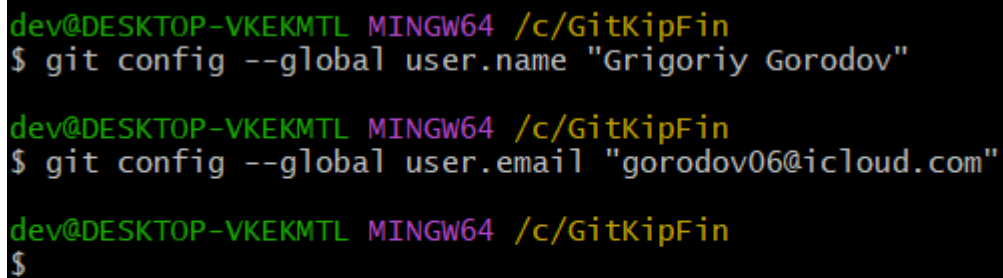


Работа с Git

Подготовка

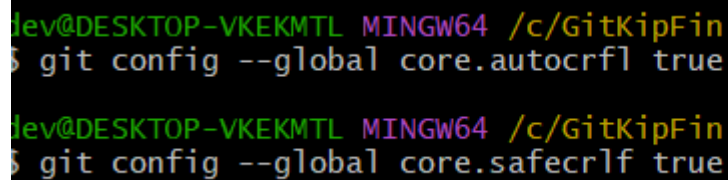


```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ git config --global user.name "Grigoriy Gorodov"

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ git config --global user.email "gorodov06@icloud.com"

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$
```

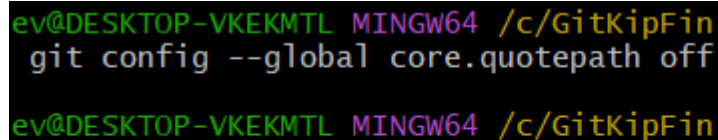
Рисунок 1 - Установка имени и электронной почты



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ git config --global core.autocrlf true

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ git config --global core.safecrlf true
```

Рисунок 2 - Параметры окончаний строк

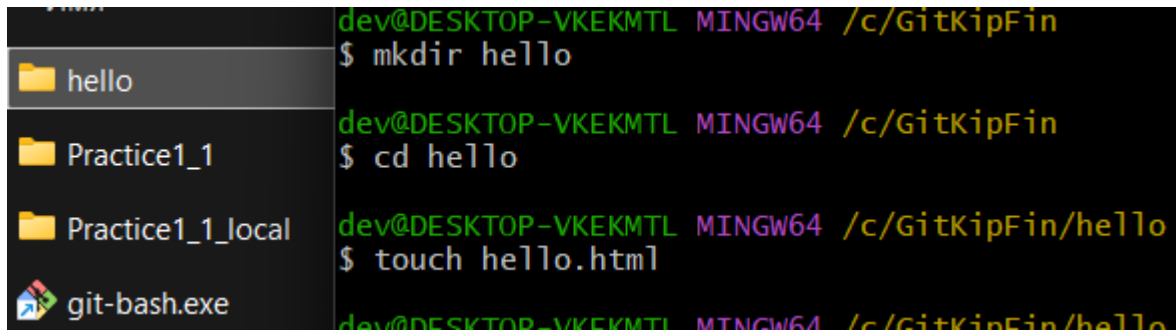


```
ev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
git config --global core.quotePath off

ev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
```

Рисунок 3 - Установка отображения unicode

Создание проекта



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ mkdir hello

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ cd hello

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello
$ touch hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello
```

Рисунок 4 - Создание каталога и файла

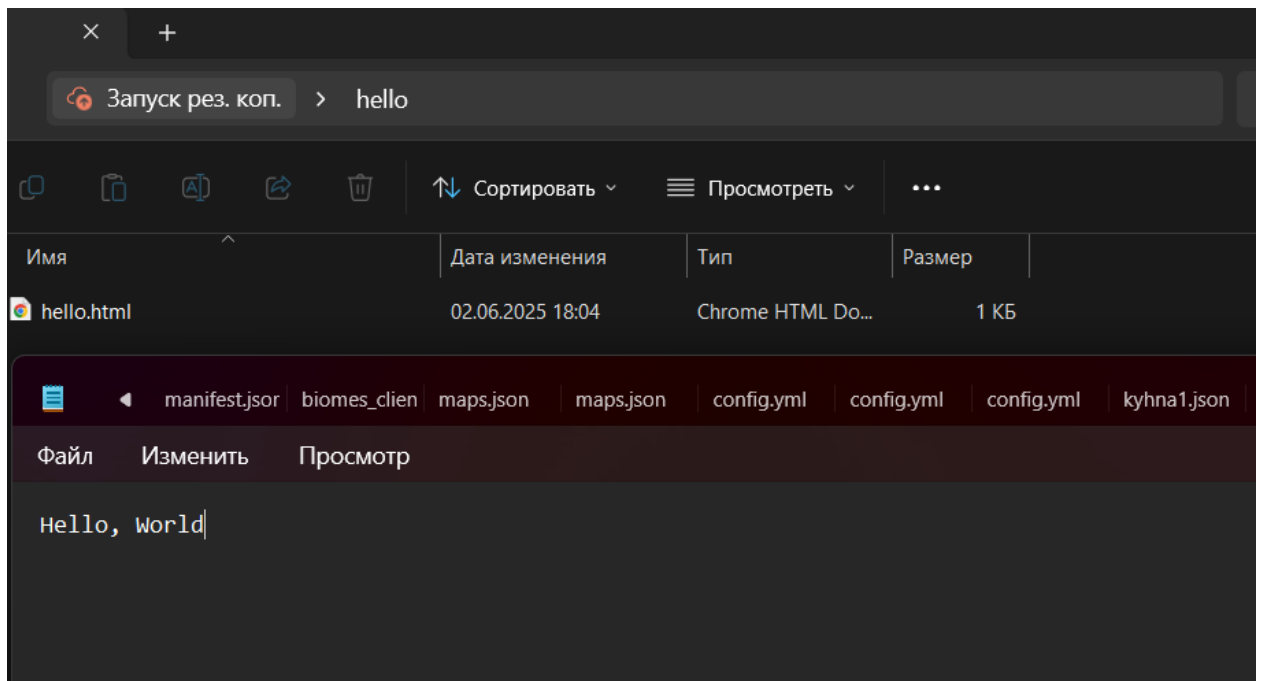
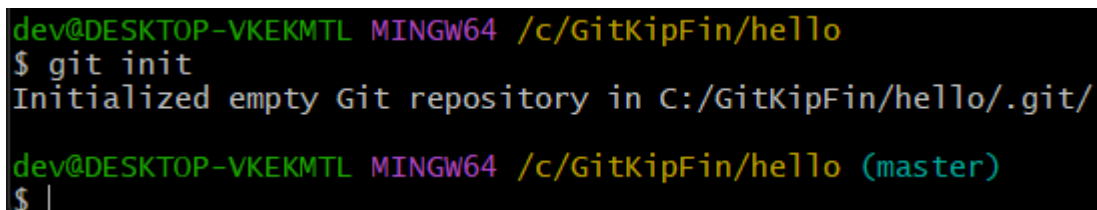


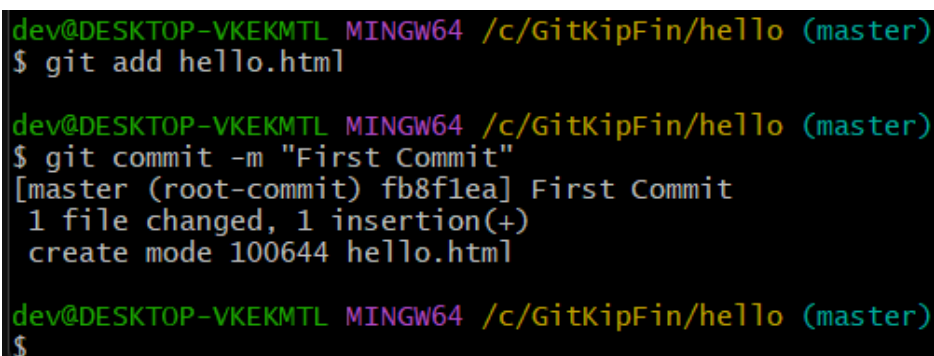
Рисунок 5 - Содержание файла



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello
$ git init
Initialized empty Git repository in C:/GitKipFin/hello/.git/

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 6 - Создание репозитория



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git commit -m "First Commit"
[master (root-commit) fb8f1ea] First Commit
1 file changed, 1 insertion(+)
create mode 100644 hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 7 - Добавление в репозиторий

Проверка состояния

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git status
On branch master
nothing to commit, working tree clean

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 8 - Проверка состояния репозитория

Внесение изменений

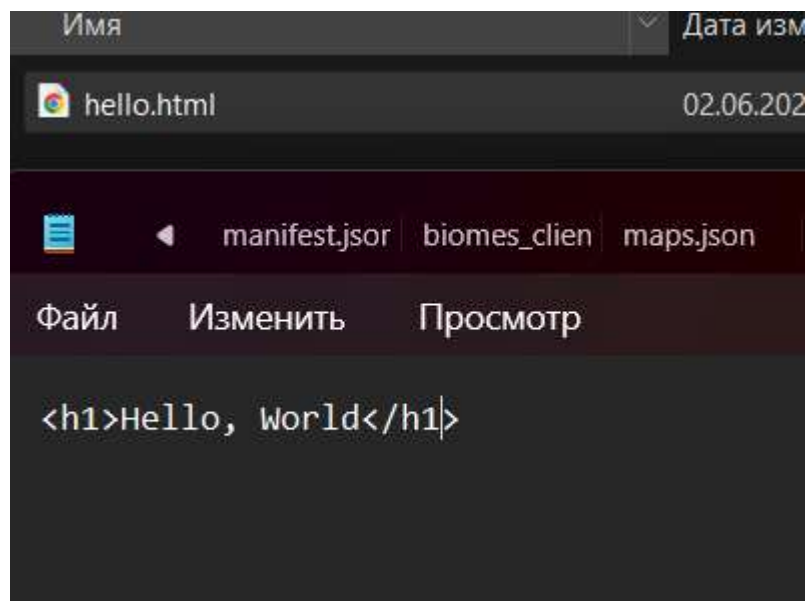


Рисунок 9 - Внесение изменений в файл

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/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")

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

Рисунок 10 - Сообщение о незафиксированных изменениях

Индексация изменений

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 11 - Команды для индексации изменений

Индексация и коммит

Имя	Дата изменения	Тип	Размер
a.html	31.01.2021 10:48	Yandex Browser H...	1 КБ
b.html	31.01.2021 10:49	Yandex Browser H...	1 КБ
hello.html	31.01.2021 10:40	Yandex Browser H...	1 КБ
v.html	31.01.2021 10:49	Yandex Browser H...	1 КБ

Рисунок 12 - Создано 3 файла

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add a.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add b.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git commit -m "Changes for a and b"
[master be84402] Changes for a and b
3 files changed, 3 insertions(+), 1 deletion(-)
create mode 100644 a.html
create mode 100644 b.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 13 - Индексация и коммит для 2 файлов

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add v.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git commit -m "Unrelated change for v"
[master 6a2eeb0] Unrelated change for v
1 file changed, 1 insertion(+)
create mode 100644 v.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 14 - Индексация и коммит для третьего файла

Коммит изменений

```
Home@LAPTOP-TH4RMAU5 MINGW64 /c/D/GitKipFin
$ cd hello

Home@LAPTOP-TH4RMAU5 MINGW64 /c/D/GitKipFin/hello (master)
$ git add .

Home@LAPTOP-TH4RMAU5 MINGW64 /c/D/GitKipFin/hello (master)
$ git commit --m "Hello"
[master 3242edf] Hello
1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 15 - Коммит изменений

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

Рисунок 16 - Ввод комментария

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello/network_hello (master)
$ git commit README
Aborting commit due to empty commit message.
```

Рисунок 17 - Коммит-сообщения

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

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

Рисунок 18 - Проверка состояния

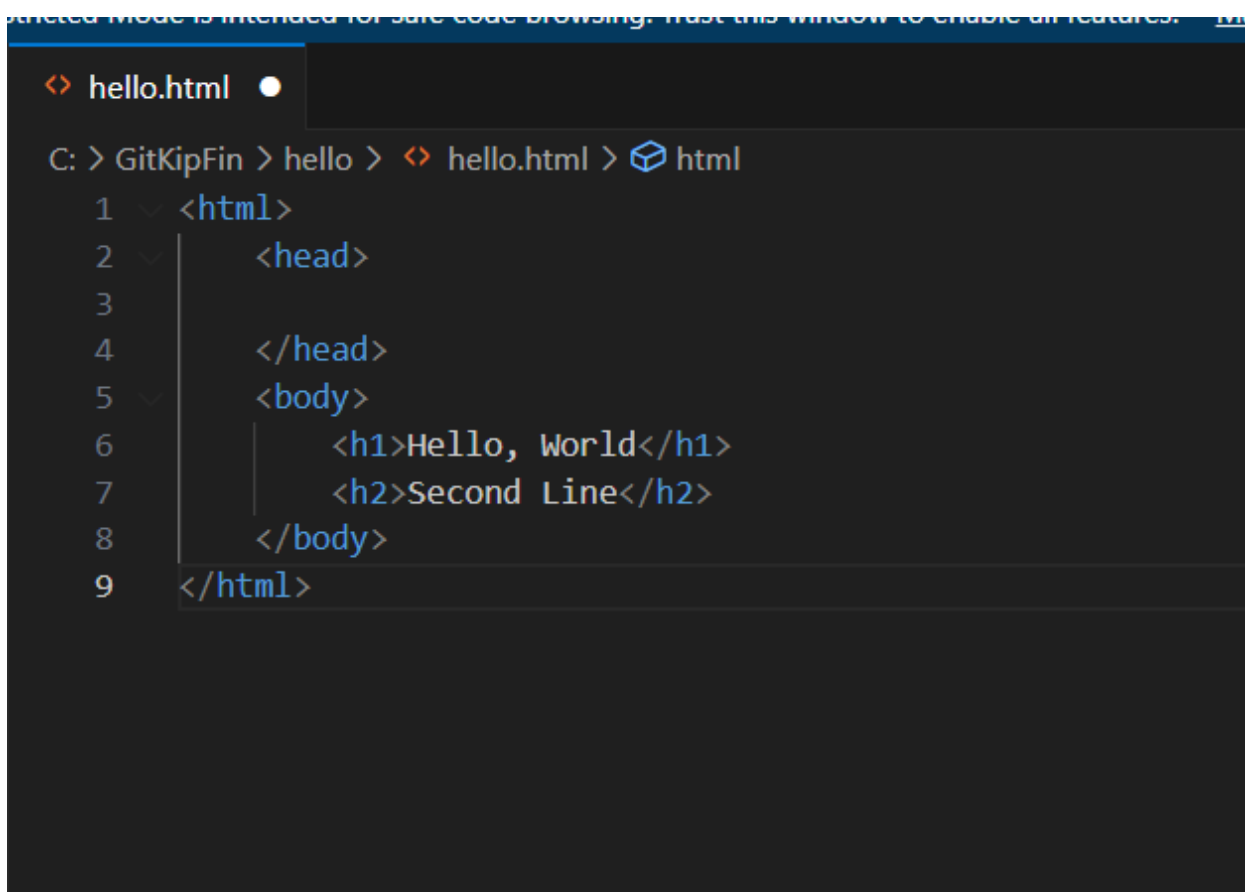
Изменения, а не файлы



The screenshot shows a code editor with a single tab titled 'hello.html'. The breadcrumb navigation at the top reads 'C: > GitKipFin > hello > hello.html > html > body > h2'. The code is as follows:

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

Рисунок 19 - Добавление тегов html и body



The screenshot shows the same code editor with the 'hello.html' tab. The breadcrumb navigation now reads 'C: > GitKipFin > hello > hello.html > html'. The code has been updated to include a head section:

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

Рисунок 20 - Добавление тега head

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/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")

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

```

Рисунок 21 - Проверка состояния

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git commit -m "Add standard HTML tags"
[master 7152c4c] Add standard HTML tags
1 file changed, 6 insertions(+), 1 deletion(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git status
On branch master
nothing to commit, working tree clean

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |

```

Рисунок 22 - Коммит 1 изменения и проверка состояния


```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add .

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git status
On branch master
nothing to commit, working tree clean

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 23 - Индексация 2 изменения и проверка состояния

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git commit -m "Add HTML header"
On branch master
nothing to commit, working tree clean

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 24 - Коммит 2 изменения

Работа с Git

История

```
commit 0cbee5c6309d2d5c0b42ebc04bf3caff79b1a6b5 (HEAD -> master)
Author: Grigoriy Gorodov <gorodov06@icloud.com>
Date: Thu Jun 5 14:00:21 2025 +0300

    Add standard HTML tags

commit 7152c4c66d545476e25cc4009564a4a0ed1e0abf
Author: Grigoriy Gorodov <gorodov06@icloud.com>
Date: Thu Jun 5 13:59:24 2025 +0300

    Add standard HTML tags

commit 6a2eeb0c0f3047a567cf7cac8ce50ed9815639fb
Author: Grigoriy Gorodov <gorodov06@icloud.com>
Date: Wed Jun 4 16:24:44 2025 +0300

    Unrelated change for v

commit be84402eaef435efcb94aaa37c79f2a78a88730a
Author: Grigoriy Gorodov <gorodov06@icloud.com>
Date: Wed Jun 4 16:23:41 2025 +0300

    Changes for a and b
:|
```

Рисунок 25 - Просмотр истории изменений

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git log --pretty=oneline
0cbee5c6309d2d5c0b42ebc04bf3caff79b1a6b5 (HEAD -> master) Add standard HTML ta
7152c4c66d545476e25cc4009564a4a0ed1e0abf Add standard HTML tags
6a2eeb0c0f3047a567cf7cac8ce50ed9815639fb Unrelated change for v
be84402eaeef435efcb94aaa37c79f2a78a88730a Changes for a and b
fb8f1eaa78cb2d569f30717d2db171d6171f54d7 First Commit

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

Рисунок 256 - Однострочный формат вывода

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git log --pretty=oneline --max-count=2
0cbee5c6309d2d5c0b42ebc04bf3caff79b1a6b5 (HEAD -> master) Add standard HTML tags
7152c4c66d545476e25cc4009564a4a0ed1e0abf Add standard HTML tags

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

Рисунок 27 - Вывод последних 2 изменений

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git log --pretty=oneline --since='10 days ago'
0cbee5c6309d2d5c0b42ebc04bf3caff79b1a6b5 (HEAD -> master) Add standard HTML tags
7152c4c66d545476e25cc4009564a4a0ed1e0abf Add standard HTML tags
6a2eeb0c0f3047a567cf7cac8ce50ed9815639fb Unrelated change for v
be84402eaeef435efcb94aaa37c79f2a78a88730a Changes for a and b
fb8f1eaa78cb2d569f30717d2db171d6171f54d7 First Commit

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git log --pretty=oneline --since='12 days ago'
0cbee5c6309d2d5c0b42ebc04bf3caff79b1a6b5 (HEAD -> master) Add standard HTML tags
7152c4c66d545476e25cc4009564a4a0ed1e0abf Add standard HTML tags
6a2eeb0c0f3047a567cf7cac8ce50ed9815639fb Unrelated change for v
be84402eaeef435efcb94aaa37c79f2a78a88730a Changes for a and b
fb8f1eaa78cb2d569f30717d2db171d6171f54d7 First Commit

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 28 - Вывод изменений начиная с определенного времени

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git log --pretty=oneline --until='5 minutes ago'
0cbee5c6309d2d5c0b42ebc04bf3caff79b1a6b5 (HEAD -> master) Add standard HT
7152c4c66d545476e25cc4009564a4a0ed1e0abf Add standard HTML tags
6a2eeb0c0f3047a567cf7cac8ce50ed9815639fb Unrelated change for v
be84402eaeef435efcb94aaa37c79f2a78a88730a Changes for a and b
fb8f1eaa78cb2d569f30717d2db171d6171f54d7 First Commit

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 29 - Вывод изменений до определенного времени

```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git log --pretty=oneline --author='Elena Verstova'
d6533725095ff921e5b35e7bfa4823de3145e85e (HEAD -> master) Add HTML header
cab488a5406ec3818b207577c1b013dd7051d2aa Add standard HTML tags
c8f53f9670de0de4745db8abb3322ea7ae6c4f80 Added h2 tag
cf8645595420fa1f29c2c7410d24acfe76e92eb6 Unrelated change for v
00106a5cbf9e9a51c35dd8c8abb3e45efc8e60c1 Changes for a and b
9234a797ad8f72f2c5b29d884abc254863ab75c8 First Commit
```

Рисунок 30 - Вывод изменений, внесенных определенным автором

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git log --pretty=oneline --author='Grigoriy Gorodov'
0cbee5c6309d2d5c0b42ebc04bf3caff79b1a6b5 (HEAD -> master) Add standard HTML tags
7152c4c66d545476e25cc4009564a4a0ed1e0abf Add standard HTML tags
6a2eeb0c0f3047a567cf7cac8ce50ed9815639fb Unrelated change for v
be84402eaeef435efcb94aaa37c79f2a78a88730a Changes for a and b
fb8f1eaa78cb2d569f30717d2db171d6171f54d7 First Commit

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

Рисунок 31 - Вывод всех изменений

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git log --all --pretty=format:"%h %cd %s (%an)" --since='12 days ago'
0cbee5c Thu Jun 5 14:00:21 2025 +0300 Add standard HTML tags (Grigoriy Gorodov)
7152c4c Thu Jun 5 13:59:24 2025 +0300 Add standard HTML tags (Grigoriy Gorodov)
6a2eeb0 Wed Jun 4 16:24:44 2025 +0300 Unrelated change for v (Grigoriy Gorodov)
be84402 Wed Jun 4 16:23:41 2025 +0300 Changes for a and b (Grigoriy Gorodov)
fb8f1ea Wed Jun 4 16:19:12 2025 +0300 First Commit (Grigoriy Gorodov)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 32 - Использование нескольких параметров

Алиасы

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git config --global alias.co checkout
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git config --global alias.ci commit
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git config --global alias.st commit
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git config --global alias.st status
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git config --global alias.br branch
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git config --global alias.hist "log --pretty=format:'%h %ad | %cd [%an]' --graph --date=short"
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git config --global alias.type 'cat-file -t'
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git config --global alias.dump 'cat-file -p'
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 33 - Настройка алиасов для некоторых команд

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

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git hist
* 0cbee5c 2025-06-05 | Add standard HTML tags (HEAD -> master) [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 34 - Выполнение алиаса hist

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ alias gs='git status'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ gs
On branch master
nothing to commit, working tree clean

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

Рисунок 35 - Установка и выполнение алиаса gs

Получение старых версий

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git hist
* 0cbee5c 2025-06-05 | Add standard HTML tags (HEAD -> master) [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]
```

Рисунок 36 - Просмотр хэшей предыдущих версий

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello ((7152c4c...))
$ git checkout master
Previous HEAD position was 7152c4c Add standard HTML tags
Switched to branch 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World</h1>
    <h2>Second Line</h2>
  </body>
</html>
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 37 - Возвращение к нужной версии и просмотр содержимого файла

Возвращение к последней версии в ветке master

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git checkout master
Already on 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World</h1>
    <h2>Second Line</h2>
  </body>
</html>
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 38 - Возвращение к последней версии в ветке master

Создание тегов версий

```
</html>
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git tag ver1

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 39 - Задание тега

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/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 7152c4c Add standard HTML tags
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello ((7152c4c...))
$ cat hello.html
<html>
  <body>
    <h1>Hello, World</h1>
    <h2>Second Line</h2>
  </body>
</html>
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello ((7152c4c...))
$ |
```

Рисунок 40 – Переход к предыдущей версии с помощью тега


```

</html>
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello ((cab488a...))
$ git tag ver1-beta

HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello ((ver1-beta))
$

```

Рисунок 41 - Задание тега предыдущей версии

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1)
$ git checkout -b ver1-beta
Switched to a new branch 'ver1-beta'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1-beta)
$ git branch ver1

```

Рисунок 42 - Переключение между версиями с помощью тегов

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1)
$ git checkout -b ver1-beta
Switched to a new branch 'ver1-beta'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1-beta)
$ |

```

Рисунок 43 - Просмотр тегов

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello ((0cbee5c...))
$ git hist master --all
warning: refname 'master' is ambiguous.
* 0cbee5c 2025-06-05 | Add standard HTML tags (HEAD, master) [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags (tag: master) [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello ((0cbee5c...))
$

```

Рисунок 44 - Просмотр тегов в логе

Отмена локальных изменений (до индексации)

```

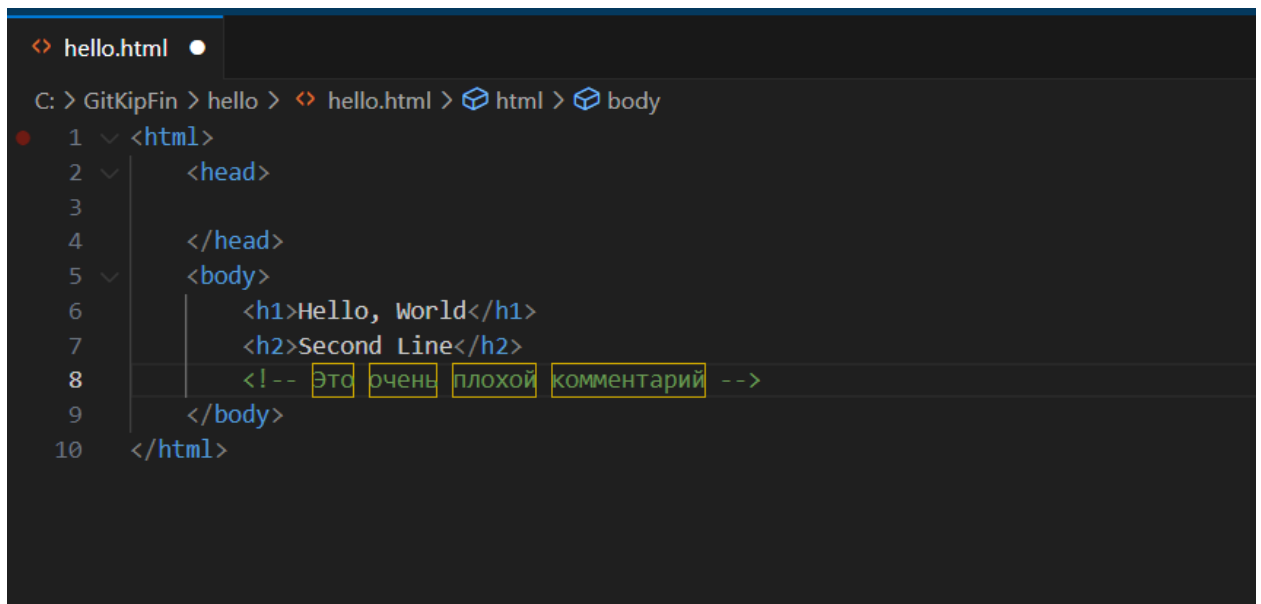
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello ((0cbee5c...))
$ git checkout master
warning: refname 'master' is ambiguous.
Switched to branch 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |

```

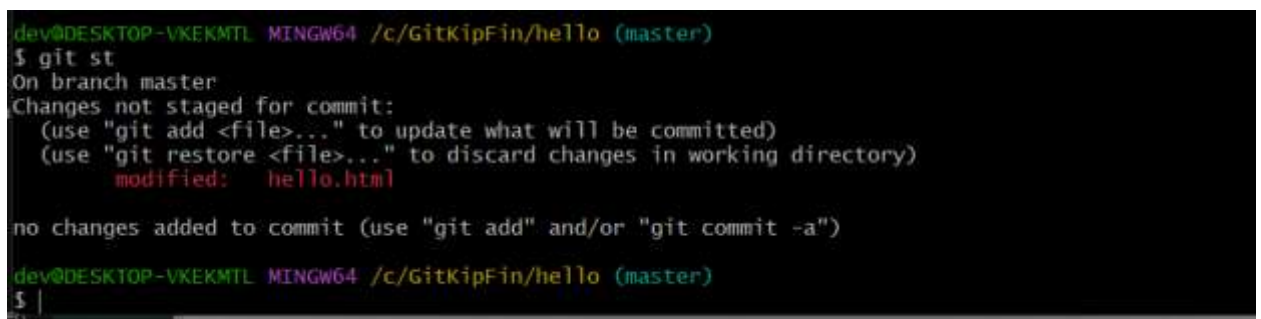
Рисунок 45 - Переключение на последний коммит

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



```
<?xml version="1.0" encoding="UTF-8" ?>
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World</h1>
    <h2>Second Line</h2>
    <!-- Это очень плохой комментарий -->
  </body>
</html>
```

Рисунок 46 - Внесение изменения в файл

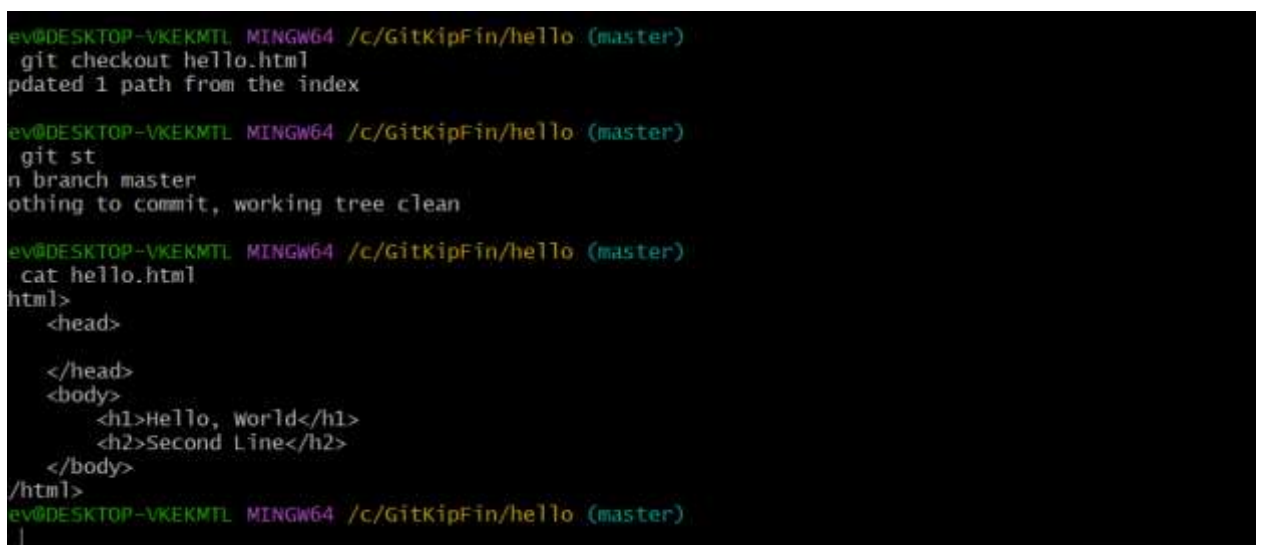


```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git st
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.html

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

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 47 - Не проиндексированное изменение



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git checkout hello.html
Updated 1 path from the index

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git st
On branch master
nothing to commit, working tree clean

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World</h1>
    <h2>Second Line</h2>
  </body>
</html>

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 268 - Возвращение к версии

Отмена проиндексированных изменений (перед коммитом)

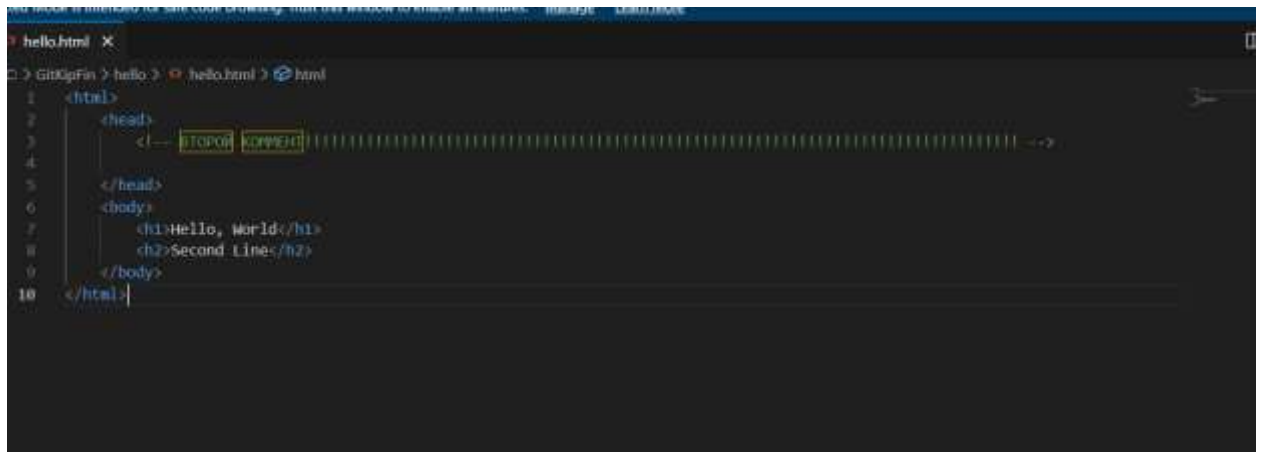


Рисунок 49 - Внесение ненужного изменения

```
</html>
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git st
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 50 - Индексация изменения

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git reset HEAD hello.html
Unstaged changes after reset:
  hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/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")

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 51 - Очистка буферной зоны

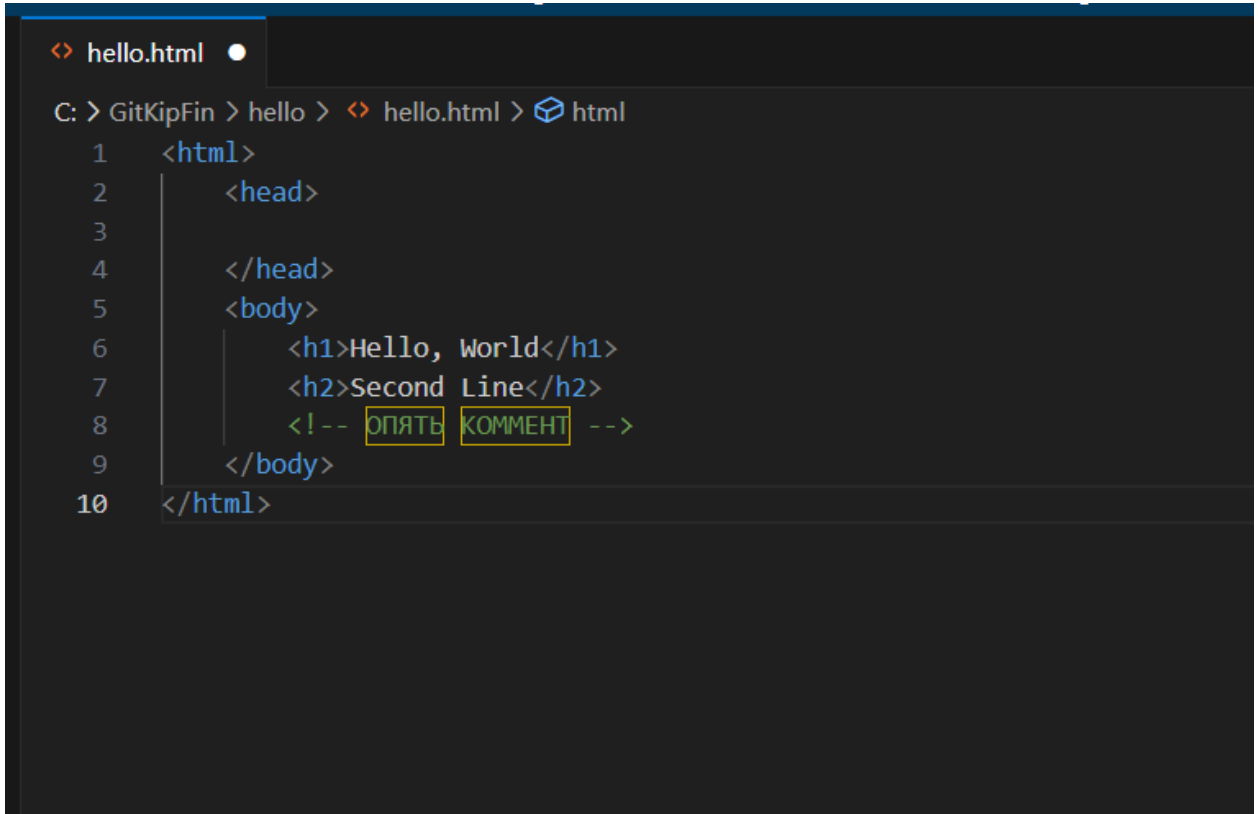
```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git checkout hello.html
Updated 1 path from the index

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git st
On branch master
nothing to commit, working tree clean

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

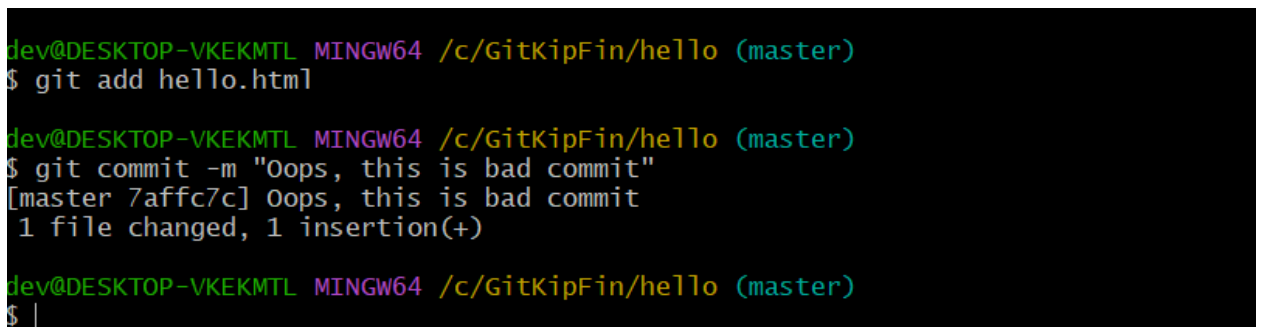
Рисунок 52 - Удаление ненужных изменений

Отмена коммитов



```
<> hello.html ●
C: > GitKipFin > hello > <> hello.html > html
1  <html>
2      <head>
3
4      </head>
5      <body>
6          <h1>Hello, World</h1>
7          <h2>Second Line</h2>
8          <!-- ОПЯТЬ КОММЕНТ -->
9      </body>
10 </html>
```

Рисунок 53 - Внесение изменения в файл

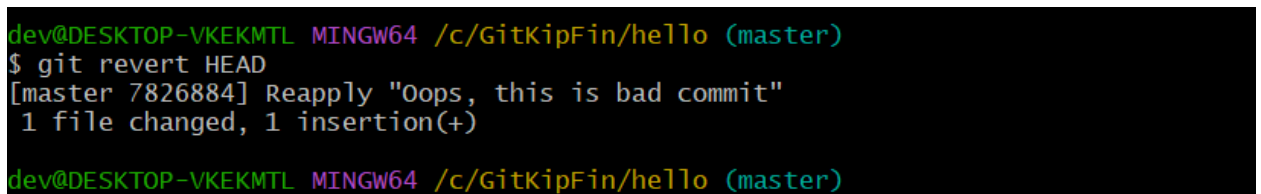


```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git commit -m "Oops, this is bad commit"
[master 7affc7c] Oops, this is bad commit
1 file changed, 1 insertion(+)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

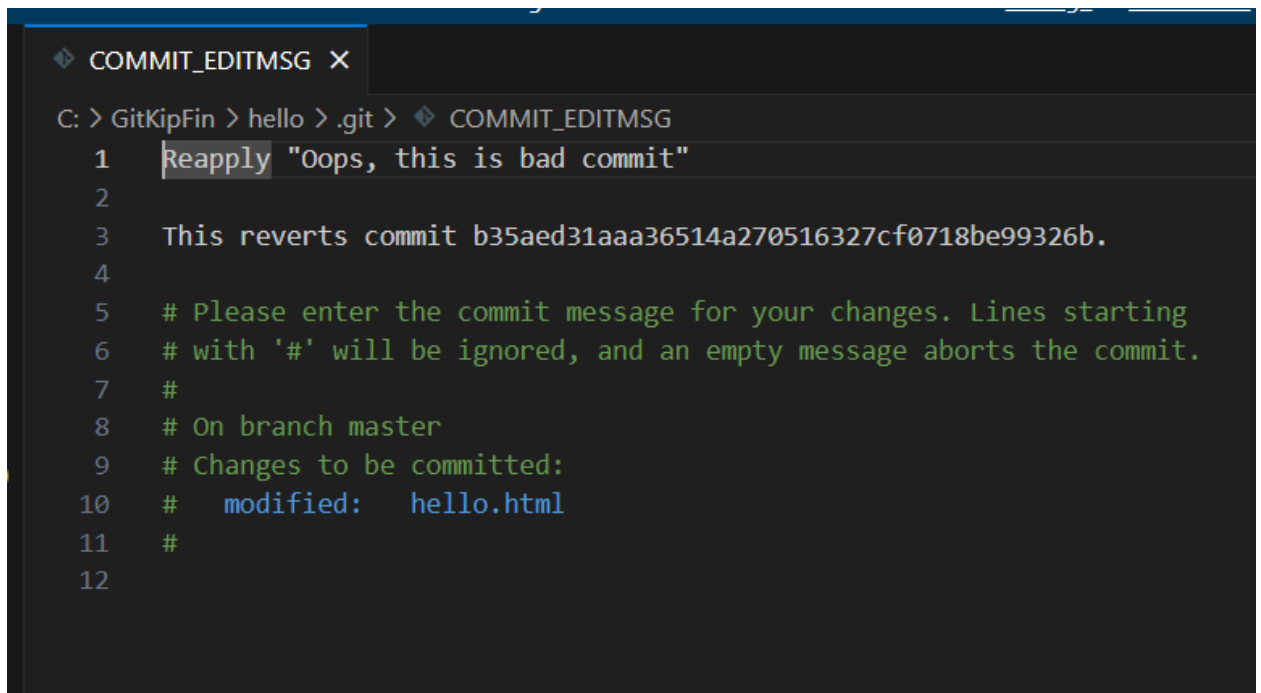
Рисунок 54 - Индексация и коммит



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git revert HEAD
[master 7826884] Reapply "Oops, this is bad commit"
1 file changed, 1 insertion(+)

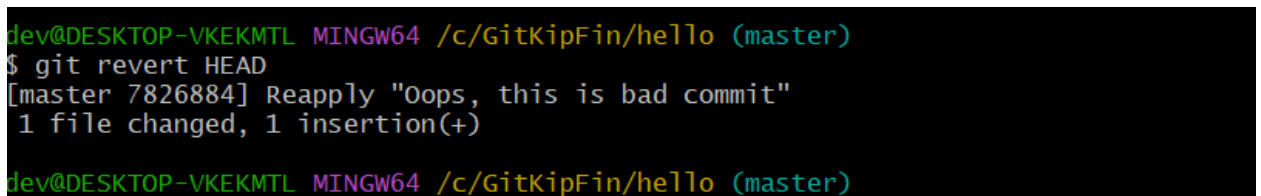
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

Рисунок 55 - Выполнение команды git revert



```
COMMIT_EDITMSG X
C: > GitKipFin > hello > .git > COMMIT_EDITMSG
1  Reapply "Oops, this is bad commit"
2
3  This reverts commit b35aed31aaa36514a270516327cf0718be99326b.
4
5  # Please enter the commit message for your changes. Lines starting
6  # with '#' will be ignored, and an empty message aborts the commit.
7  #
8  # On branch master
9  # Changes to be committed:
10 #   modified:   hello.html
11 #
12
```

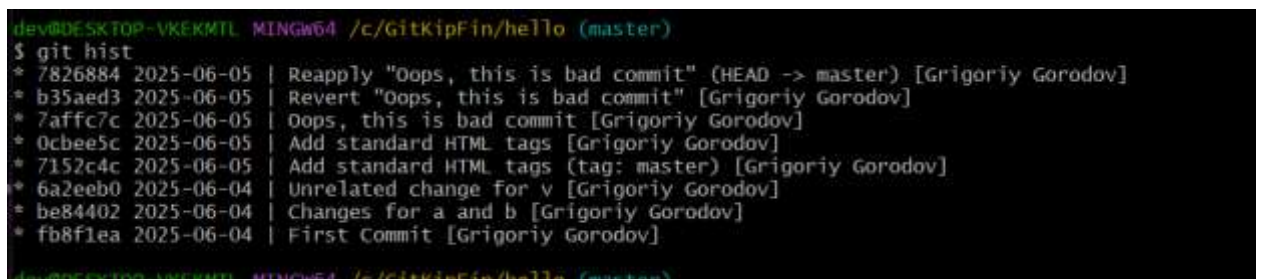
Рисунок 56 - Коммит сообщение в редакторе



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git revert HEAD
[master 7826884] Reapply "Oops, this is bad commit"
1 file changed, 1 insertion(+)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

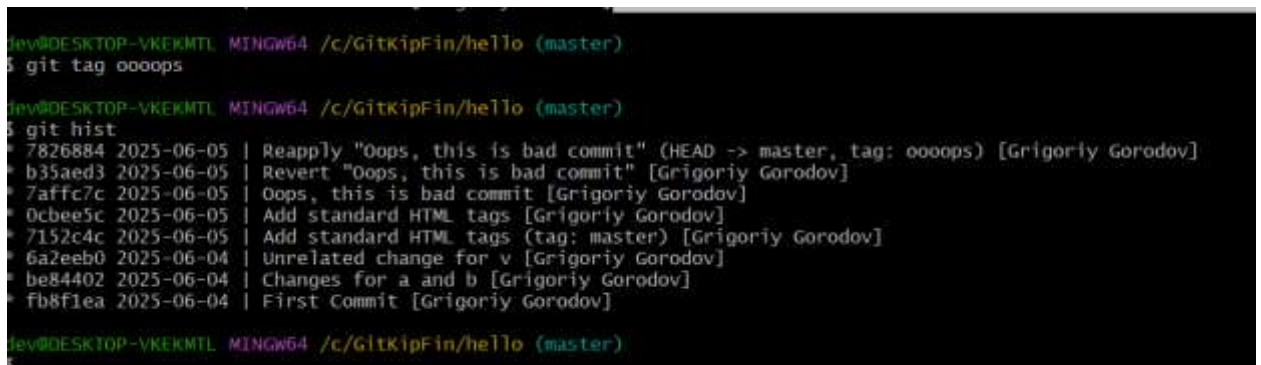
Рисунок 57 - Редактор закрыт



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git hist
* 7826884 2025-06-05 | Reapply "Oops, this is bad commit" (HEAD -> master) [Grigoriy Gorodov]
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags (tag: master) [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

Рисунок 58 - Все коммиты при просмотре лога



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git tag oooops

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git hist
* 7826884 2025-06-05 | Reapply "Oops, this is bad commit" (HEAD -> master, tag: oooops) [Grigoriy Gorodov]
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags (tag: master) [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

Рисунок 59 - Отметка тегом коммита

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1)
$ git reset --hard ver1
HEAD is now at 7826884 Reapply "Oops, this is bad commit"

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1)
$ git hist
* 7826884 2025-06-05 | Reapply "Oops, this is bad commit" (HEAD -> ver1, tag: oooops, ver1-beta, master) [Grigoriy Gorodov]
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags (tag: master) [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1)

```

Рисунок 60 - Сброс коммита

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1)
$ git hist --all
* 7826884 2025-06-05 | Reapply "Oops, this is bad commit" (HEAD -> ver1, tag: oooops, ver1-beta, master) [Grigoriy Gorodov]
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags (tag: master) [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1)

```

Рисунок 61 - Сброшенные коммиты находятся по-прежнему в репозитории

Удаление тега

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1)
$ git tag -d oooops
Deleted tag 'oooops' (was 7826884)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1)
$ git hist --all
* 7826884 2025-06-05 | Reapply "Oops, this is bad commit" (HEAD -> ver1, ver1-beta, master) [Grigoriy Gorodov]
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags (tag: master) [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

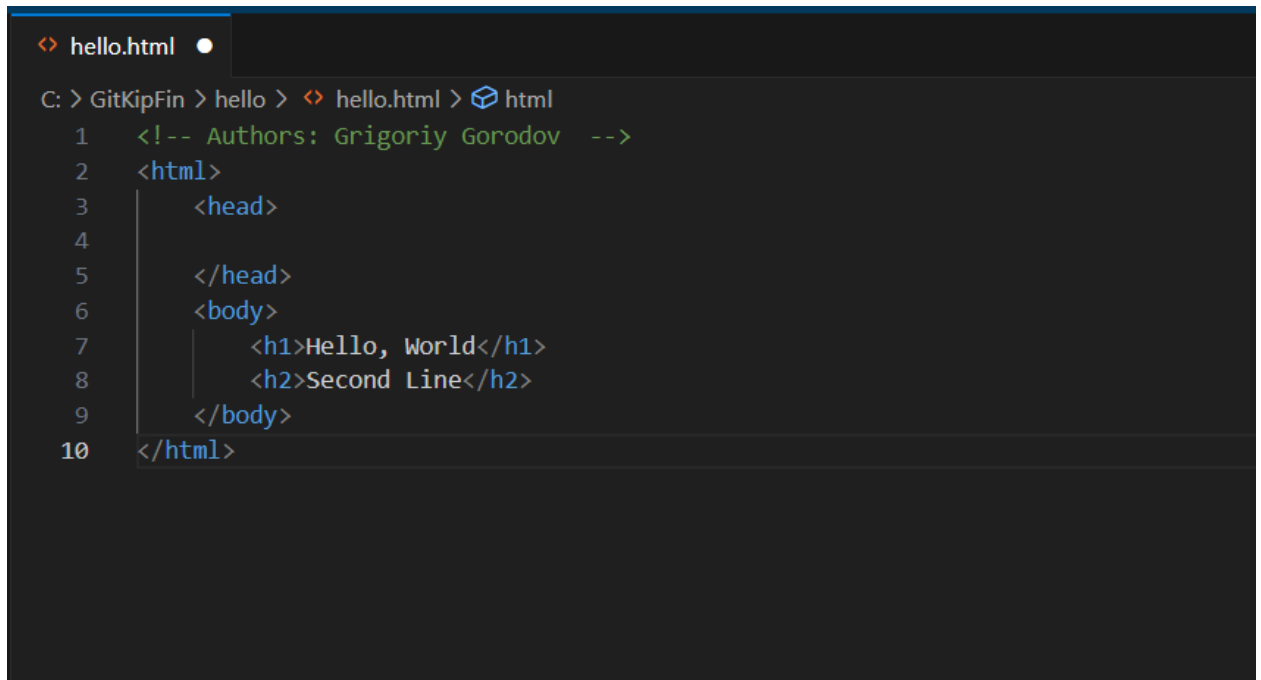
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1)
$

```

Рисунок 62 - Удаление тега

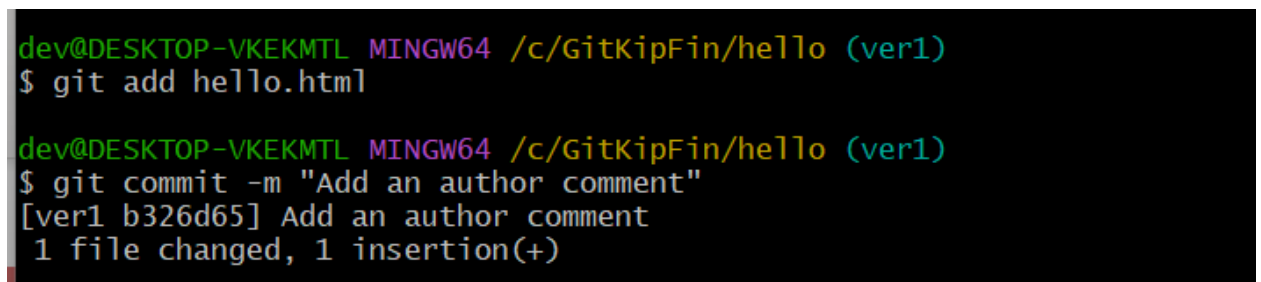
Работа с Git

Внесение изменений в коммиты



```
<> hello.html ●
C: > GitKipFin > hello > <> hello.html > html
1  <!-- Authors: Grigoriy Gorodov -->
2  <html>
3      <head>
4
5      </head>
6      <body>
7          <h1>Hello, World</h1>
8          <h2>Second Line</h2>
9      </body>
10 </html>
```

Рисунок 63 - Добавление комментария в файл



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1)
$ git add hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (ver1)
$ git commit -m "Add an author comment"
[ver1 b326d65] Add an author comment
1 file changed, 1 insertion(+)
```

Рисунок 64 - Индексация и коммит



```
<> hello.html ●
C: > GitKipFin > hello > <> hello.html > html > body > h2
1  <!-- Authors: Grigoriy Gorodov (gorodov06@icloud.com) -->
2  <html>
3      <head>
4
5      </head>
6      <body>
7          <h1>Hello, World</h1>
8          <h2>Second Line</h2>
9      </body>
10 </html>
```

Рисунок 65 - Добавление электронной почты

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git commit --amend -m "Add an author/email comment"
[master 7726709] Add an author/email comment
Date: Thu Jun 5 17:12:55 2025 +0300
1 file changed, 1 insertion(+)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 66 - Индексация и изменение коммита

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git hist
* 7726709 2025-06-05 | Add an author/email comment (HEAD -> master) [Grigoriy Gorodov]
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags (tag: master) [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 67 - Последний коммит изменен

Перемещение файлов

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ mkdir lib

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git mv hello.html lib

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    hello.html -> lib/hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 278 - Перемещение файла

```
ev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git commit -m "Moved hello.html to lib"
master 149a10a] Moved hello.html to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)

ev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

Рисунок 69 – Коммит перемещения

Подробнее о структуре

```
index.html X
C:\> GitKipFin > hello > index.html > html > body > iframe
1 <html>
2   <body>
3     <iframe src="lib/hello.html" width="200" height="200"></iframe>
4   </body>
5
6 </html>
```

Рисунок 70 - Содержимое файла index.html

Далее нужно проиндексировать и закоммитить файл (рисунок 9).

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add index.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git commit -m "Added index.html"
[master d74408c] Added index.html
1 file changed, 6 insertions(+)
create mode 100644 index.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 71 - Индексация и коммит

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

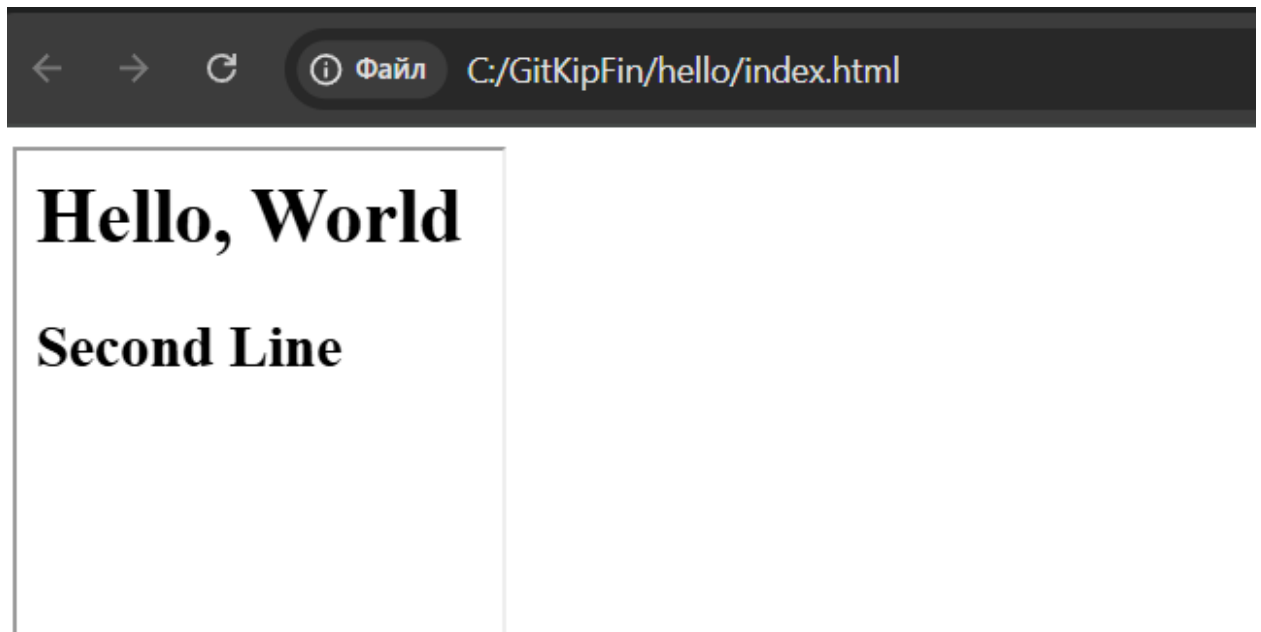


Рисунок 72 - Файл index.html, открытый в браузере

Каталог .git

```
create mode 100644 index.html
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ ls -C .git
COMMIT_EDITMSG HEAD ORIG_HEAD config description hooks/ index info/ logs/ objects/ refs/
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

Рисунок 73 - Содержание каталога .git

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ ls -C .git/objects
0c/ 18/ 2a/ 3c/ 71/ 77/ 7a/ 84/ a2/ ad/ b9/ cb/ d8/ f4/ info/
14/ 23/ 2e/ 6a/ 76/ 78/ 7e/ a1/ ab/ b3/ be/ d7/ ee/ fb/ pack/
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

Рисунок 74 - Содержание каталога objects

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ ls -C .git/objects
0c/ 18/ 2a/ 3c/ 71/ 77/ 7a/ 84/ a2/ ad/ b9/ cb/ d8/ f4/ info/
14/ 23/ 2e/ 6a/ 76/ 78/ 7e/ a1/ ab/ b3/ be/ d7/ ee/ fb/ pack/

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ ls -C .git/objects/8d
ls: cannot access '.git/objects/8d': No such file or directory

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ ls -C .git/objects/7a
fc7cf6538fa8ee84ddc9275f8b7deedb493a2

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

Рисунок 75 - Просмотр каталога 8d

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 76 - Просмотр файла конфигурации


```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ ls .git/refs
heads/  tags/

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ ls .git/refs/heads
master  ver1  ver1-beta

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ ls .git/refs/tags
master  ver1

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ cat .git/refs/tags/ver1
d74408cb64d415bb4f04c95459b3538cb6fd6a66

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 77 - Просмотр файлов и веток

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ cat .git/HEAD
ref: refs/heads/master

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 78 - Содержимое файла HEAD

Работа с объектами git

Для начала необходимо просмотреть последний коммит (рисунок 17).

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git hist --max-count=1
d74408c 2025-06-05 | Added index.html (HEAD -> master, tag: ver1) [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 79 - Последний коммит

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git cat-file -t d74408c
commit

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git cat-file -p d74408c
tree b31485bf14e9f4a1765bcd0d17db4dcedfb70531
parent 149a10a934125c53710fb32849000c27c166bda8
author Grigoriy Gorodov <gorodov06@icloud.com> 1749138605 +0300
committer Grigoriy Gorodov <gorodov06@icloud.com> 1749138605 +0300

Added index.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 280 - Просмотр объекта коммита


```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git type d74408c
commit

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git dump d74408c
tree b31485bf14e9f4a1765bcd0d17db4dcedfb70531
parent 149a10a934125c53710fb32849000c27c166bda8
author Grigoriy Gorodov <gorodov06@icloud.com> 1749138605 +0300
committer Grigoriy Gorodov <gorodov06@icloud.com> 1749138605 +0300

Added index.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 81 - Использование алиасов

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git cat-file -p b31485b
100644 blob 1856e9be02756984c385482a07e42f42efd5d2f3      a.html
100644 blob 1856e9be02756984c385482a07e42f42efd5d2f3      b.html
100644 blob 76aabb98e5a97343d55002c879b73afcea8d2626      index.html
040000 tree d7c11a5d87dc1d35b2a99f9734d508818dcc620a      lib
100644 blob 1856e9be02756984c385482a07e42f42efd5d2f3      v.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 82 - Просмотр дерева каталогов

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git cat-file -p d7c11a5d87
100644 blob ee756563f08bea1c3045ec7ab15f4a1c347c3c8d      hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 83 - Просмотр каталога lib

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git cat-file -p ee756563
<!-- Authors: Grigoriy Gorodov (gorodov06@icloud.com) -->
<html>
  <head>

  </head>
  <body>
    <h1>Hello, world</h1>
    <h2>Second Line</h2>
  </body>
</html>
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 84 - Вывод содержимого файла hello.html

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git hist
* d74408c 2025-06-05 | Added index.html (HEAD -> master, tag: ver1) [Grigoriy Gorodov]
* 149a10a 2025-06-05 | Moved hello.html to lib [Grigoriy Gorodov]
* 7726709 2025-06-05 | Add an author/email comment [Grigoriy Gorodov]
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags (tag: master) [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git cat-file -p fb8f1ea
tree a198ffba84d1d90ed776697f6cd9a8a36f2bca3c
author Grigoriy Gorodov <gorodov06@icloud.com> 1749043152 +0300
committer Grigoriy Gorodov <gorodov06@icloud.com> 1749043152 +0300
First Commit

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git cat-file -p a198ffb
100644 blob 1856e9be02736984c385482a07e42f42efd3d2f3    hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git cat-file -p 1856e9b
Hello, World
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$

```

Рисунок 85 - Просмотр содержимого файла при первом коммите

Работа с Git

Создание ветки

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git checkout -b style
Switched to a new branch 'style'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git status
On branch style
nothing to commit, working tree clean

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$

```

Рисунок 29 - Создание новой ветки style

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

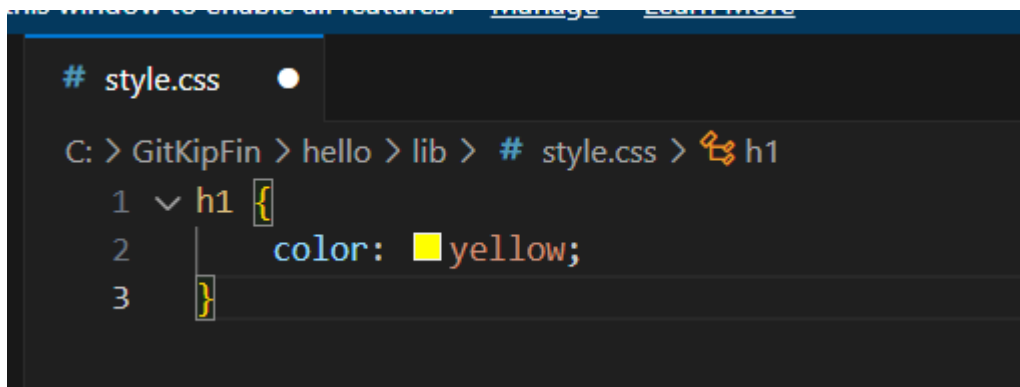
```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ touch lib/style.css

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$

```

Рисунок 30 - Создание файла стилей

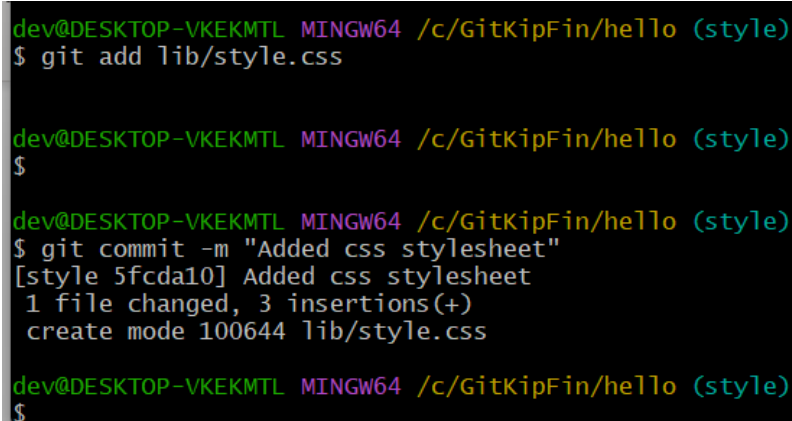


```
# style.css

C: > GitKipFin > hello > lib > # style.css > h1
1  h1 {
2      color: yellow;
3  }
```

Рисунок 31 - Код style.css

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



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git add lib/style.css

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git commit -m "Added css stylesheet"
[style 5fcda10] Added css stylesheet
1 file changed, 3 insertions(+)
create mode 100644 lib/style.css

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$
```

Рисунок 32 - Индексация и коммит нового файла

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



```
hello.html X

C: > GitKipFin > hello > lib > hello.html
1  <!-- Authors: Grigoriy Gorodov (gorodov06@icloud.com) -->
2  <html>
3      <head>
4          <link rel="stylesheet" type="text/css" media="all" href="style.css">
5      </head>
6      <body>
7          <h1>Hello, world</h1>
8          <h2>Second Line</h2>
9      </body>
10 </html>
```

Рисунок 33 - Изменения в файле hello.html

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git add lib/hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git commit -m "Hello.html uses style.css"
[style 248ec37] Hello.html uses style.css
1 file changed, 1 insertion(+), 1 deletion(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$
```

Рисунок 34 - Индексация и коммит

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

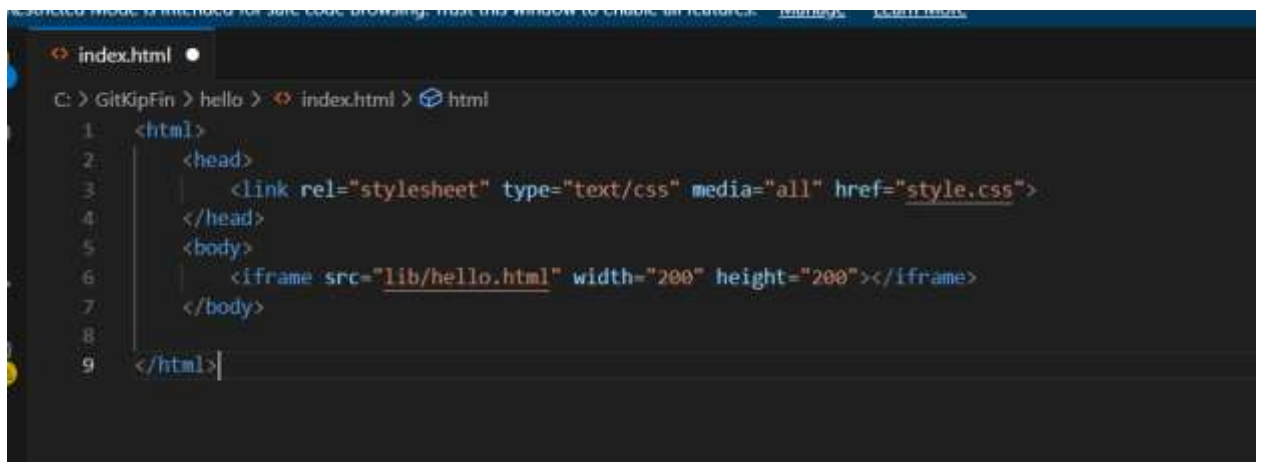


Рисунок 35 - Изменения в файле index.html

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git add index.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git commit -m "Update.html"
[style 0c92b34] Update.html
1 file changed, 3 insertions(+)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$
```

Рисунок 36 - Индексация и коммит

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

Навигация по веткам

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git hist --all
* 0c92b34 2025-06-05 | Update.html (HEAD -> style) [Grigoriy Gorodov]
* 248ec37 2025-06-05 | Hello.html uses style.css [Grigoriy Gorodov]
* 5fcda10 2025-06-05 | Added css stylesheet [Grigoriy Gorodov]
* d74408c 2025-06-05 | Added index.html (tag: ver1, master) [Grigoriy Gorodov]
* 149a10a 2025-06-05 | Moved hello.html to lib [Grigoriy Gorodov]
* 7726709 2025-06-05 | Add an author/email comment [Grigoriy Gorodov]
| * b326d65 2025-06-05 | Add an author comment (ver1) [Grigoriy Gorodov]
| * 7826884 2025-06-05 | Reapply "Oops, this is bad commit" (ver1-beta) [Grigoriy Gorodov]
| /
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags (tag: master) [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$
```

Рисунок 37 - Просмотр истории

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git checkout master
warning: refname 'master' is ambiguous.
Switched to branch 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ cat lib/hello.html
<!-- Authors: Grigoriy Gorodov (gorodov06@icloud.com) -->
<html>
  <head>

  </head>
  <body>
    <h1>Hello, world</h1>
    <h2>Second Line</h2>
  </body>
</html>
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 38 - Переключение на ветку master

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git checkout style
Switched to branch 'style'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ cat lib/hello.html
<!-- Authors: Grigoriy Gorodov (gorodov06@icloud.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>
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ |

```

Рисунок 39 - Переключение на ветку style

Изменения в ветке master

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git checkout master
warning: refname 'master' is ambiguous.
Switched to branch 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$

```

Рисунок 40 - Переключение на ветку master

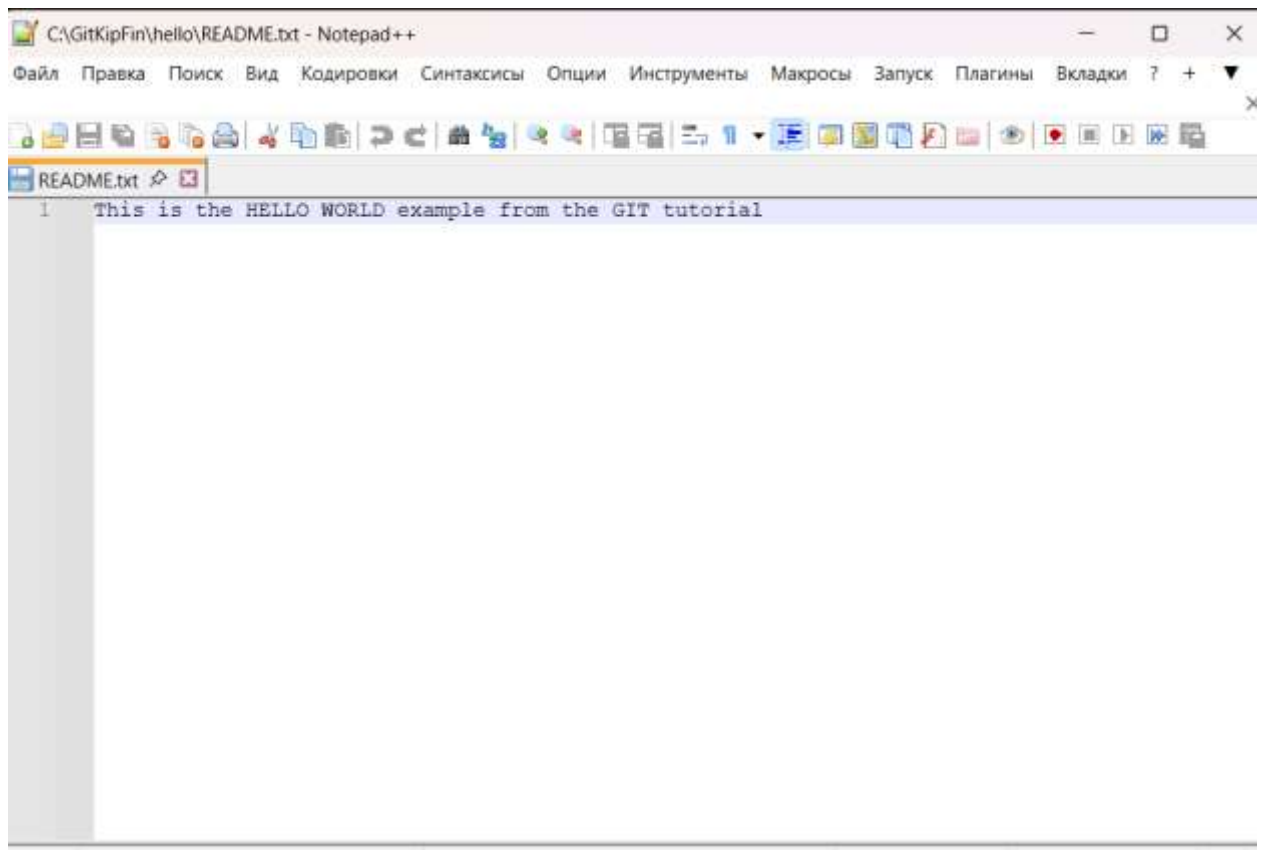


Рисунок 41 - Содержимое файла README

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add README

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git commit -m "Added README"
[master 4dd3ad0] Added README
1 file changed, 1 insertion(+)
create mode 100644 README

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 42 - Индексация и коммит

Просмотр отличающихся веток

На рисунке 15 можно увидеть дерево коммитов.


```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git hist --all
* 4dd3ad0 2025-06-05 | Added README (HEAD -> master) [Grigoriy Gorodov]
* 0c92b34 2025-06-05 | Update.html (style) [Grigoriy Gorodov]
* 248ec37 2025-06-05 | Hello.html uses style.css [Grigoriy Gorodov]
* 5fcd10 2025-06-05 | Added css stylesheet [Grigoriy Gorodov]
//
* d74408c 2025-06-05 | Added index.html (tag: ver1) [Grigoriy Gorodov]
* 149a10a 2025-06-05 | Moved hello.html to lib [Grigoriy Gorodov]
* 7726709 2025-06-05 | Add an author/email comment [Grigoriy Gorodov]
* b326d65 2025-06-05 | Add an author comment (ver1) [Grigoriy Gorodov]
* 7826884 2025-06-05 | Reapply "Oops, this is bad commit" (ver1-beta) [Grigoriy Gorodov]
//
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags (tag: master) [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]
```

Рисунок 43 - Дерево коммитов

Слияние

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git checkout style
Switched to branch 'style'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git merge master
```

Рисунок 44 - Переключение на ветку style

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

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$
```

Рисунок 45 - Слияние с веткой master


```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git hist --all
* e1f6391 2025-06-05 | Merge branch 'master' into style (HEAD -> style) [Grigoriy Gorodov]
|
| * 4dd3ad0 2025-06-05 | Added README (master) [Grigoriy Gorodov]
| * 0c92b34 2025-06-05 | Update.html [Grigoriy Gorodov]
| * 248ec37 2025-06-05 | Hello.html uses style.css [Grigoriy Gorodov]
| * 5fcd10 2025-06-05 | Added css stylesheet [Grigoriy Gorodov]
|
| * d74408c 2025-06-05 | Added index.html (tag: ver1) [Grigoriy Gorodov]
| * 149a10a 2025-06-05 | Moved hello.html to lib [Grigoriy Gorodov]
| * 7726709 2025-06-05 | Add an author/email comment [Grigoriy Gorodov]
| * b326d65 2025-06-05 | Add an author comment (ver1) [Grigoriy Gorodov]
| * 7826884 2025-06-05 | Reapply "Oops, this is bad commit" (ver1-beta) [Grigoriy Gorodov]
|
| * b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
| * 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
| * 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
| * 7152c4c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
| * 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
| * be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
| * fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]
|
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ |

```

Рисунок 46 - Просмотр истории

Создание конфликта

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git checkout master
Switched to branch 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |

```

Рисунок 47 - Переход в ветку master

```

<> hello.html X
C: > GitKipFin > hello > lib > <> hello.html > html
1  <!-- Authors: Grigoriy Gorodov (gorodov06@icloud.com) -->
2  <html>
3      <head>
4
5      </head>
6      <body>
7          <h1>Hello, World, DUUUUUUDEEEEE</h1>
8          <h2>Second Line</h2>
9      </body>
10 </html>

```

Рисунок 48 - Внесение изменений

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add lib/hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git commit -m 'Life if great!'
[master b13246e] Life if great!
1 file changed, 1 insertion(+), 1 deletion(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 49 - Индексация и коммит

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git hist --all
* b13246e 2025-06-05 | Life if great! (HEAD -> master) [Grigoriy Gorodov]
* elf6391 2025-06-05 | Merge branch 'master' into style (style) [Grigoriy Gorodov]
/
/
* 4dd3ad0 2025-06-05 | Added README [Grigoriy Gorodov]
* 0c92b34 2025-06-05 | Update.html [Grigoriy Gorodov]
* 248ec37 2025-06-05 | Hello.html uses style.css [Grigoriy Gorodov]
* 5feda10 2025-06-05 | Added css stylesheet [Grigoriy Gorodov]
/
* d74408c 2025-06-05 | Added index.html (tag: ver1) [Grigoriy Gorodov]
* 149a18a 2025-06-05 | Moved hello.html to lib [Grigoriy Gorodov]
* 7726709 2025-06-05 | Add an author/email comment [Grigoriy Gorodov]
* b326d65 2025-06-05 | Add an author comment (ver1) [Grigoriy Gorodov]
* 7826884 2025-06-05 | Reapply "Oops, this is bad commit" (ver1-beta) [Grigoriy Gorodov]
/
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 50 - Конфликт изменений

Разрешение конфликтов

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git checkout style
Switched to branch 'style'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git merge master
Auto-merging lib/hello.html
Merge made by the 'ort' strategy.
 lib/hello.html | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$
```

Рисунок 51 - Ошибка при слиянии

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

```
<html>
  <head>
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<< HEAD (Current Change)
    <link rel="stylesheet" type="text/css" media="all" href="style.css">
    <!-- no stylw -->
>>>>>> master (Incoming Change)
  </head>
```

Рисунок 52 - Просмотр файла hello.html при наличии конфликта

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

```
hello.html X
C:\GitKipFin> hello > lib > hello.html > ...
1  <!-- Authors: Grigoriy Gorodov (gorodov@icloud.com) -->
2  <html>
3    <head>
4      <link rel="stylesheet" type="text/css" media="all" href="style.css">
5    </head>
6    <body>
7      <h1>Hello, world, DUUUUUUUDEEEEE</h1>
8      <h2>Second Line</h2>
9    </body>
10 </html>
```

Рисунок 53 - Решение конфликта вручную

Затем следует произвести индексацию и коммит (рисунок 26).

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git add lib/hello.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git commit -m "Merged master fixed conflict."
[style 4394663] Merged master fixed conflict.
1 file changed, 1 insertion(+), 1 deletion(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$
```

Рисунок 54 - Индексация и коммит

Работа с Git

Сброс ветки style

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git checkout style
Already on 'style'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git hist
* 4394663 2025-06-05 | Merged master fixed conflict. (HEAD -> style) [Grigoriy Gorodov]
* 62b5d77 2025-06-05 | Merge branch 'master' into style [Grigoriy Gorodov]
* b13246e 2025-06-05 | Life if great! (master) [Grigoriy Gorodov]
* e1f6391 2025-06-05 | Merge branch 'master' into style [Grigoriy Gorodov]
* 4dd3ad0 2025-06-05 | Added README [Grigoriy Gorodov]
* 0c92b34 2025-06-05 | Update.html [Grigoriy Gorodov]
* 248ec37 2025-06-05 | Hello.html uses style.css [Grigoriy Gorodov]
* 5fcdad0 2025-06-05 | Added css stylesheet [Grigoriy Gorodov]
* d74408c 2025-06-05 | Added index.html (tag: ver1) [Grigoriy Gorodov]
* 149a10a 2025-06-05 | Moved hello.html to lib [Grigoriy Gorodov]
* 7726709 2025-06-05 | Add an author/email comment [Grigoriy Gorodov]
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]
$

```

Рисунок 55 - Просмотр истории

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git reset --hard d74408c
HEAD is now at d74408c Added index.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git hist --all
* b13246e 2025-06-05 | Life if great! (master) [Grigoriy Gorodov]
* 4dd3ad0 2025-06-05 | Added README [Grigoriy Gorodov]
* d74408c 2025-06-05 | Added index.html (HEAD -> style, tag: ver1) [Grigoriy Gorodov]
* 149a10a 2025-06-05 | Moved hello.html to lib [Grigoriy Gorodov]
* 7726709 2025-06-05 | Add an author/email comment [Grigoriy Gorodov]
* b326d65 2025-06-05 | Add an author comment (ver1) [Grigoriy Gorodov]
* 7826884 2025-06-05 | Reapply "Oops, this is bad commit" (ver1-beta) [Grigoriy Gorodov]
/
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$

```

Рисунок 56 - Сброс ветки style

Сброс ветки master

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git checkout master
Switched to branch 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git hist
* b13246e 2025-06-05 | Life if great! (HEAD -> master) [Grigoriy Gorodov]
* 4dd3ad0 2025-06-05 | Added README [Grigoriy Gorodov]
* d74408c 2025-06-05 | Added index.html (tag: ver1, style) [Grigoriy Gorodov]
* 149a10a 2025-06-05 | Moved hello.html to lib [Grigoriy Gorodov]
* 7726709 2025-06-05 | Add an author/email comment [Grigoriy Gorodov]
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$

```

Рисунок 57 - Переключение на master и просмотр истории

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git hist --all
* b13246e 2025-06-05 | Life if great! (HEAD -> master) [Grigoriy Gorodov]
* 4dd3ad0 2025-06-05 | Added README [Grigoriy Gorodov]
* d74408c 2025-06-05 | Added index.html (tag: ver1, style) [Grigoriy Gorodov]
* 149a10a 2025-06-05 | Moved hello.html to lib [Grigoriy Gorodov]
* 7726709 2025-06-05 | Add an author/email comment [Grigoriy Gorodov]
* b326d65 2025-06-05 | Add an author comment (ver1) [Grigoriy Gorodov]
* 7826884 2025-06-05 | Reapply "Oops, this is bad commit" (ver1-beta) [Grigoriy Gorodov]
/
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$

```

Рисунок 58 - Сброс ветки master

Перебазирование

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git checkout style
Switched to branch 'style'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git rebase master
Successfully rebased and updated refs/heads/style.

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git hist
* b13246e 2025-06-05 | Life if great! (HEAD -> style, master) [Grigoriy Gorodov]
* 4dd3ad0 2025-06-05 | Added README [Grigoriy Gorodov]
* d74408c 2025-06-05 | Added index.html (tag: ver1) [Grigoriy Gorodov]
* 149a10a 2025-06-05 | Moved hello.html to lib [Grigoriy Gorodov]
* 7726709 2025-06-05 | Add an author/email comment [Grigoriy Gorodov]
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$

```

Рисунок 59 - Перебазирование веток

Слияние в ветку master

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (style)
$ git checkout master
Switched to branch 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git merge style
Already up to date.

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 60 - Слияние веток

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git hist
* b13246e 2025-06-05 | Life is great! (HEAD -> master, style) [Grigoriy Gorodov]
* 4dd3ad0 2025-06-05 | Added README. [Grigoriy Gorodov]
* d74408c 2025-06-05 | Added index.html (tag: ver1) [Grigoriy Gorodov]
* 149a10a 2025-06-05 | Moved hello.html to lib [Grigoriy Gorodov]
* 7726709 2025-06-05 | Add an author/email comment [Grigoriy Gorodov]
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 61 - Просмотр истории

Клонирование репозитория

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ cd

dev@DESKTOP-VKEKMTL MINGW64 ~
$ pwd
/c/Users/dev

dev@DESKTOP-VKEKMTL MINGW64 ~
$ cd C:\GitKipFin

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ git clone hello cloned_hello
Cloning into 'cloned_hello'...
done.

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ ls
Practice1/ Practice1_local/ cloned_hello/ git-bash.exe lnk* hello/

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$
```

Рисунок 62 - Переход в рабочий каталог и его клонирование

Просмотр клонированного репозитория

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ cd cloned_hello

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ ls
README a.html b.html index.html lib/ v.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
```

Рисунок 63 - Просмотр содержимого клонированного репозитория

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git hist --all
* b13246e 2025-06-05 | Life if great! (HEAD -> master, origin/style, origin/master, origin/HEAD) [Grigoriy Gorodov]
* 4dd3ad0 2025-06-05 | Added README [Grigoriy Gorodov]
* d74408c 2025-06-05 | Added index.html (tag: ver1) [Grigoriy Gorodov]
* 149a10a 2025-06-05 | Moved hello.html to lib [Grigoriy Gorodov]
* 7726709 2025-06-05 | Add an author/email comment [Grigoriy Gorodov]
* b326d65 2025-06-05 | Add an author comment (origin/ver1) [Grigoriy Gorodov]
* 7826884 2025-06-05 | Reapply "Oops, this is bad commit" (origin/ver1-beta) [Grigoriy Gorodov]
/
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
```

Рисунок 64 - Просмотр логов клонированного каталога

Origin

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git remote
origin

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git remote show origi
fatal: 'origi' does not appear to be a git repository
fatal: Could not read from remote repository.

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

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git remote show origin
* remote origin
  Fetch URL: C:/GitKipFin/hello
  Push URL: C:/GitKipFin/hello
  HEAD branch: master
  Remote branches:
    master tracked
    style tracked
    ver1 tracked
    ver1-beta 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)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$
```

Рисунок 65 - Просмотр данных об origin

Удаленные ветки

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git branch
* master

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/style
remotes/origin/ver1
remotes/origin/ver1-beta

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$
```

Рисунок 66 - Просмотр удаленных веток

Работа с Git

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ cd ../hello

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 67 - Переход в оригинальный репозиторий

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

```
restricted mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

i README x
C: > GitKipFin > hello > i README
1 This is the HELLO WORLD example from the GIT tutorial
2 [changed in original]
```

Рисунок 68 - Изменения в файле README


```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add README

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git commit -m "Changed README in original repo"
[master 159a6c2] Changed README in original repo
1 file changed, 2 insertions(+), 1 deletion(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 69 - Индексация и коммит новых изменений

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ cd ../cloned_hello

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/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), 348 bytes | 24.00 KiB/s, done.
From C:/GitKipFin/hello
   b13246e..159a6c2  master    -> origin/master

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$
```

Рисунок 70 - Извлечение изменений

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git hist --all
* 159a6c2 2025-06-05 | Changed README in original repo (origin/master, origin/HEAD) [Grigoriy Gorodov]
* b13246e 2025-06-05 | Life if great! (HEAD -> master, origin/style) [Grigoriy Gorodov]
* 4dd3ad0 2025-06-05 | Added README [Grigoriy Gorodov]
* d74408c 2025-06-05 | Added index.html (tag: ver1) [Grigoriy Gorodov]
* 149a10a 2025-06-05 | Moved hello.html to lib [Grigoriy Gorodov]
* 7726709 2025-06-05 | Add an author/email comment [Grigoriy Gorodov]
* b326d65 2025-06-05 | Add an author comment (origin/ver1) [Grigoriy Gorodov]
* 7826884 2025-06-05 | Reapply "Oops, this is bad commit" (origin/ver1-beta) [Grigoriy Gorodov]
/
* b35aed3 2025-06-05 | Revert "Oops, this is bad commit" [Grigoriy Gorodov]
* 7affc7c 2025-06-05 | Oops, this is bad commit [Grigoriy Gorodov]
* 0cbee5c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 7152c4c 2025-06-05 | Add standard HTML tags [Grigoriy Gorodov]
* 6a2eeb0 2025-06-04 | Unrelated change for v [Grigoriy Gorodov]
* be84402 2025-06-04 | Changes for a and b [Grigoriy Gorodov]
* fb8f1ea 2025-06-04 | First Commit [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$
```

Рисунок 71 - Просмотр истории

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ cat README
This is the HELLO WORLD example from the GIT tutorial
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$
```

Рисунок 72 - Вывод содержимого файла README

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git merge origin/master
Updating b13246e..159a6c2
Fast-forward
 README | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$
```

Рисунок 73 - Слияние изменений

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ cat README
This is the HELLO WORLD example from the GIT tutorial
(chaged in original)
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$
```

Рисунок 74 - Вывод содержимого файла README

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git pull
Already up to date.
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$
```

Рисунок 75 - Команда git pull

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git branch --track style origin/style
branch 'style' set up to track 'origin/style'.

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git branch -a
* master
  style
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/style
remotes/origin/ver1
remotes/origin/ver1-beta

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git hist --max-count=2
* 139a6c2 2025-06-05 | Changed README in original repo (HEAD -> master, origin/master, origin/HEAD) [Grigoriy Gorodov]
* b13246e 2025-06-05 | Life is great! (origin/style, style) [Grigoriy Gorodov]

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$

```

Рисунок 76 - Добавление локальной ветки

Далее необходимо создать чистый репозиторий (рисунок 11).

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ cd ..

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ git clone --bare hello hello.git
Cloning into bare repository 'hello.git'...
done.

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ ls hello.git
HEAD config description hooks/ info/ objects/ packed-refs refs/

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$

```

Рисунок 77 - Создание чистого репозитория

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ cd hello

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git remote add shared ../hello.git

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$

```

Рисунок 78 - Добавление удаленного репозитория

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

```

① README ●
C: > GitKipFin > hello > ① README
1 This is the HELLO WORLD example from the GIT tutorial
2 [changed in original and pushed to shared]

```

Рисунок 79 - Внесение изменений в файл

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git checkout master
M      README
Already on 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git add README

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git commit -m "Added shared comment to README"
[master d3cf180] Added shared comment to README
1 file changed, 1 insertion(+), 1 deletion(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$

```

Рисунок 80 - Индексация и коммит

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git push shared master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 343 bytes | 343.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
To ../hello.git
    159a6c2..d3cf180  master -> master

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$

```

Рисунок 81 - Отправка изменений в общий репозиторий

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ cd ../cloned_hello

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$

```

Рисунок 82 - Переход в клонированный репозиторий

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git remote add shared ../hello.git

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git branch --track shared master
branch 'shared' set up to track 'master'.

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git pull shared master
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 323 bytes | 26.00 KiB/s, done.
From ../hello
 * branch                master      -> FETCH_HEAD
 * [new branch]          master      -> shared/master
Updating 159a6c2..d3cf180
Fast-forward
 README | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ cat README
This is the HELLO WORLD example from the GIT tutorial
(changed in original and pushed to shared)
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$

```

Рисунок 83 - Команды, извлекающие общие изменения

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

```

(changed in original and pushed to shared)
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/cloned_hello (master)
$ git daemon --verbose --export-all
[11204] Ready to rumble

```

Рисунок 84 - Настройка сервера

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ git clone hello.git network_hello
Cloning into 'network_hello'...
done.

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ cd network_hello

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/network_hello (master)
$ cd ..

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ cd hello

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ cd network_hello

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello/network_hello (master)
$ ls
README  a.html  b.html  index.html  lib/  v.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello/network_hello (master)
$

```

Рисунок 85 - Клонирование проекта

Работа с Git

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ git clone https://github.com/schacon/ticgit
Cloning into 'ticgit'...
remote: Enumerating objects: 1857, done.
remote: Total 1857 (delta 0), reused 0 (delta 0), pack-reused 1857 (from 1)
Receiving objects: 100% (1857/1857), 334.06 KiB | 2.01 MiB/s, done.
Resolving deltas: 100% (837/837), done.

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin
$ cd ticgit

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$ git remote
origin

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$
```

Рисунок 86 – Клонирование репозитория и просмотр удаленных репозиторий

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$
```

Рисунок 87 - Просмотр удаленных репозиторий с ключом `-v`

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


```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$ git remote add pb https://github.com/paulboone/ticgit

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$ git remove -v
git: 'remove' is not a git command. See 'git --help'.

The most similar command is
    remote

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)
pb      https://github.com/paulboone/ticgit (fetch)
pb      https://github.com/paulboone/ticgit (push)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$

```

Рисунок 88 - Добавление удаленного репозитория

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$ git fetch pb
remote: Enumerating objects: 43, done.
remote: Counting objects: 100% (22/22), done.
remote: Total 43 (delta 22), reused 22 (delta 22), pack-reused 21 (from 1)
Unpacking objects: 100% (43/43), 5.99 KiB | 35.00 KiB/s, done.
From https://github.com/paulboone/ticgit
* [new branch]      master    -> pb/master
* [new branch]      ticgit    -> pb/ticgit

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$

```

Рисунок 89 - Использование имени вместо пути

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$ git fetch 'C:/GitKipFin/cloned_hello'
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 40 (delta 16), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (40/40), 3.59 KiB | 20.00 KiB/s, done.
From C:/GitKipFin/cloned_hello
* branch            HEAD      -> FETCH_HEAD

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$ |

```

Рисунок 90 - Получение данных из удаленных проектов

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

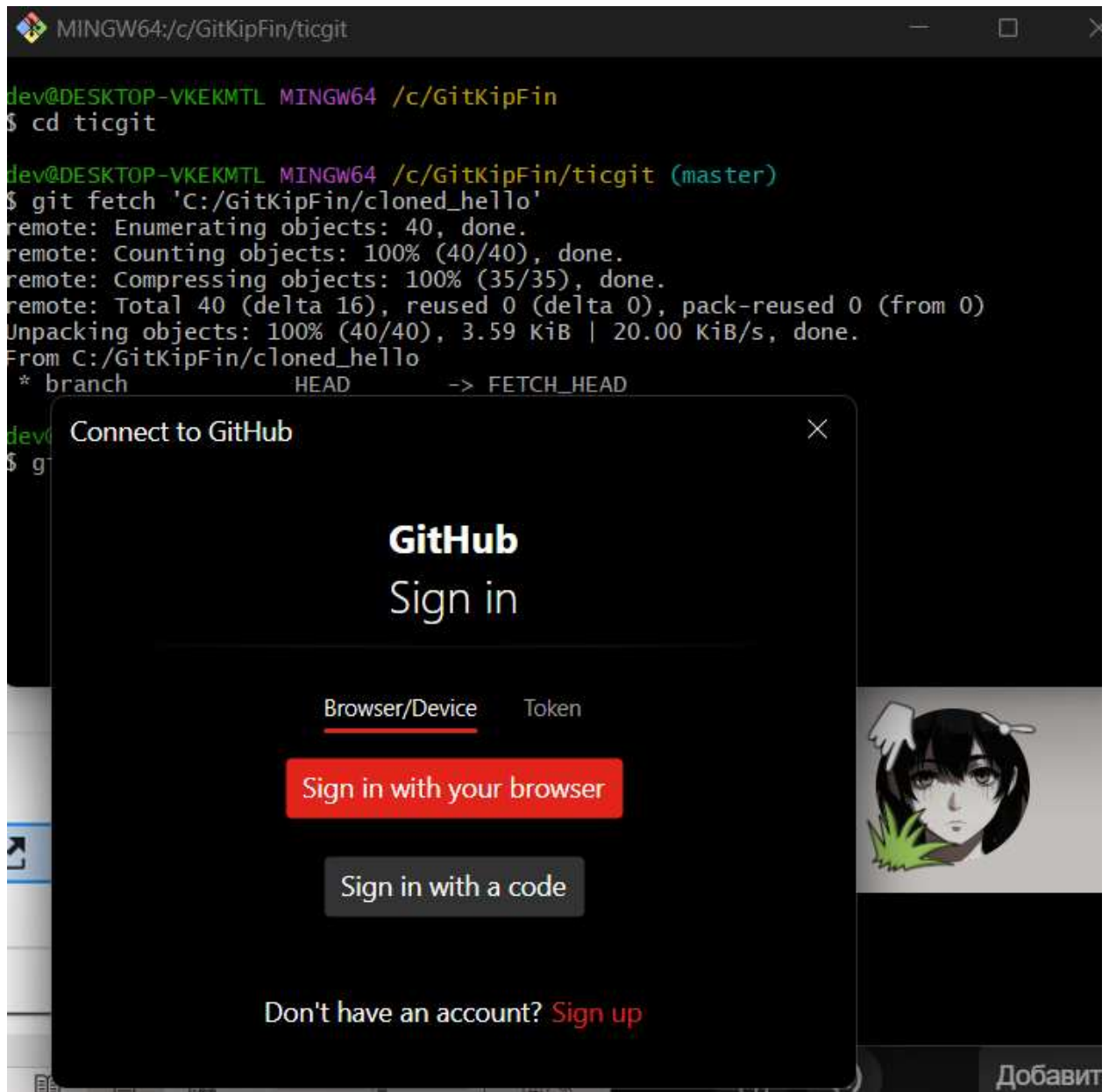


Рисунок 91 - Отправка изменений в удаленный репозиторий

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

```

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

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$ |

```

Рисунок 92 - Информация об удаленном репозитории

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

Рисунок 93 - Переименование удаленного репозитория

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$ git remote rename pb paul
Renaming remote references: 100% (4/4), done.

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$ git remove
git: 'remove' is not a git command. See 'git --help'.

The most similar command is
  remote

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$ git remote
origin
paul

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/ticgit (master)
$ |

```

Рисунок 94 - Удаление удаленного репозитория

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git tag
ver1

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$
```

Рисунок 95 - Просмотр тегов

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git tag
v2.1
ver1

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 96 - Создание аннотированной метки

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

```

[4] vi
#pgm: Grigoriy Gorodtskov:cytoskov66@cloud.com
date: Thu Mar 5 22:48:15 2025 +0300

by version 2.3

commit 81cf18012362a8b6f88998bd11625ef1579d70d (HEAD -- master, tag: v2.3, shared/master)
author: Grigoriy Gorodtskov <cytoskov66@cloud.com>
date: Thu Mar 5 20:38:35 2025 +0300

    added shared commit to README

diff --git a/README.md b/README.md
index f84e051..ee25266 100644
--- a/README.md
+++ b/README.md
@@ -1,2,3,4 @@
 This is the HELLO WORLD example from the GIT tutorial
 (changed as original)
 (no newline at end of file)
 (changed as original and quoted by shared)
 (no newline at end of file)

```

Рисунок 97 - Просмотр данных тега

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git tag v2.1-lw

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git tag
v2.1
v2.1-lw
ver1

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |

```

Рисунок 98 - Создание легковесной метки

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git show v2.1-lw
commit d3cf180332b7a08c81099dbd211625ef3529d70d (HEAD -> master, tag: v2.1-lw, tag: v2.1, shared/master)
Author: Grigoriy Gorodov <gorodov06@icloud.com>
Date: Thu Jun 5 20:30:35 2025 +0300

    Added shared comment to README

diff --git a/README b/README
index f84e055..ee05266 100644
--- a/README
+++ b/README
@@ -1,2 +1,2 @@
 This is the HELLO WORLD example from the GIT tutorial
-(chaged in original)
 \ No newline at end of file
+(chaged in original and pushed to shared)
 \ No newline at end of file

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$

```

Рисунок 99 - Просмотр данных тега

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git log --pretty=oneline
d3cf180332b7a08c81099dbd211625ef3529d70d (HEAD -> master, tag: v2.1-lw, tag: v2.1, shared/master) Added shared comment to README
159a6c257c77a0eb3509ddb41cb3341f155837 Changed README in original repo
b13246e271373897cfbc24684becfb627d3bd732 (style) Life is great!
4dd3ad09dd3a1bf50e3beae6dac6ab8afffe7 Added README
d74408cb64d415bb4f04c95459b3538cb6fd6a66 (tag: ver1) Added index.html
149a10a934125c53710fb32849000c27c166bda8 Moved hello.html to lib
7726709ac91acf6608e3675b228cacc626d8cfaf Add an author/email comment
b35aed31aaa36514a270516327cf0718be99326b Revert "Oops, this is bad commit"
7affc7cf6538fa8ee84ddc9275f8b7deedb493a2 Oops, this is bad commit
0cbee5c6309d2d5c0b42ebc04bf3caff79b1a6b5 Add standard HTML tags
7152c4c66d545476e25cc4009564a4a0ed1e0abf Add standard HTML tags
6a2eeb0c0f3047a567cf7cac8ce50ed9815639fb Unrelated change for v
be84402eaeef435efcb94aaa37c79f2a78a88730a Changes for a and b
fb8f1eaa78cb2d569f30717d2db171d6171f54d7 First Commit

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$

```

Рисунок 100 - Просмотр истории

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git tag -a ver3 159a6c2 -m "Tag message1"

```

Рисунок 101 - Создание тега определенному коммиту

```

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

```

Рисунок 102 - Ввод сообщения в текстовом редакторе

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git show ver3
tag ver3
Tagger: Grigoriy Gorodov <gorodov06@icloud.com>
Date: Thu Jun 5 23:07:41 2025 +0300

Tag message1

commit 159a6c257c77a0eb3509ddb41cb3341f155837 (tag: ver3)
Author: Grigoriy Gorodov <gorodov06@icloud.com>
Date: Thu Jun 5 20:25:15 2025 +0300

    Changed README in original repo

diff --git a/README b/README
index 6db78ad..f84e055 100644
--- a/README
+++ b/README
@@ -1,2 @@
-This is the HELLO WORLD example from the GIT tutorial
\ No newline at end of file
+This is the HELLO WORLD example from the GIT tutorial
+(changed in original)
\ No newline at end of file

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |

```

Рисунок 103 - Просмотр данных тега

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git push shared v2.1
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 168 bytes | 168.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To ../hello.git
 * [new tag]          v2.1 -> v2.1

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |

```

Рисунок 104 - Отправка тега на удаленный сервер

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

```

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

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |

```

Рисунок 105 - Отправка всех тегов на сервер

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git tag -d v2.1-lw
Deleted tag 'v2.1-lw' (was d3cf180)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
```

Рисунок 106 - Удаление тега

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git push shared :refs/tags/v2.1-lw
To ../hello.git
- [deleted]          v2.1-lw

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ |
```

Рисунок 107 - Удаление тегов с сервера

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


```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/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 d3cf180 Added shared comment to README
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello ((v2.1))
$ |

```

Рисунок 108 - Переключение на метку

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello ((v2.1))
$ git checkout -b version2 v2.1
Switched to a new branch 'version2'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ |

```

Рисунок 109 - Создание новой ветки

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ git config --global alias.co checkout

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ git config --global alias.br branch

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ git config --global alias.ci commit

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ git config --global alias.st status

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ |

```

Рисунок 110 - Задание алиасов

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ git config --global alias.unstage 'reset HEAD --'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ |
```

Рисунок 111 - Создание псевдонима исключения файла из индекса

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ git unstage fileA

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ git reset HEAD -- fileA

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ |
```

Рисунок 112 - Использование созданного псевдонима

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ git config --global alias.last 'log -1 HEAD'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$
```

Рисунок 113 - Создание алиаса для просмотра последнего коммита

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ git last
commit d3cf180332b7a08c81099dbd211625ef3529d70d (HEAD -> version2, tag: v2.1, shared/master, master)
Author: Grigoriy Gorodov <gorodov06@icloud.com>
Date: Thu Jun 5 20:30:35 2025 +0300

    Added shared comment to README

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$
```

Рисунок 114 - Результат работы созданного алиаса

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ git config --global alias.visual '!gitk'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (version2)
$ |
```

Рисунок 115 - Создание псевдонима внешней команды

Работа с Git

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git add README test.rb LICENSE

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git commit -m 'Initial commit of my project'
[master c923143] Initial commit of my project
3 files changed, 3 insertions(+), 2 deletions(-)
create mode 100644 LICENSE
create mode 100644 test.rb

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$
```

Рисунок 1 - Индексация и коммит 3 файлов

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git branch testing

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git checkout testing
```

Рисунок 2 - Создание ветки testing

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git checkout testing
Switched to branch 'testing'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (testing)
$
```

Рисунок 3 - Переключение на ветку testing

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (testing)
$ git add test.rb

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (testing)
$ git commit -m 'New changes'
[testing 04f1387] New changes
1 file changed, 1 insertion(+), 1 deletion(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (testing)
$
```

Рисунок 4 - Индексация и коммит файла test.rb

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (testing)
$ git checkout master
Switched to branch 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ |
```

Рисунок 5 - Переключение на ветку master

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git add test.rb

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git commit -a -m 'Another changes'
[master 0ef0848] Another changes
1 file changed, 1 insertion(+), 1 deletion(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ |
```

Рисунок 6 - Еще индексация и коммит test.rb

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git checkout -b iss53
Switched to a new branch 'iss53'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (iss53)
$
```

Рисунок 7 - Создание и переключение на ветку iss53

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (iss53)
$ git add index.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (iss53)
$ git commit -m 'Added html'
[iss53 61e6df2] Added html
1 file changed, 5 insertions(+), 5 deletions(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (iss53)
$
```

Рисунок 8 - Индексация и коммит файла index.html

Далее нужно переключить ветку на master (рисунок 9).

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (iss53)
$ git checkout master
Switched to branch 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$
```

Рисунок 9 - Переключение на ветку master

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (hotfix)
$ git add index.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (hotfix)
$ git commit -a -m 'Fixed the broken email address'
[hotfix efd193e] Fixed the broken email address
1 file changed, 1 insertion(+), 1 deletion(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (hotfix)
$
```

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (hotfix)
$ git checkout master
Switched to branch 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git merge hotfix
Updating 0ef0848..efd193e
Fast-forward
 index.html | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$
```

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

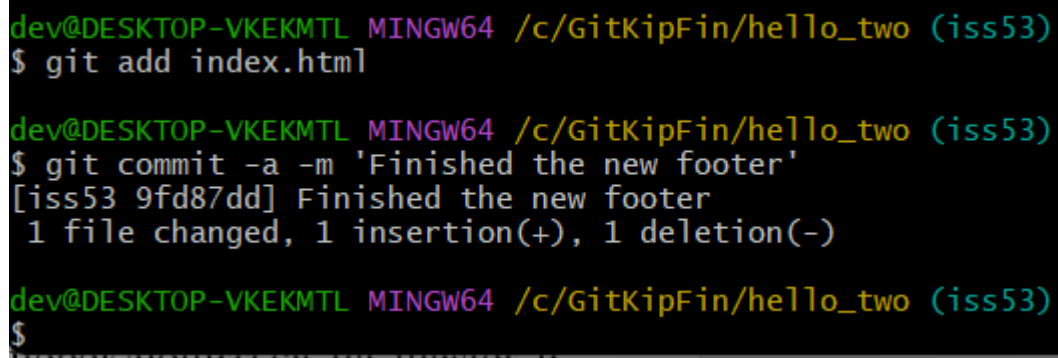
После слияния ветку hotfix можно удалить (рисунок 12).1

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git branch -d hotfix
Deleted branch hotfix (was efd193e).

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$
```

Рисунок 12 - Удаление ветки hotfix

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

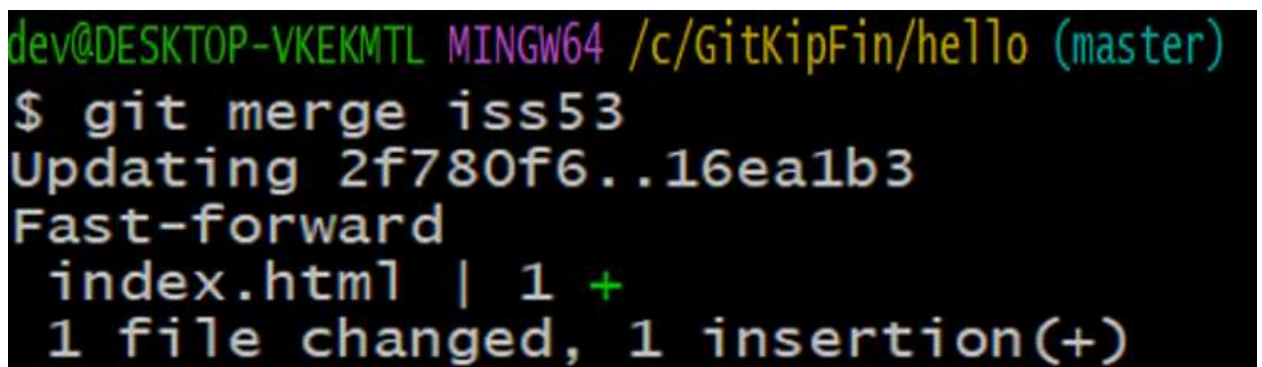


```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (iss53)
$ git add index.html

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (iss53)
$ git commit -a -m 'Finished the new footer'
[iss53 9fd87dd] Finished the new footer
1 file changed, 1 insertion(+), 1 deletion(-)

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (iss53)
$
```

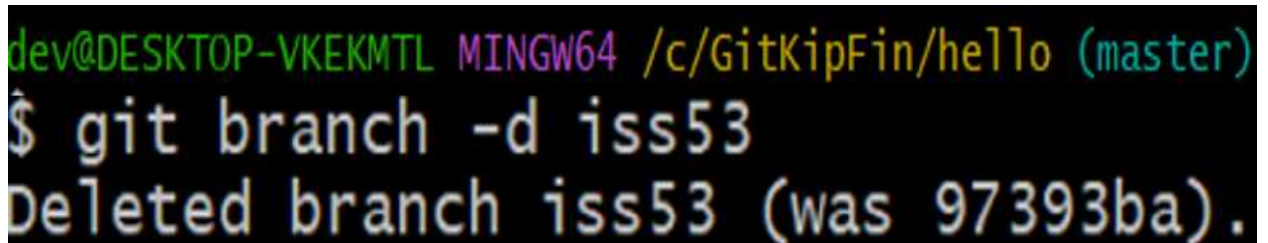
Рисунок 13 - Индексация и коммит index.html на ветке iss53



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git merge iss53
Updating 2f780f6..16ea1b3
Fast-forward
 index.html | 1 +
1 file changed, 1 insertion(+)
```

Рисунок 14 - Слияние веток

После этого ветку iss53 нужно удалить (рисунок 15).



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git branch -d iss53
Deleted branch iss53 (was 97393ba).
```

Рисунок 15 - Удаление ветки iss53

Работа с Git

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git branch
* master
  style
  testing
  ver1
  ver1-beta
  version2
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
```

Рисунок 116 - Список существующих веток

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

```
ev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
git branch -v
master      34bd928 Finish the new Footessr
style       b13246e Life if great!
testing     04f1387 New changes
ver1        b326d65 Add an author comment
ver1-beta   7826884 Reapply "Oops, this is bad commit"
version2    d3cf180 Added shared comment to README
ev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
```

Рисунок 117 - Список веток с последними коммитами

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git branch --merged
* master
  style
  version2

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$
```

Рисунок 118 - Список веток слитых с текущей

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git branch --no-merged
testing
ver1
ver1-beta

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$
```

Рисунок 119 - Список веток не слитых с текущей

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$ git branch -D testing
Deleted branch testing (was 04f1387).

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two (master)
$
```

Рисунок 120 - Удаление ветки

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/Eternity-blip/repo1
  Push URL: https://github.com/Eternity-blip/repo1
  HEAD branch: main
  Remote branch:
    main tracked
  Local branches configured for 'git pull':
    main merges with remote main
    serv merges with remote main
  Local ref configured for 'git push':
    main pushes to main (local out of date)
```

Рисунок 121 - Просмотр удаленных веток

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 285 bytes | 95.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Eternity-blip/repo1
   cedd680..f6b94bc  main -> main
```

Рисунок 122 - Отправка изменений

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git fetch origin
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 693 bytes | 18.00 KiB/s, done.
From https://github.com/Eternity-blip/repo1
   f6b94bc..68cbf5f  main      -> origin/main
```

Рисунок 123 - Выполнение команды `git fetch`

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

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two/repo1 (serv)
$
```

Рисунок 124 - Создание ветки на основе удаленной ветки

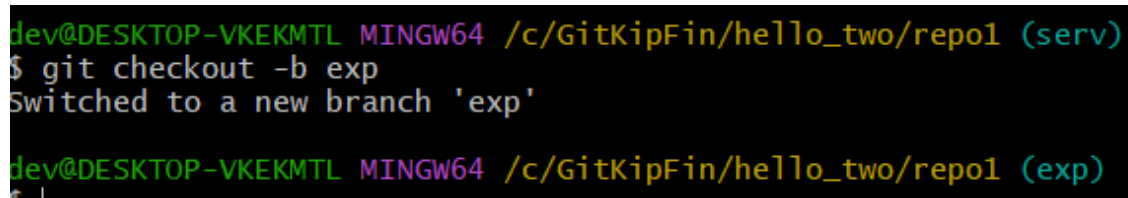
Для удаления веток на удаленном сервере используется команда, показанная на рисунке 10.

```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git push origin :serv
To https://github.com/Eternity-blip/repo1
- [deleted]          serv
```

Рисунок 125 - Удаление ветки на сервере

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

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



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two/repo1 (serv)
$ git checkout -b exp
Switched to a new branch 'exp'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two/repo1 (exp)
$ |
```

Рисунок 126 - Перемещение изменений

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



```
dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git rebase --onto master se cl
Successfully rebased and updated refs/heads/cl.
```

Рисунок 127 - Перемещение изменений с параметром onto

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two/repo1 (exp)
$ git checkout -b cl
Switched to a new branch 'cl'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two/repo1 (cl)
$ git checkout master
Switched to branch 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two/repo1 (master)
$ git merge cl
Already up to date.

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two/repo1 (master)
$

```

Рисунок 128 - Слияние веток master и cl

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git rebase master se
Successfully rebased and updated refs/heads/se.

```

Рисунок 129 - Перемещение изменений

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (se)
$ git checkout master
Switched to branch 'master'

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello (master)
$ git merge se
Updating 08d256f..4a98db2
Fast-forward
 e.txt | 0
 q.txt | 0
 repo1 | 1 +
3 files changed, 1 insertion(+)
create mode 100644 e.txt
create mode 100644 q.txt
create mode 160000 repo1

```

Рисунок 130 - Слияние веток master и se

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

```

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two/repo1 (master)
$ git branch -D cl
Deleted branch cl (was 4d50079).

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two/repo1 (master)
$ git branch -D se
Deleted branch se (was 4d50079).

dev@DESKTOP-VKEKMTL MINGW64 /c/GitKipFin/hello_two/repo1 (master)
$

```

Рисунок 131 - Удаление веток cl и se