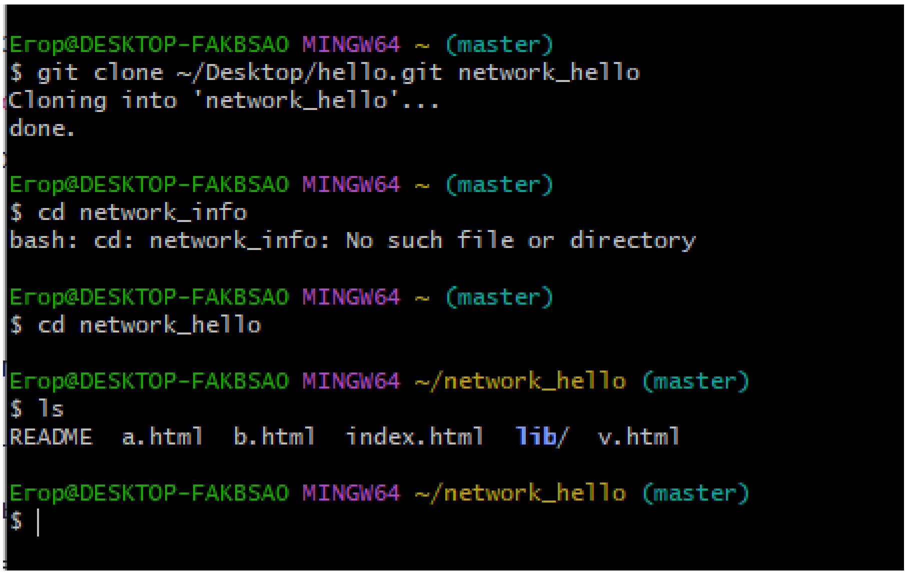


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git remote add shared ../hello.git

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git branch --track shared master
branch 'shared' set up to track 'master'.

Егор@DESKTOP-FAKBSA0 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), 351 bytes | 58.00 KiB/s, done.
From ../hello
 * branch            master     -> FETCH_HEAD
 * [new branch]      master     -> shared/master
Updating bcb4a0e..4802844
Fast-forward
 README | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ cat REAMDE
cat: REAMDE: No such file or directory

Егор@DESKTOP-FAKBSA0 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)
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ |
```

Для настройки git сервера нужно выполнить команду daemon –verbose –export-all –base-path= Затем в другом окне можно проверить работу сервера, сделав копию проекта hello



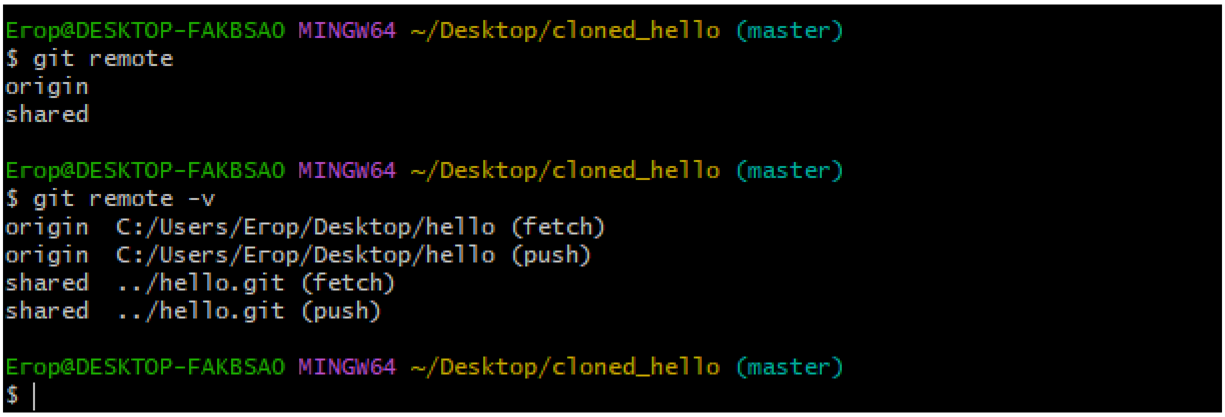
```
Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/cloned_hello (master)
$ git daemon --verbose --export-all --base-path=.
[17180] Ready to rumble
```



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

Erop@DESKTOP-FAKBSAO MINGW64 ~ (master)
$ cd network_info
bash: cd: network_info: No such file or directory

Erop@DESKTOP-FAKBSAO MINGW64 ~ (master)
$ cd network_hello

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

Erop@DESKTOP-FAKBSAO MINGW64 ~/network_hello (master)
$ |
```

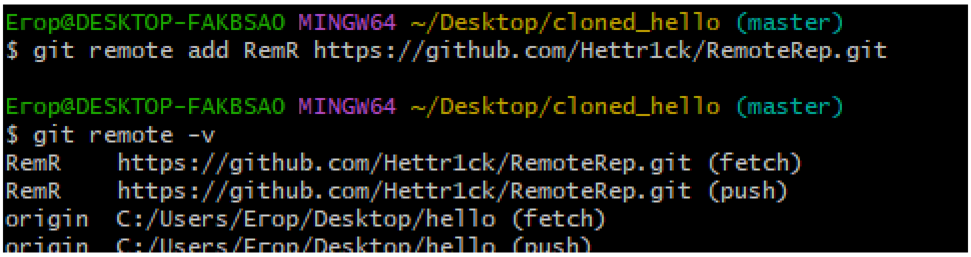
Git 1.7

Для того, чтобы просмотреть список настроенных удалённых репозиториев, необходимо запустить команду git remote



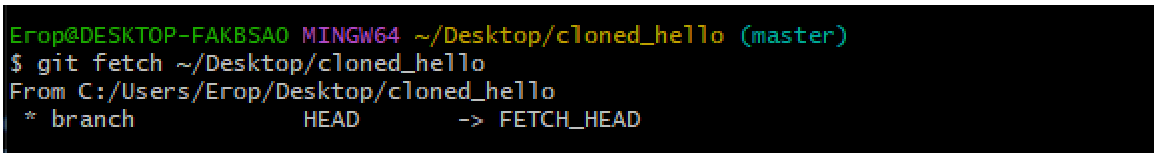
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git remote
origin
shared

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git remote -v
origin  C:/Users/Егор/Desktop/hello (fetch)
origin  C:/Users/Егор/Desktop/hello (push)
shared  ../hello.git (fetch)
shared  ../hello.git (push)

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ |
```

Для добавления удаленного репозитория с новым именем используется команда git remote add



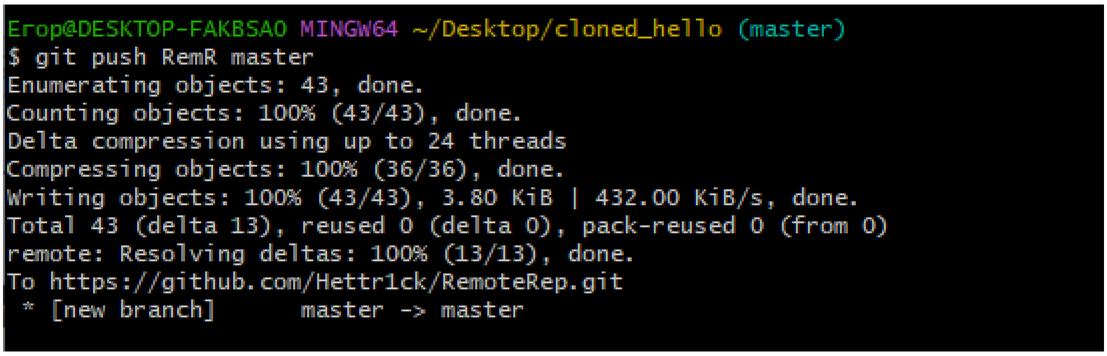
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git remote add RemR https://github.com/Hettr1ck/RemoteRep.git

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git remote -v
RemR    https://github.com/Hettr1ck/RemoteRep.git (fetch)
RemR    https://github.com/Hettr1ck/RemoteRep.git (push)
origin  C:/Users/Егор/Desktop/hello (fetch)
origin  C:/Users/Егор/Desktop/hello (push)
```

Для получения данных из удалённых проектов используется команда git fetch



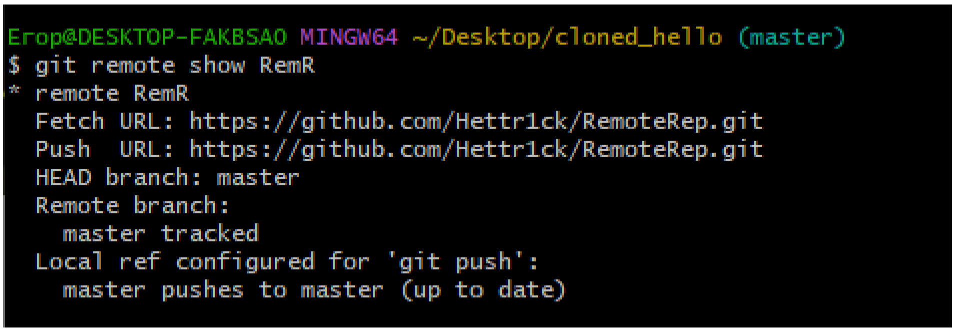
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git fetch ~/Desktop/cloned_hello
From C:/Users/Егор/Desktop/cloned_hello
 * branch            HEAD       -> FETCH_HEAD
```

Для отправки изменений в удаленный репозиторий используется команда git push



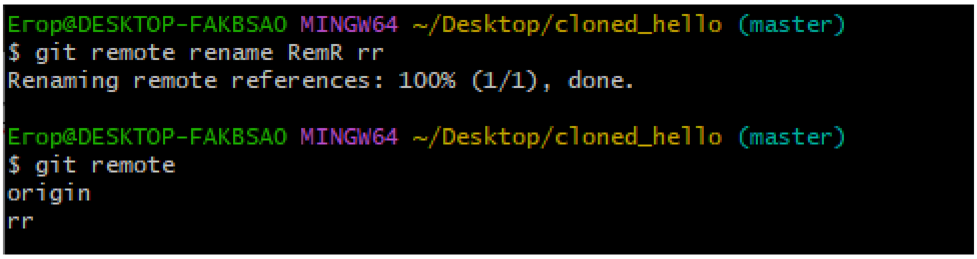
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git push RemR master
Enumerating objects: 43, done.
Counting objects: 100% (43/43), done.
Delta compression using up to 24 threads
Compressing objects: 100% (36/36), done.
Writing objects: 100% (43/43), 3.80 KiB | 432.00 KiB/s, done.
Total 43 (delta 13), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (13/13), done.
To https://github.com/Hettr1ck/RemoteRep.git
 * [new branch]      master -> master
```

Для получения информации об одном из удалённых репозиториев, можно использовать команду git remote show



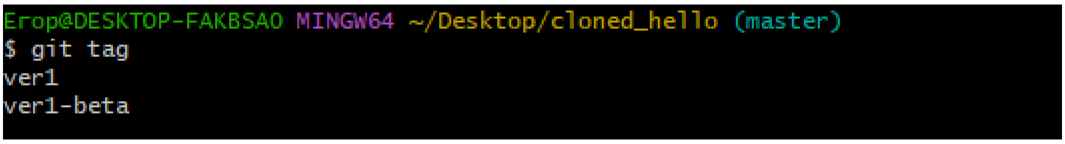
```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git remote show RemR
* remote RemR
  Fetch URL: https://github.com/Hettr1ck/RemoteRep.git
  Push  URL: https://github.com/Hettr1ck/RemoteRep.git
  HEAD branch: master
  Remote branch:
    master tracked
  Local ref configured for 'git push':
    master pushes to master (up to date)
```

Для переименования удаленных репозиториев используется команда git remote rename



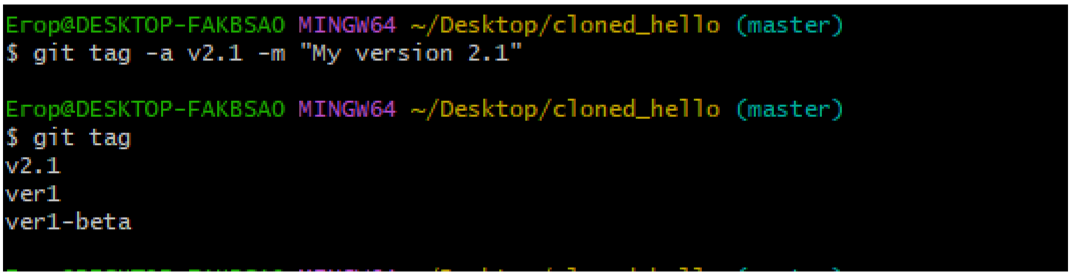
```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git remote rename RemR rr
Renaming remote references: 100% (1/1), done.

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git remote
origin
rr
```

Просмотреть существующие теги можно с помощью команды git tag



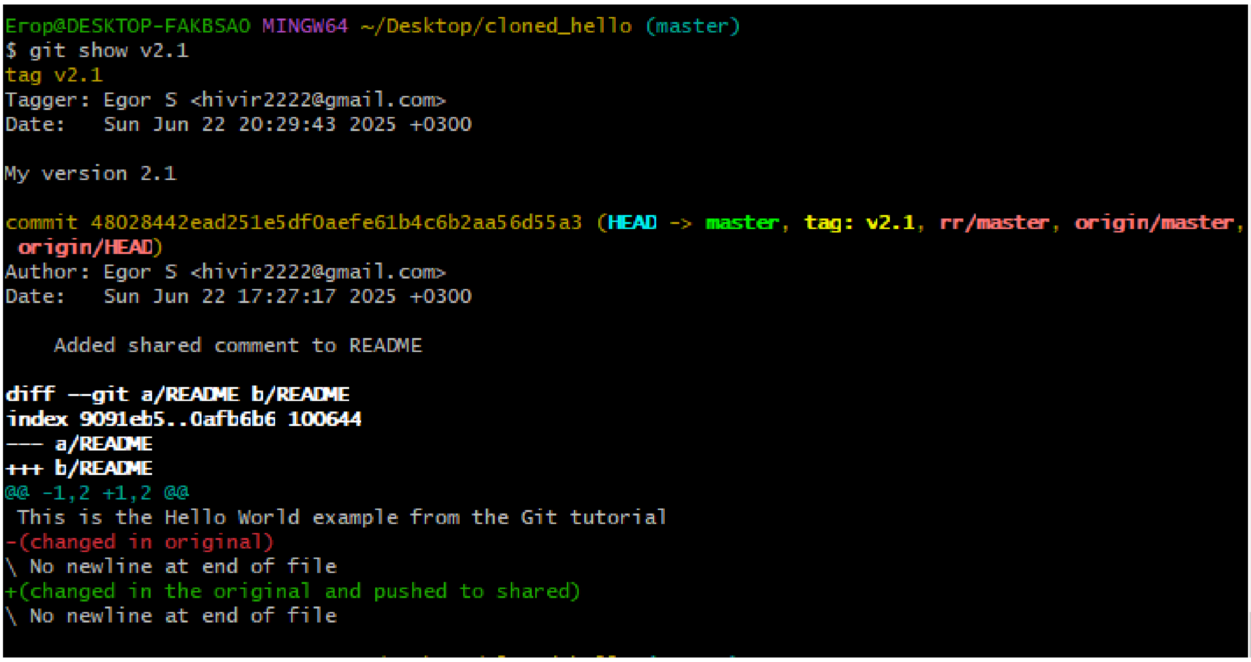
```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git tag
ver1
ver1-beta
```

Для создания аннотированной метки нужно выполнить команду



```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git tag -a v2.1 -m "My version 2.1"

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git tag
v2.1
ver1
ver1-beta
```

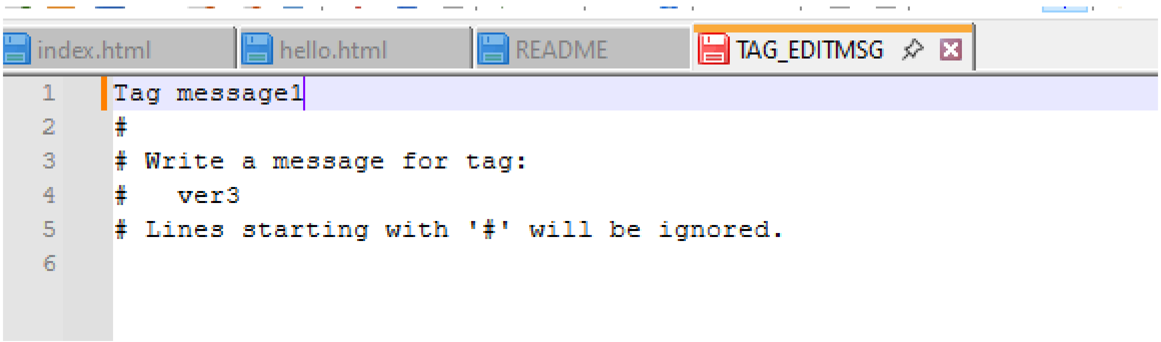
Команда git show осуществляет просмотр данных тегов вместе с коммитом



```
Erop@DESKTOP-FAKBSAO MINGW64 ~/Desktop/cloned_hello (master)
$ git show v2.1
tag v2.1
Tagger: Egor S <hivir2222@gmail.com>
Date:   Sun Jun 22 20:29:43 2025 +0300

My version 2.1

commit 48028442ead251e5df0aefe61b4c6b2aa56d55a3 (HEAD -> master, tag: v2.1, rr/master, origin/master,
 origin/HEAD)
Author: Egor S <hivir2222@gmail.com>
Date:   Sun Jun 22 17:27:17 2025 +0300

    Added shared comment to README

diff --git a/README b/README
index 9091eb5..0afb6b6 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
```

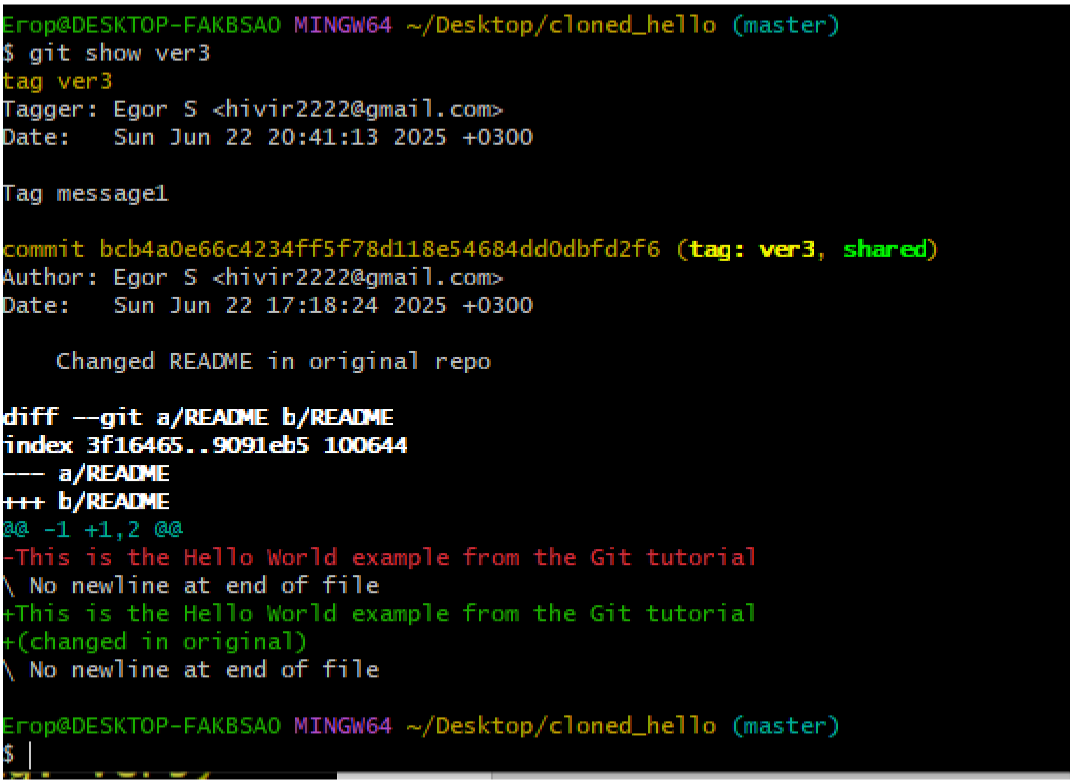
Для отметки определенного коммита тегом надо указать его хэш

Создание тега определенному коммиту



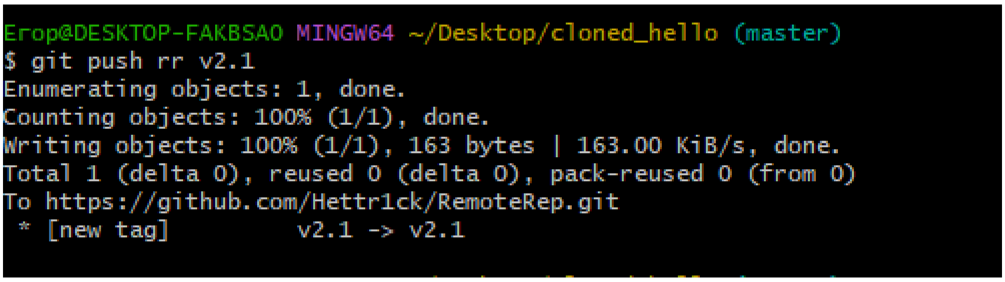
```
Tag message1
#
# Write a message for tag:
#   ver3
# Lines starting with '#' will be ignored.

```

Данные этого тега можно просмотреть аналогичным образом



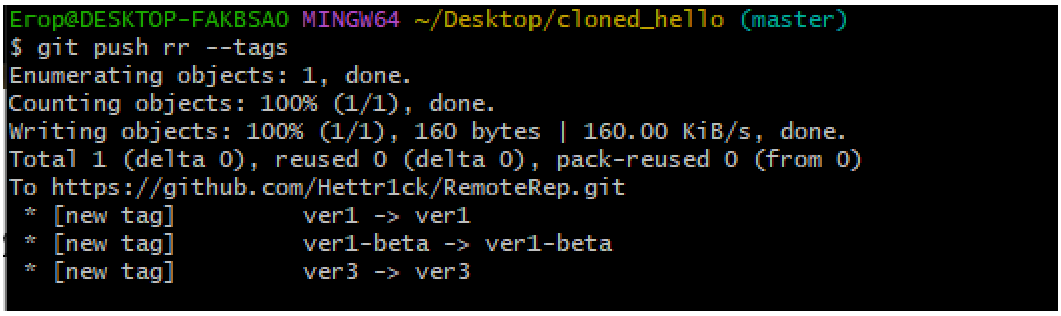
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git show ver3
tag ver3
Tagger: Egor S <hivir2222@gmail.com>
Date:   Sun Jun 22 20:41:13 2025 +0300

Tag message1

commit bcb4a0e66c4234ff5f78d118e54684dd0dbfd2f6 (tag: ver3, shared)
Author: Egor S <hivir2222@gmail.com>
Date:   Sun Jun 22 17:18:24 2025 +0300

    Changed README in original repo

diff --git a/README b/README
index 3f16465..9091eb5 100644
--- a/README
+++ b/README
@@ -1 +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

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ |
```

По умолчанию, команда git push не отправляет теги на удалённые сервера. Нужно выполнить команду git push rr



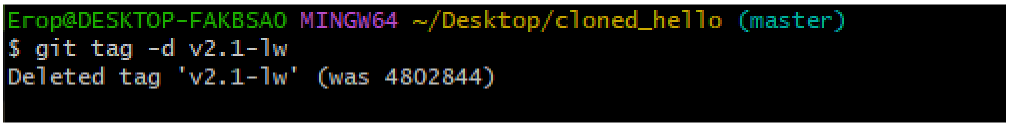
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git push rr v2.1
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 163 bytes | 163.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Hettr1ck/RemoteRep.git
 * [new tag]         v2.1 -> v2.1
```

Можно использовать опцию --tags для команды git push. В таком случае все теги отправятся на удалённый сервер



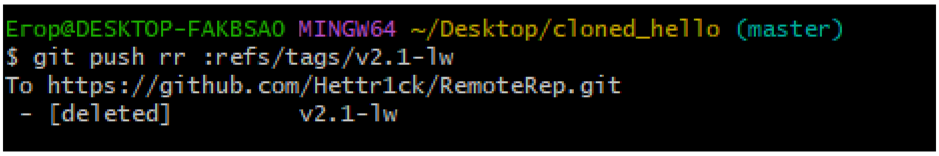
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git push rr --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 160 bytes | 160.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Hettr1ck/RemoteRep.git
 * [new tag]         ver1 -> ver1
 * [new tag]         ver1-beta -> ver1-beta
 * [new tag]         ver3 -> ver3
```

Для того, чтобы удалить тег, надо использовать команду git tag с параметром –d



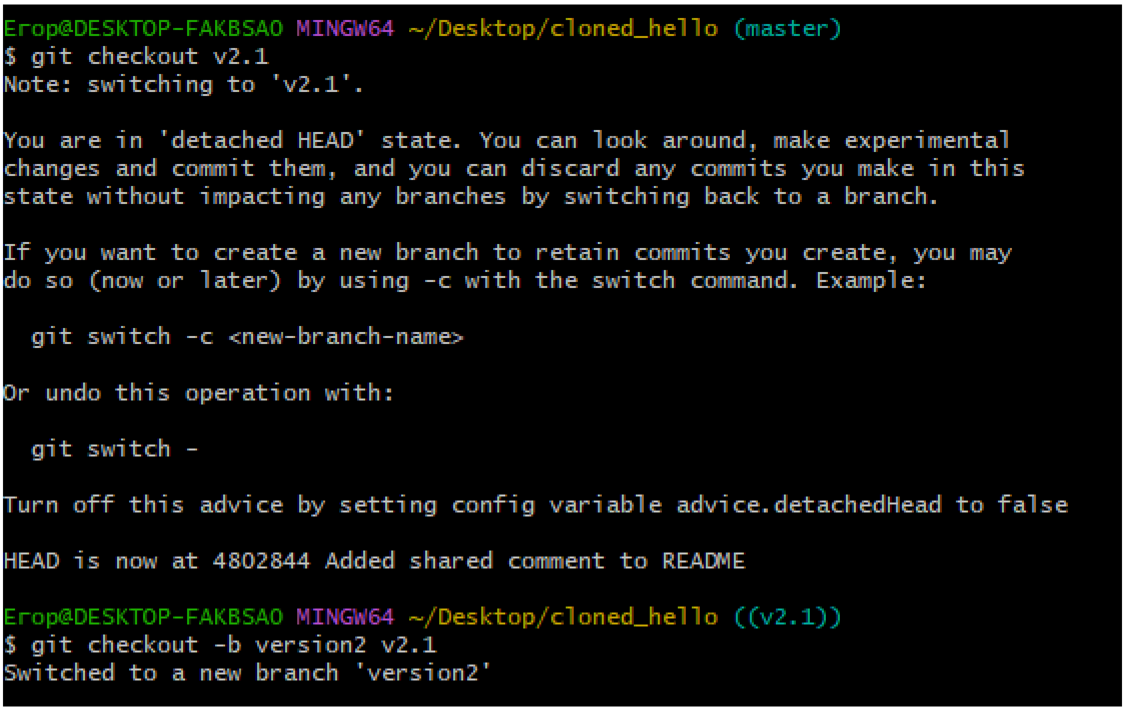
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git tag -d v2.1-lw
Deleted tag 'v2.1-lw' (was 4802844)
```

Для удаления тега с сервера используется команда :refs/tags



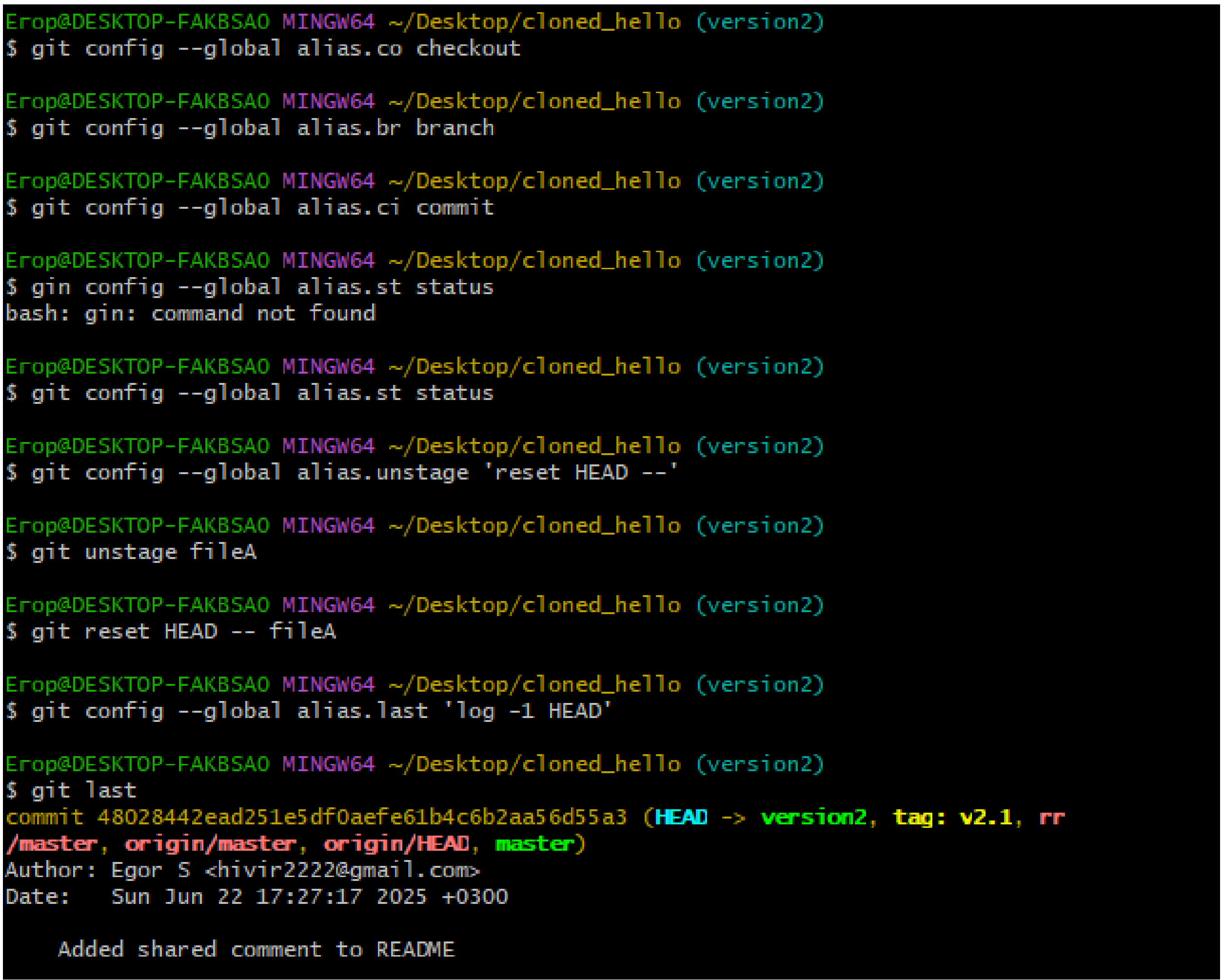
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git push rr :refs/tags/v2.1-lw
To https://github.com/Hettr1ck/RemoteRep.git
 - [deleted]         v2.1-lw
```

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

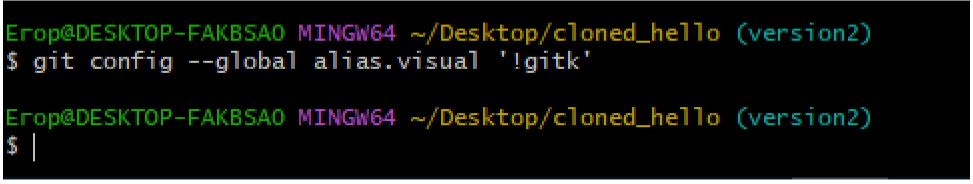


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (master)
$ git checkout v2.1
Note: switching to 'v2.1'.

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 4802844 Added shared comment to README

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello ((v2.1))
$ git checkout -b version2 v2.1
Switched to a new branch 'version2'
```

Создание алиасов:



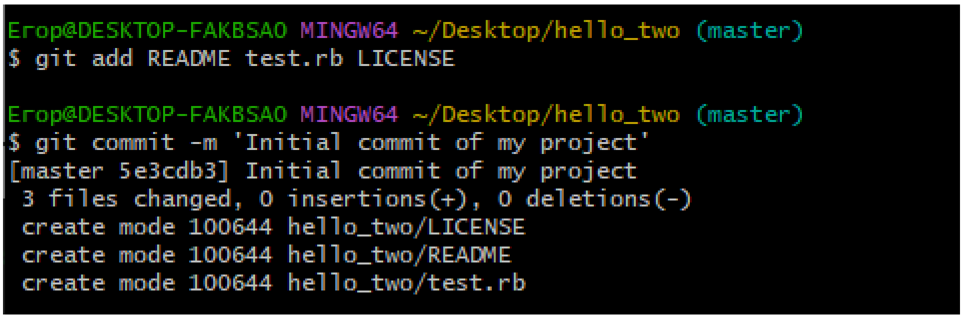
```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (version2)
$ git config --global alias.co checkout

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (version2)
$ git config --global alias.br branch

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (version2)
$ git config --global alias.ci commit

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (version2)
$ gin config --global alias.st status
bash: gin: command not found

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (version2)
$ git config --global alias.st status

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (version2)
$ git config --global alias.unstage 'reset HEAD --'

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (version2)
$ git unstage fileA

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (version2)
$ git reset HEAD -- fileA

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (version2)
$ git config --global alias.last 'log -1 HEAD'

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (version2)
$ git last
commit 48028442ead251e5df0aefe61b4c6b2aa56d55a3 (HEAD -> version2, tag: v2.1, rr
/master, origin/master, origin/HEAD, master)
Author: Egor S <hivir2222@gmail.com>
Date:   Sun Jun 22 17:27:17 2025 +0300

    Added shared comment to README
```



```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (version2)
$ git config --global alias.visual '!gitk'

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/cloned_hello (version2)
$ |
```

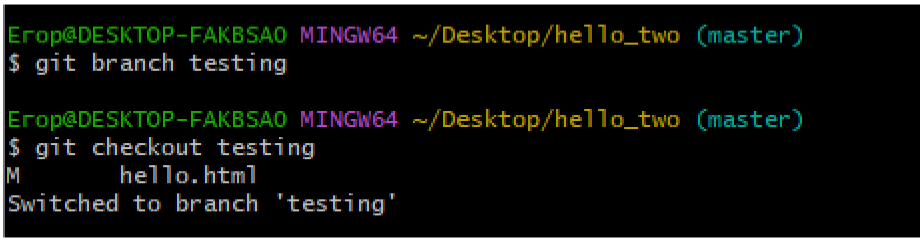
Git 1.8

Для начала следует создать репозиторий, создать 3 файла и добавить их в коммит



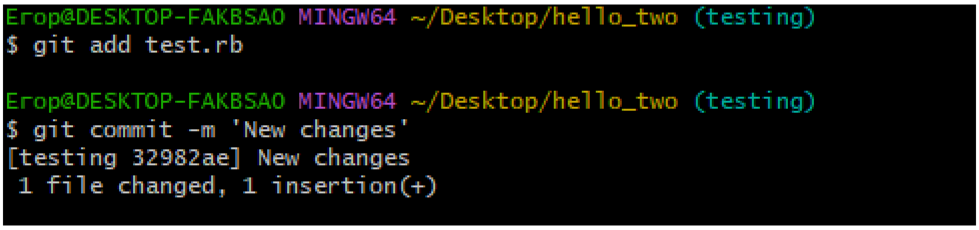
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git add README test.rb LICENSE

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git commit -m 'Initial commit of my project'
[master 5e3cdb3] Initial commit of my project
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hello_two/LICENSE
 create mode 100644 hello_two/README
 create mode 100644 hello_two/test.rb
```

Затем надо создать ветку testing и переключиться на нее



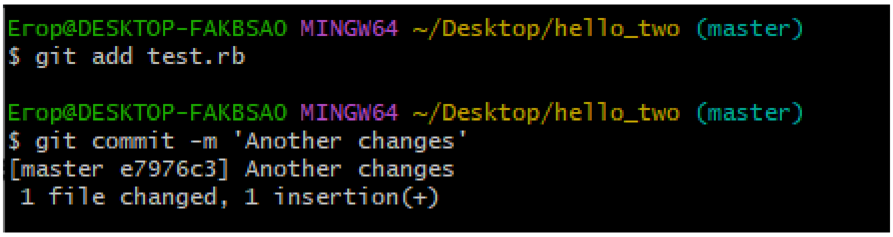
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git branch testing

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git checkout testing
M       hello.html
Switched to branch 'testing'
```

Далее надо внести изменения в файл test.rb и создать коммит



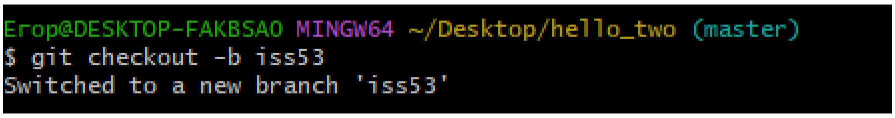
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (testing)
$ git add test.rb

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (testing)
$ git commit -m 'New changes'
[testing 32982ae] New changes
 1 file changed, 1 insertion(+)
```

Затем необходимо переключиться на ветку master и внести изменения в файл test.rb на этой ветке



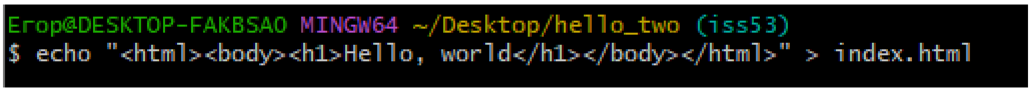
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git add test.rb

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git commit -m 'Another changes'
[master e7976c3] Another changes
 1 file changed, 1 insertion(+)
```

Команда git checkout -b позволяет сразу создать и переключиться на ветку

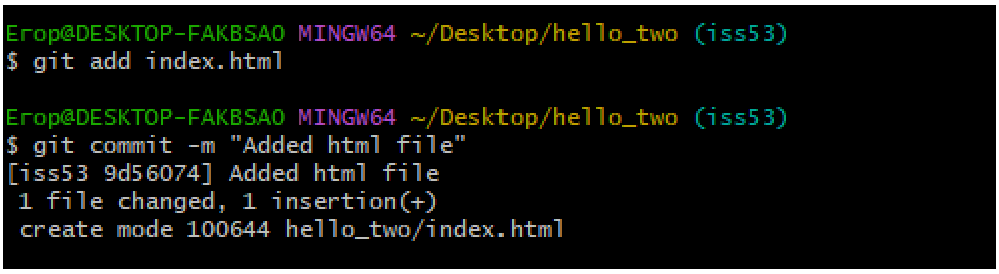


```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git checkout -b iss53
Switched to a new branch 'iss53'
```

В новой ветке нужно внести в файл изменения

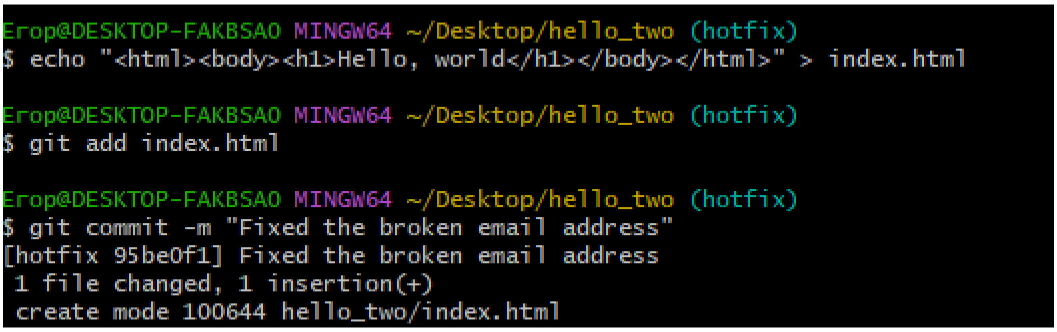


```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (iss53)
$ echo "<html><body><h1>Hello, world</h1></body></html>" > index.html
```

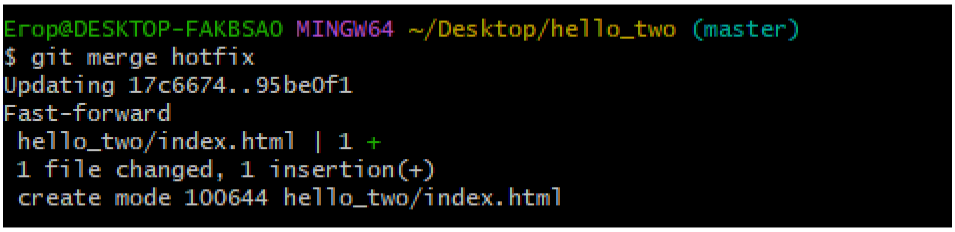


```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (iss53)
$ git add index.html

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (iss53)
$ git commit -m "Added html file"
[iss53 9d56074] Added html file
 1 file changed, 1 insertion(+)
 create mode 100644 hello_two/index.html
```

Затем надо на ветке hotfix добавить изменения в файл index.html, а затем слить эту ветку и master

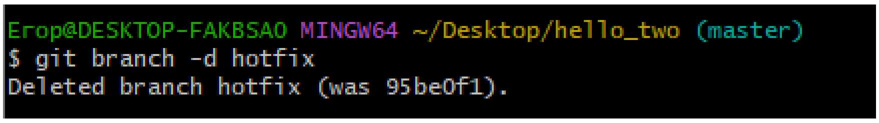


```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (hotfix)
$ echo "<html><body><h1>Hello, world</h1></body></html>" > index.html

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (hotfix)
$ git add index.html

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (hotfix)
$ git commit -m "Fixed the broken email address"
[hotfix 95be0f1] Fixed the broken email address
 1 file changed, 1 insertion(+)
 create mode 100644 hello_two/index.html
```



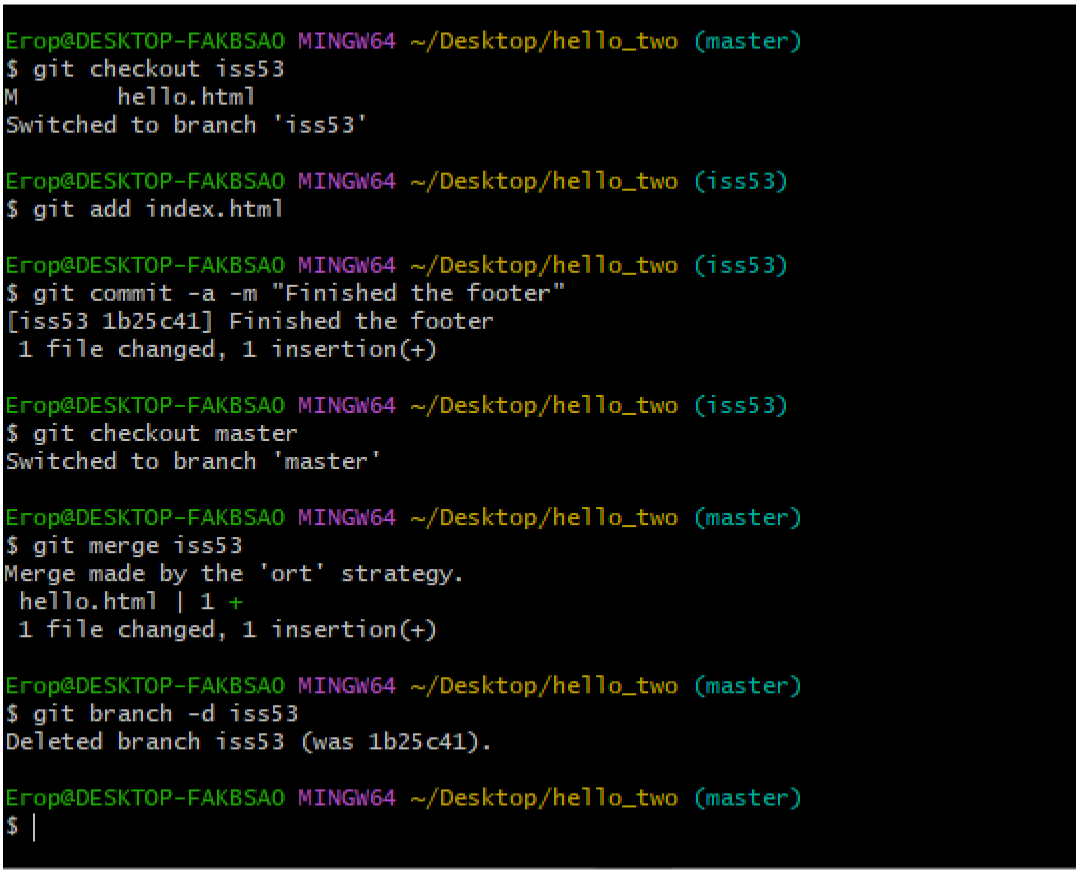
```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git merge hotfix
Updating 17c6674..95be0f1
Fast-forward
 hello_two/index.html | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 hello_two/index.html
```

Удаление ветки hotfix



```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git branch -d hotfix
Deleted branch hotfix (was 95be0f1).
```

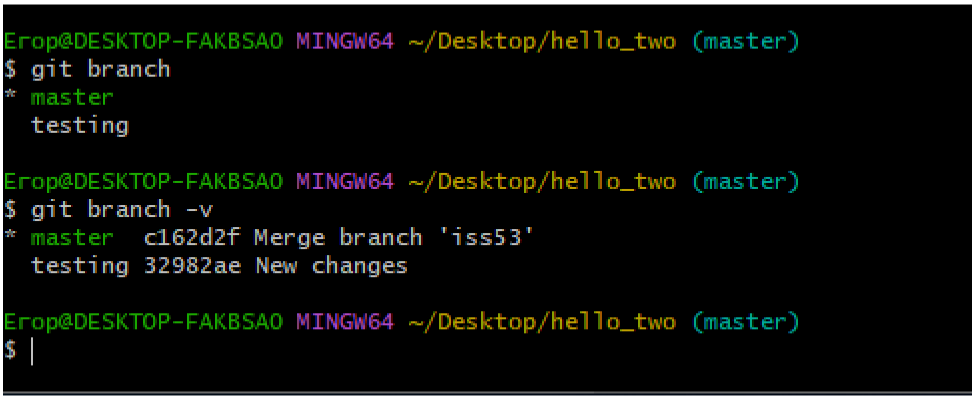
Слияние веток master и iss53 с удалением последней



```
Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git checkout iss53
M       hello.html
Switched to branch 'iss53'

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (iss53)
$ git add index.html

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (iss53)
$ git commit -a -m "Finished the footer"
[iss53 1b25c41] Finished the footer
 1 file changed, 1 insertion(+)

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (iss53)
$ git checkout master
Switched to branch 'master'

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git merge iss53
Merge made by the 'ort' strategy.
 hello.html | 1 +
 1 file changed, 1 insertion(+)

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git branch -d iss53
Deleted branch iss53 (was 1b25c41).

Erop@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ |
```

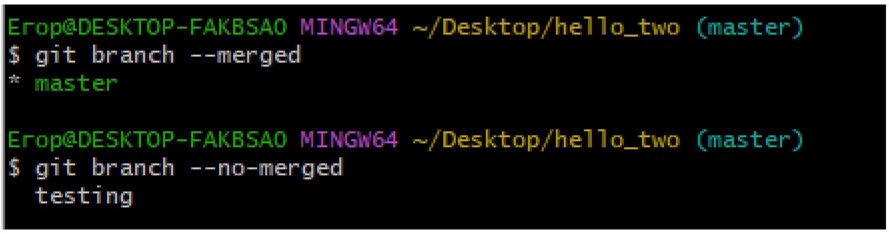
Git 1.9

Просмотр всех веток и последние коммиты в каждой:



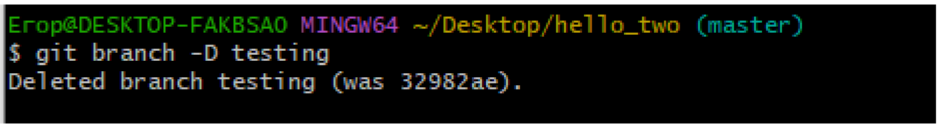
```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git branch
* master
  testing

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git branch -v
* master  c162d2f Merge branch 'iss53'
  testing 32982ae New changes

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$
```

Опции --merged и --no-merged могут отфильтровать этот список для вывода только тех веток, которые слиты или ещё не слиты в текущую ветку



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git branch --merged
* master

Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git branch --no-merged
  testing
```

Затем следует удалить ветку testing



```
Егор@DESKTOP-FAKBSA0 MINGW64 ~/Desktop/hello_two (master)
$ git branch -D testing
Deleted branch testing (was 32982ae).
```