

УП_1

Работа с Git

Подготовка

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop
$ git config --global user.name "Kafarov Emin"
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop
$ git config --global user.email "Eminkafarov22@mail.ru"
```

Рисунок 1 - Установка имени и электронной почты

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop
$ git config --global core.autocrlf true
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop
$ git config --global core.safecrlf true
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop
$ ^C
```

Рисунок 2 - Параметры окончаний строк

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop
$ git config --global core.quotePath off
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop
$ ^C
```

Рисунок 3 - Установка отображения unicode

Создание проекта

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



Рисунок 4 - Создание каталога и файла

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

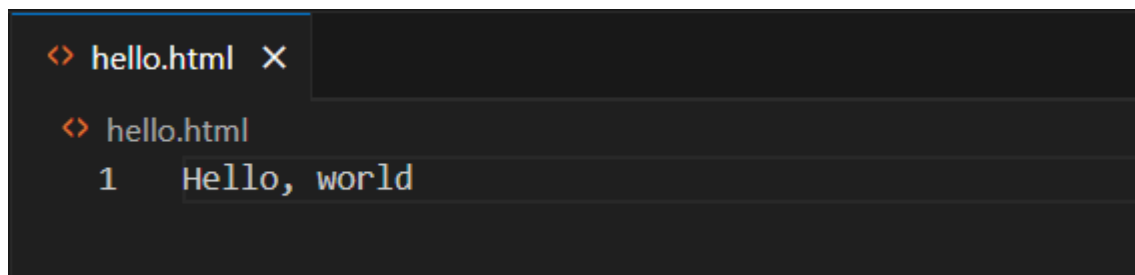


Рисунок 5 - Содержание файла

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

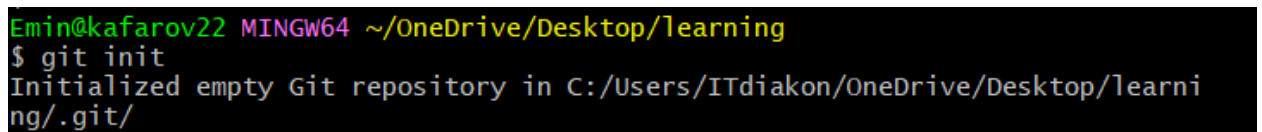


Рисунок 6 - Создание репозитория

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

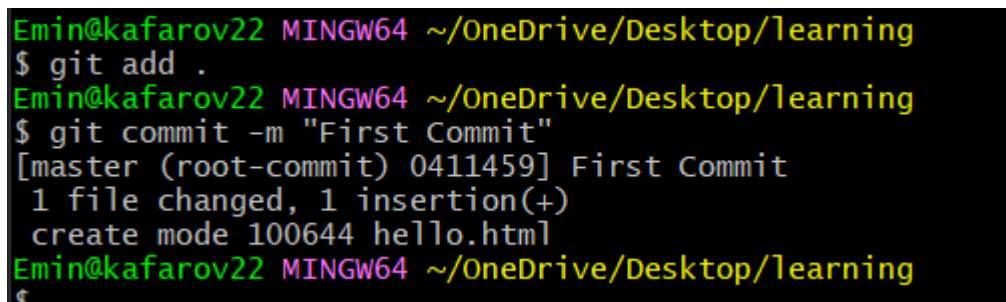


Рисунок 7 - Добавление в репозиторий

Проверка состояния

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git status
On branch master
nothing to commit, working tree clean
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 8 - Проверка состояния репозитория

Внесение изменений

Сначала необходимо внести изменения в файл (рисунок 9).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git status
On branch master
nothing to commit, working tree clean
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 9 - Внесение изменений в файл

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ touch file.txt
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ 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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file.txt

no changes added to commit (use "git add" and/or "git commit -a")
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 10 - Сообщение о незафиксированных изменениях

Индексация изменений

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git add .
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file.txt
        modified:   hello.html

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 11 - Команды для индексации изменений

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

Индексация и коммит

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

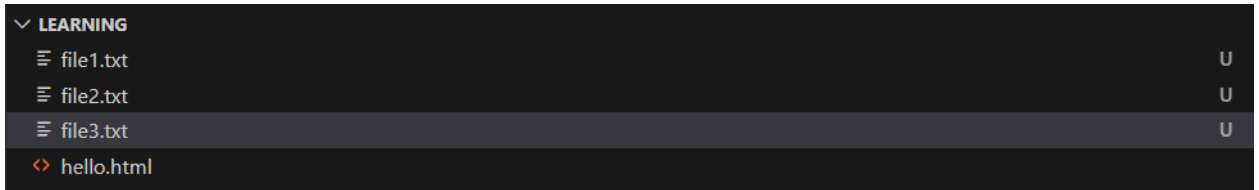


Рисунок 12 - Создано 3 файла

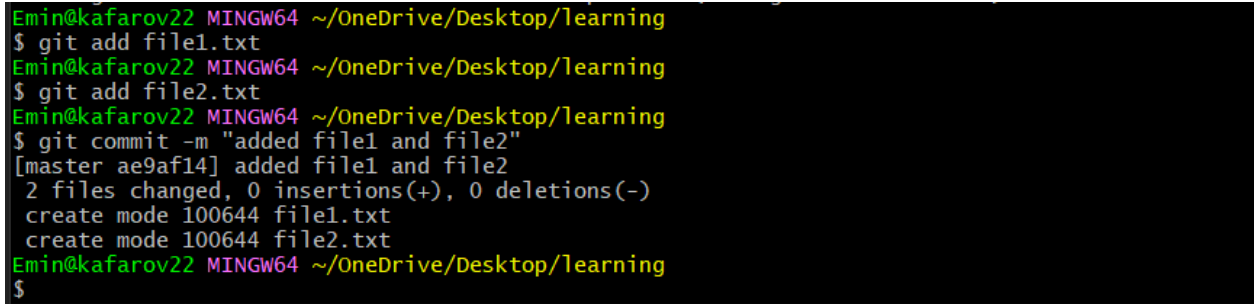


Рисунок 13 - Индексация и коммит для 2 файлов

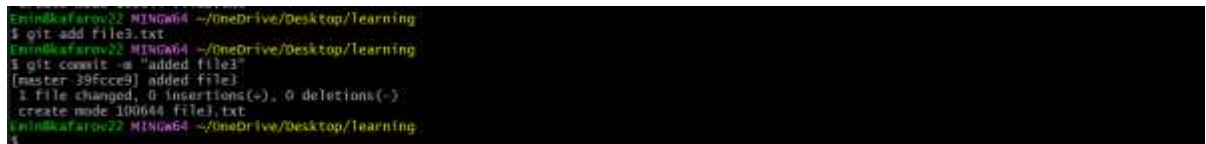
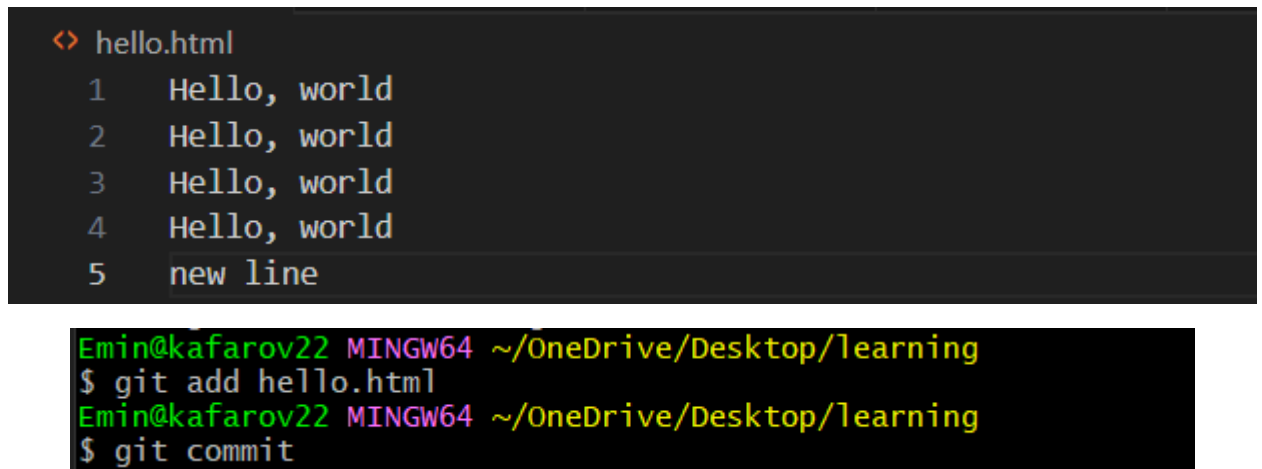


Рисунок 14 - Индексация и коммит для третьего файла

Коммит изменений

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

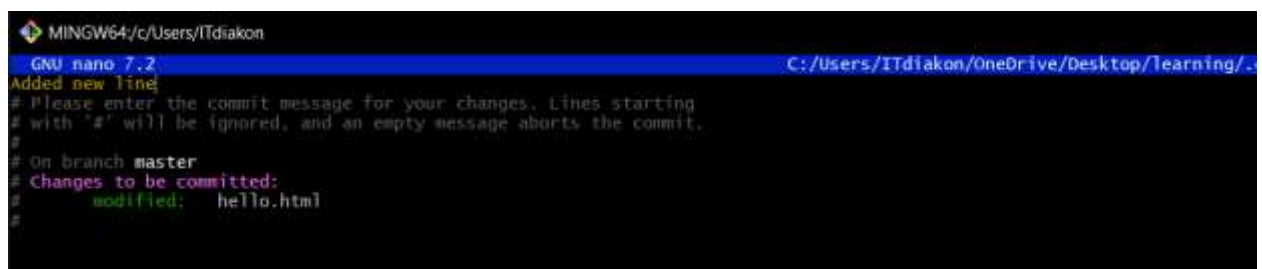


```
<> hello.html
1   Hello, world
2   Hello, world
3   Hello, world
4   Hello, world
5   new line

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git add hello.html
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git commit
```

Рисунок 15 - Коммит изменений

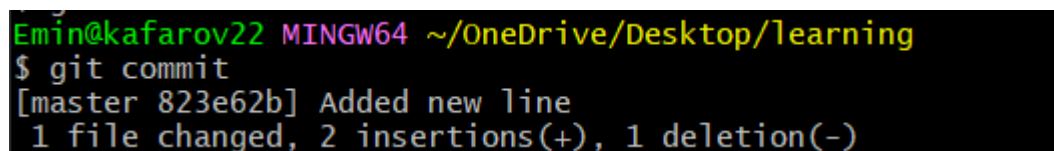
После этого будет открыт редактор, в котором необходимо на 1 строке записать комментарий коммита (рисунок 16).



```
MINGW64/c/Users/ITdiakon
GNU nano 7.2 C:/Users/ITdiakon/OneDrive/Desktop/learning/...
Added new line
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#   modified:   hello.html
#
```

Рисунок 16 - Ввод комментария

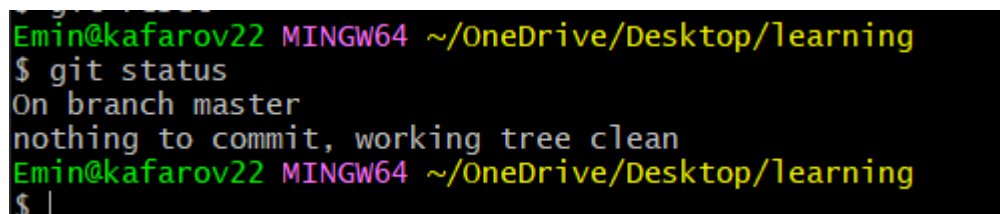
После выхода из текстового редактора будет указано следующее сообщение (рисунок 17).



```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git commit
[master 823e62b] Added new line
1 file changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 17 - Коммит-сообщения

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



```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git status
On branch master
nothing to commit, working tree clean
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 18 - Проверка состояния

Изменения, а не файлы

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

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

```
<> hello.html > ...
1  <html>
2  |   <body>
3  |       <h1>Hello, world</h1>
4  |   </body>
5  </html>
6  |
```

Рисунок 19 - Добавление тегов html и body

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

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>My Page</title>
5  </head>
6  <body>
7  |   <h1>Hello, world</h1>
8  </body>
9  </html>
```

Рисунок 20 - Добавление тега head

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.html

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 21 - Проверка состояния

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git commit -m "first change"
[master 3339bcb] first change
1 file changed, 8 insertions(+), 5 deletions(-)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ 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")
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
```

Рисунок 22 - Коммит 1 изменения и проверка состояния

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

```
HEAD is now at 3555beb first change
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git add .
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 23 - Индексация 2 изменения и проверка состояния

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git commit -m "second change"
[master 1c72d3b] second change
1 file changed, 8 insertions(+), 7 deletions(-)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 24 - Коммит 2 изменения

УП_2

Работа с Git

История

Для того, чтобы просмотреть список произведенных изменений в проекте, используется команда `git log` (рисунок 25).

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git log
commit 1c72d3b9c42885bca601282dcf50a7fd5104850b (HEAD -> master)
Author: Kafarov Emin <Eminkafarov22@mail.ru>
Date: Mon Jun 16 08:49:32 2025 +0300

    second change

commit 3339bcb0da366256558a3e0f988fb335265df3fb
Author: Kafarov Emin <Eminkafarov22@mail.ru>
Date: Mon Jun 16 08:46:47 2025 +0300

    first change

commit 823e62be8afd35f5a96da846b2e906e29722a11b
Author: Kafarov Emin <Eminkafarov22@mail.ru>
Date: Mon Jun 16 08:22:03 2025 +0300

    Added new line

commit 39fcce99447ed43b35c954a8eea05e2d832174e0
Author: Kafarov Emin <Eminkafarov22@mail.ru>
Date: Mon Jun 16 08:19:06 2025 +0300

    added file3

commit ae9af14bb961f67a8d70586ead82466578d2afc0
Author: Kafarov Emin <Eminkafarov22@mail.ru>
Date: Mon Jun 16 08:17:25 2025 +0300

    added file1 and file2

commit 53634d6136e5968a2966e991317519a82154a1ad
Author: Kafarov Emin <Eminkafarov22@mail.ru>
Date: Mon Jun 16 08:15:10 2025 +0300

    new commit

commit 04114598b132fc03d950bc7fa388849a89bcf7dc
Author: Kafarov Emin <Eminkafarov22@mail.ru>
Date: Mon Jun 16 08:05:51 2025 +0300

    First Commit

```

Рисунок 25 - Просмотр истории изменений

На рисунке 25 была выведена полная история. Для того, чтобы увидеть однострочный формат используется команда `git log --pretty=oneline` (рисунок 26).

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git log --oneline
1c72d3b (HEAD -> master) second change
3339bcb first change
823e62b Added new line
39fcce9 added file3
ae9af14 added file1 and file2
53634d6 new commit
0411459 First Commit
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$

```

Рисунок 26 - Однострочный формат вывода

Далее на рисунках 27-32 показано несколько вариантов вывода истории изменений.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git log --oneline -n 2
1c72d3b (HEAD -> master) second change
3339bcb first change
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 27 - Вывод последних 2 изменений

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git log --oneline --since="2025-06-15"
1c72d3b (HEAD -> master) second change
3339bcb first change
823e62b Added new line
39fcce9 added file3
ae9af14 added file1 and file2
53634d6 new commit
0411459 First Commit
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 28 - Вывод изменений начиная с определенного времени

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git log --oneline --until="2025-06-16"
1c72d3b (HEAD -> master) second change
3339bcb first change
823e62b Added new line
39fcce9 added file3
ae9af14 added file1 and file2
53634d6 new commit
0411459 First Commit
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 29 - Вывод изменений до определенного времени

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git log --oneline --author="Kafarov Emin"
1c72d3b (HEAD -> master) second change
3339bcb first change
823e62b Added new line
39fcce9 added file3
ae9af14 added file1 and file2
53634d6 new commit
0411459 First Commit
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 30 - Вывод изменений, внесенных определенным автором

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git log --oneline --all
1c72d3b (HEAD -> master) second change
3339bcb first change
823e62b Added new line
39fcce9 added file3
ae9af14 added file1 and file2
53634d6 new commit
0411459 First Commit
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 31 - Вывод всех изменений

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git log --oneline --all --author="Kafarov Emin" --until="2025-06-16" --since="2025-06-15"
1c72d3b (HEAD -> master) second change
3339bcb first change
823e62b Added new line
39fcce9 added file3
ae9af14 added file1 and file2
53634d6 new commit
0411459 First Commit
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 32 - Использование нескольких параметров

Алиасы

Для настройки алиасов используется команда, показанная на рисунке 33.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.co checkout
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.ci commit
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.st status
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.br branch
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.hist "log --pretty=format:'%h %ad | %s%d [%an]' --graph --date=short"
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.type 'cat-file -t'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.dump 'cat-file -p'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 33 - Настройка алиасов для некоторых команд

При выполнении алиаса будет выполнена определенная команда и выведены нужные данные (рисунок 34).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git hist
* 1c72d3b 2025-06-16 | second change (HEAD -> master) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 34 - Выполнение алиаса hist

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ alias gs='git status'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ gs
On branch master
nothing to commit, working tree clean
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git st
On branch master
nothing to commit, working tree clean
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 35 - Установка и выполнение алиаса gs

Получение старых версий

Для того, чтобы вернуть рабочий каталог к предыдущему состоянию, можно использовать следующий способ: для начала нужно узнать хэши предыдущих версий, что можно сделать с помощью ранее заданного алиаса hist (рисунок 36).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git hist
* 1c72d3b 2025-06-16 | second change (HEAD -> master) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
```

Рисунок 36 - Просмотр хэшей предыдущих версий

Далее нужно выполнить команду `git checkout` с номером нужного хэша (достаточно первых 7 знаков). После этого можно просмотреть содержимое файла с помощью команды `cat` (рисунок 37).

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git co 0411459
Note: switching to '0411459'.

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 0411459 First Commit
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ cat hello.html && echo -e "\nreturn in commit with hash 0411459"
Hello, world
return in commit with hash 0411459
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |

```

Рисунок 37 - Возвращение к нужной версии и просмотр содержимого файла

Возвращение к последней версии в ветке master

Для возвращения к последней версии в ветке master (имя ветки по умолчанию) надо ввести команду `git checkout master`, что показано на рисунке 38.

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git co master
Previous HEAD position was 0411459 First Commit
Switched to branch 'master'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ cat hello.html && echo -e "\n"
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
</head>
<body>
  <h1>Hello, world</h1>
</body>
</html>
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$

```

Рисунок 38 - Возвращение к последней версии в ветке master

Создание тегов версий

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git tag v1.0
```

Рисунок 39 - Задание тега

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git tag v1.0
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git tag v0.9 HEAD^
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git co v1.0^
Note: switching to 'v1.0^'.

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 3339bcb first change
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ cat hello.html
<html>
  <body>
    <h1>Hello, world</h1>
  <head>
    HEAD
  </head>
</body>
</html>
```

Рисунок 40 – Переход к предыдущей версии с помощью тега

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git tag v0.9 HEAD^
```

Рисунок 41 - Задание тега предыдущей версии

Теперь с помощью тегов можно переключаться между версиями (рисунок 42).


```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git co v1.0
Previous HEAD position was 3339bcb first change
HEAD is now at 1c72d3b second change
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git co v0.9
Previous HEAD position was 1c72d3b second change
HEAD is now at 3339bcb first change
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git co v1.0
Previous HEAD position was 3339bcb first change
HEAD is now at 1c72d3b second change
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$

```

Рисунок 42 - Переключение между версиями с помощью тегов

Для просмотра всех тегов используется команда `git tag` (рисунок 43).

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git tag
v0.9
v1.0

```

Рисунок 43 - Просмотр тегов

Также можно посмотреть теги в логе, как показано на рисунке 44.

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git hist
* 1c72d3b 2025-06-16 | second change (HEAD, tag: v1.0, master) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning

```

Рисунок 44 - Просмотр тегов в логе

Отмена локальных изменений (до индексации)

Сначала нужно переключиться на последний коммит `master` (рисунок 45).

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git co master
Switched to branch 'master'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning

```

Рисунок 45 - Переключение на последний коммит

Далее для работы нужно внести изменение в файл (рисунок 46).


```
<> hello.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>My Page</title>
5  </head>
6  <body>
7    <h1>Hello, world</h1>
8  </body>
9  </html>
10
11  "new line"
```

Рисунок 46 - Внесение изменения в файл

После выполнения команды `git status` будет показано, что есть не проиндексированное изменение (рисунок 47).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ 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")
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 47 - Не проиндексированное изменение

Для переключения в версию файла без изменений используется команда `git checkout hello.html` (рисунок 48). Команда `git status` покажет, что не было произведено изменений, не зафиксированных в каталоге.

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git checkout -- hello.html
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git st
On branch master
nothing to commit, working tree clean
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ cat hello.html
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
</head>
<body>
  <h1>Hello, world</h1>
</body>
</html>
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |

```

Рисунок 48 - Возвращение к версии

Отмена проиндексированных изменений (перед коммитом)

Для того, чтобы научиться отменять проиндексированные изменения, сначала нужно внести ненужное изменение в файл (рисунок 49). После этого производится индексация (рисунок 50).

```

<> hello.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>My Page</title>
5  </head>
6  <body>
7  |   <h1>Hello, world</h1>
8  </body>
9  </html>
10
11  "new line 2"

```

Рисунок 49 - Внесение ненужного изменения

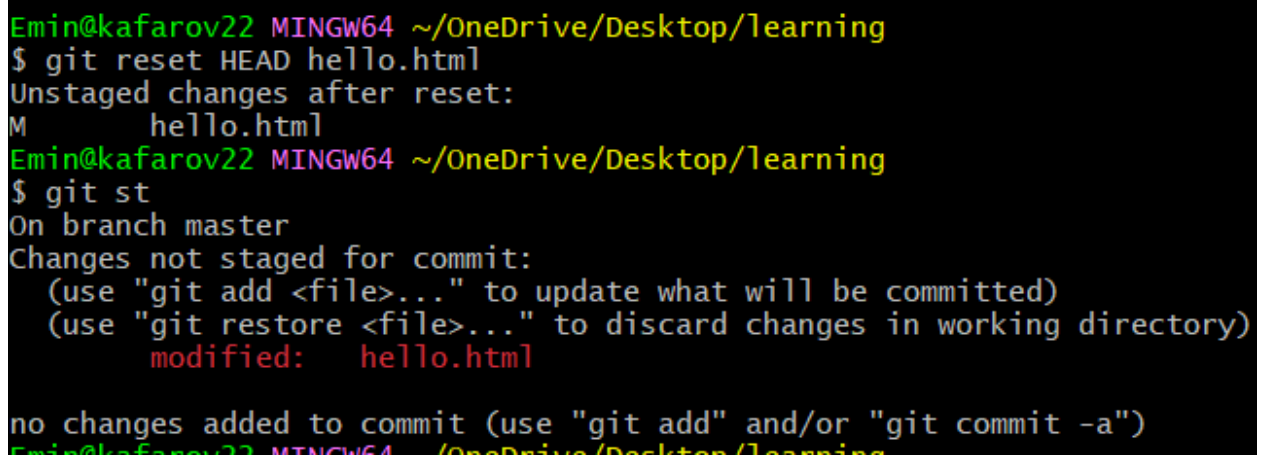
```

$ git add .
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git st
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$

```

Рисунок 50 - Индексация изменения

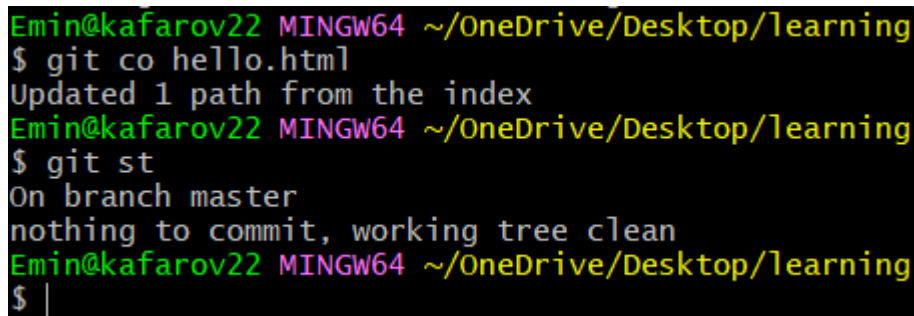
Для отмены индексации изменения используется команда `git reset HEAD hello.html` (рисунок 51). Команда `reset` сбрасывает буферную зону к HEAD и очищает ее от проиндексированных изменений. Но для удаления ненужного по-прежнему используется команда `git checkout` (рисунок 52).



```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git reset HEAD hello.html
Unstaged changes after reset:
M      hello.html
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ 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")
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
```

Рисунок 51 - Очистка буферной зоны



```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git co hello.html
Updated 1 path from the index
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git st
On branch master
nothing to commit, working tree clean
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 52 - Удаление ненужных изменений

Отмена коммитов

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

Для начала надо внести изменение, проиндексировать его и записать коммит (рисунки 53-54).

```
<> hello.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>My Page</title>
5  </head>
6  <body>
7  |   <h1>Hello, world</h1>
8  </body>
9  </html>
10 |
11 | ЭТА ЛИНИЯ БУДЕТ ЗАКОММИЧЕНА|
```

Рисунок 53 - Внесение изменения в файл

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ alias gac='git add . && git commit -m "New commit"'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ gac
[master 38edf18] New commit
1 file changed, 3 insertions(+), 1 deletion(-)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ gs
On branch master
nothing to commit, working tree clean
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 54 - Индексация и коммит

Для создания коммита, который удалит ненужные изменения, используется команда `git revert HEAD` (рисунок 55). После этого будет открыт редактор, в котором можно отредактировать коммит сообщение (рисунок 56), затем надо сохранить файл и закрыть редактор (рисунок 57).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git revert HEAD
```

Рисунок 55 - Выполнение команды `git revert`

```
MINGW64:/c/Users/ITdiakon
GNU nano 7.2                               C:/Users/ITd
Revert "New commit"

This reverts commit 38edf1841873bec26e0c7ee2362c0d712ea2caf1.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#   modified:   hello.html
#
```

Рисунок 56 - Коммит сообщение в редакторе

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git revert HEAD
[master 4b1fb6f] Revert "New commit"
1 file changed, 1 insertion(+), 3 deletions(-)
```

Рисунок 57 - Редактор закрыт

При проверке лога будут показаны все коммиты, в том числе и отмененные (рисунок 58).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git hist
* 4b1fb6f 2025-06-16 | Revert "New commit" (HEAD -> master) [Kafarov Emin]
* 38edf18 2025-06-16 | New commit [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
```

Рисунок 58 - Все коммиты при просмотре лога

Перед удалением коммита последний из них нужно отметить тегом, чтобы не потерять его (рисунок 59).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git tag oops
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git log -n 1
commit 4b1fb6f1d433f1dc1cc354a1c25497457c5d876f (HEAD -> master, tag: oops)
Author: Kafarov Emin <Eminkafarov22@mail.ru>
Date: Mon Jun 16 10:04:54 2025 +0300

    Revert "New commit"

    This reverts commit 38edf1841873bec26e0c7ee2362c0d712ea2caf1.
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git hist
* 4b1fb6f 2025-06-16 | Revert "New commit" (HEAD -> master, tag: oops) [Kafarov Emin]
* 38edf18 2025-06-16 | New commit [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 59 - Отметка тегом коммита

Для сброса коммитов используется команда `git reset --hard v1.0` (рисунок 60). Она сбрасывает ветку до версии с тегом v1.0.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git reset --hard v1.0
HEAD is now at 1c72d3b second change
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git hist
* 1c72d3b 2025-06-16 | second change (HEAD -> master, tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 60 - Сброс коммита

Но при просмотре лога с помощью команды `git hist --all` отмененные коммиты по-прежнему будут показываться, так как они всё еще находятся в репозитории (рисунок 61).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git hist --all
* 4b1fb6f 2025-06-16 | Revert "New commit" (tag: ooops) [Kafarov Emin]
* 38edf18 2025-06-16 | New commit [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (HEAD -> master, tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 61 - Сброшенные коммиты находятся по-прежнему в репозитории

Удаление тега

Так как тег «oooops» больше не нужен, его и коммиты, на которые он указывает, можно удалить с помощью команды `git tag -d` (рисунок 62).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git tag -d ooops
Deleted tag 'oops' (was 4b1fb6f)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git hist --all
* 1c72d3b 2025-06-16 | second change (HEAD -> master, tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

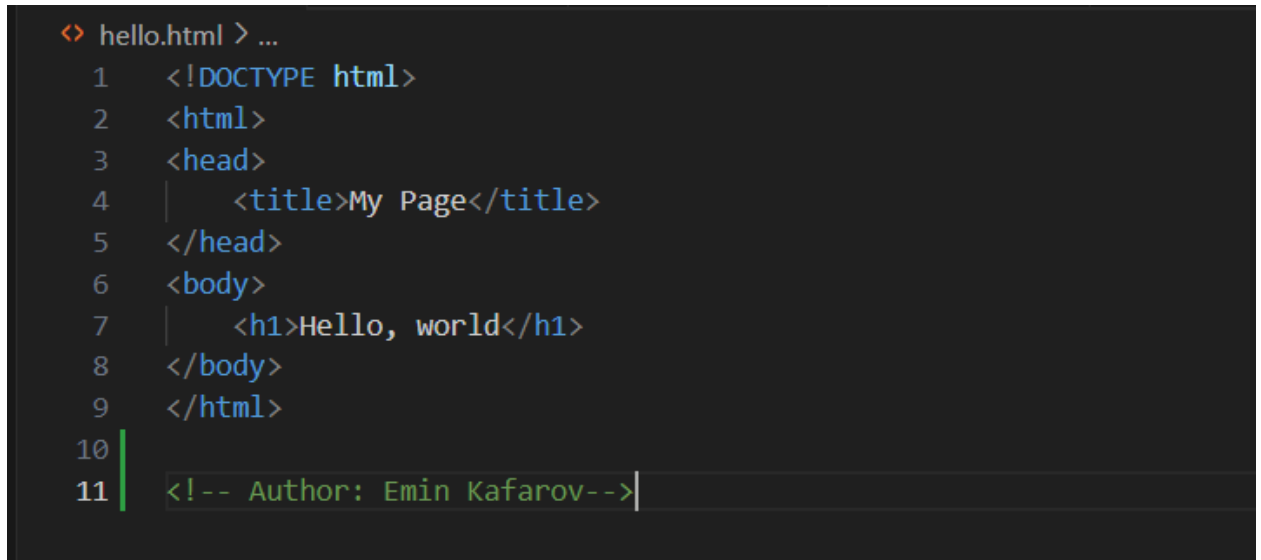
Рисунок 62 - Удаление тега

УП_3

Работа с Git

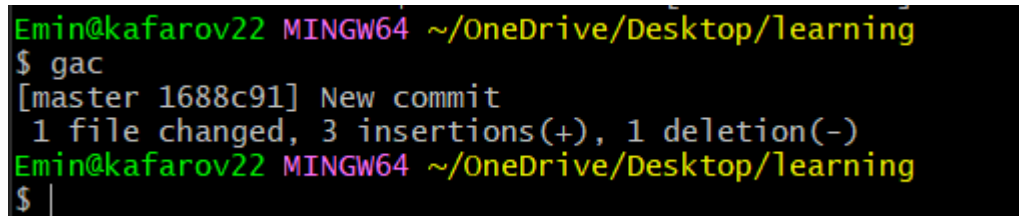
Внесение изменений в коммиты

Для начала будет создан коммит, в который позже будут внесены изменения. На рисунках 63 и 64 происходит добавление комментария в файл hello.html и его индексация и коммит.



```
<> hello.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>My Page</title>
5  </head>
6  <body>
7  |   <h1>Hello, world</h1>
8  </body>
9  </html>
10 |
11 | <!-- Author: Emin Kafarov-->
```

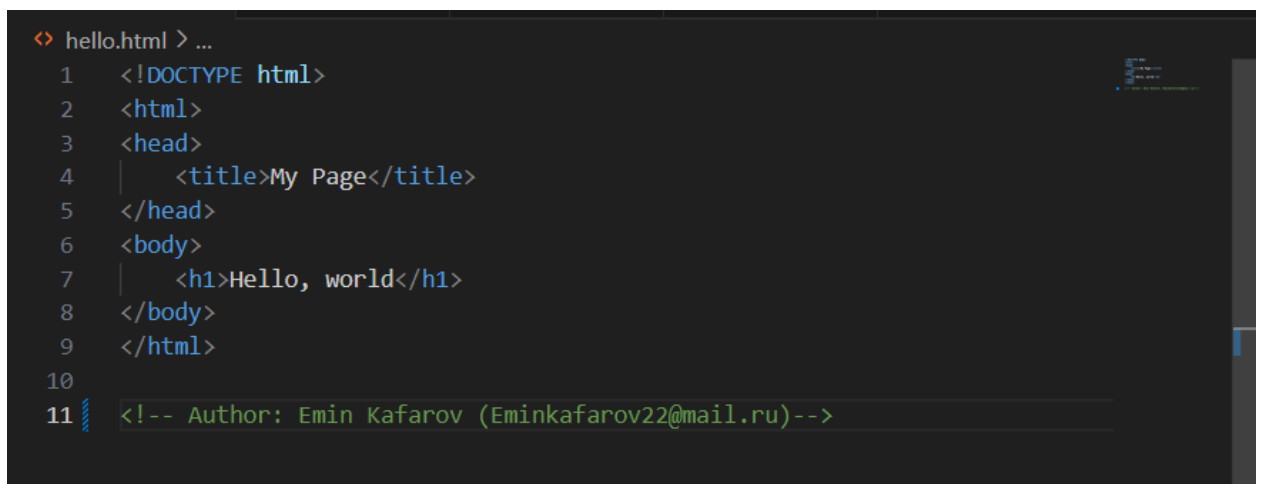
Рисунок 63 - Добавление комментария в файл



```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ gac
[master 1688c91] New commit
1 file changed, 3 insertions(+), 1 deletion(-)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 64 - Индексация и коммит

Далее необходимо добавить электронную почту в комментарий (рисунок 65).



```
<> hello.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>My Page</title>
5  </head>
6  <body>
7  |   <h1>Hello, world</h1>
8  </body>
9  </html>
10 |
11 | <!-- Author: Emin Kafarov (Eminkafarov22@mail.ru)-->
```

Рисунок 65 - Добавление электронной почты

Но для того, чтобы не создавать отдельный коммит ради электронной почты, можно изменить предыдущий так, как показано на рисунке 66.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git commit --amend -m "Add na author/email comment"
[master 2d99b79] Add na author/email comment
Date: Mon Jun 16 10:22:26 2025 +0300
1 file changed, 3 insertions(+), 1 deletion(-)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git hist
```

Рисунок 66 - Индексация и изменение коммита

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git hist --all
* 2d99b79 2025-06-16 | Add na author/email comment (HEAD -> master) [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
```

Рисунок 67 - Последний коммит изменен

Перемещение файлов

Для перемещения файлов в пределах репозитория используются команды, показанные на рисунке 68. После выполнения данных команды git индексирует эти изменения (удаление файла hello.html и создание файла lib/hello.html).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ mkdir lib
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git mv hello.html lib/
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ gs
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    hello.html -> lib/hello.html
```

Рисунок 68 - Перемещение файла

Далее надо осуществить коммит данного перемещения.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git commit -m "Moved hello.html to lib"
[master f230c94] Moved hello.html to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 69 – Коммит перемещения

Подробнее о структуре

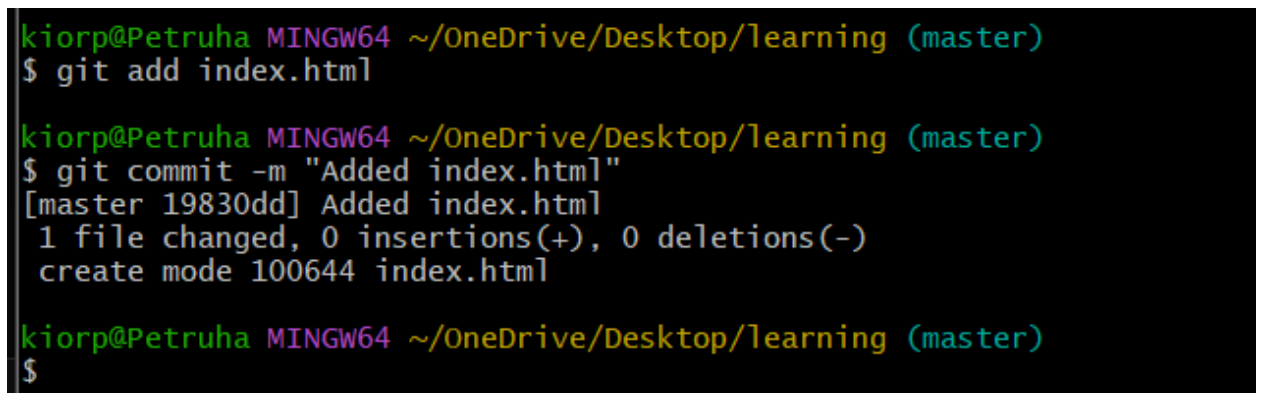
Необходимо добавить еще один файл в репозиторий. Это будет файл index.html с кодом, показанным на рисунке 70.



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

Рисунок 70 - Содержимое файла index.html

Далее нужно проиндексировать и закоммитить файл (рисунок 71).



```
kiorp@Petruha MINGW64 ~/OneDrive/Desktop/learning (master)
$ git add index.html

kiorp@Petruha MINGW64 ~/OneDrive/Desktop/learning (master)
$ git commit -m "Added index.html"
[master 19830dd] Added index.html
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.html

kiorp@Petruha MINGW64 ~/OneDrive/Desktop/learning (master)
$
```

Рисунок 71 - Индексация и коммит

При открытии файла index.html будет виден кусок страницы hello.html (рисунок 72).

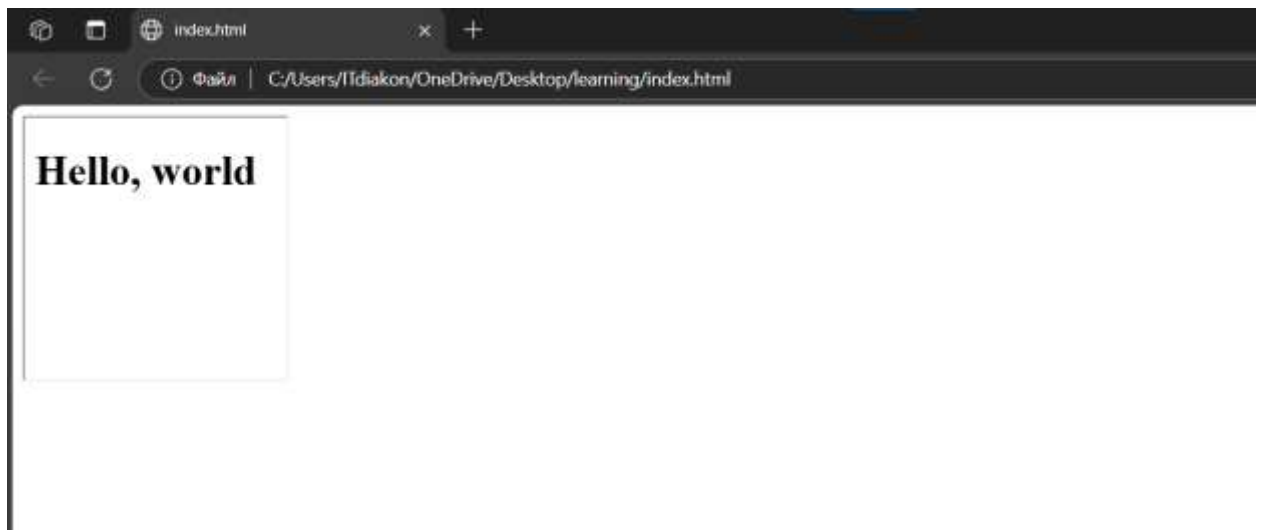


Рисунок 72 - Файл index.html, открытый в браузере

Каталог .git

Чтобы посмотреть структуру каталога .git необходимо выполнить команду, показанную на рисунке 73.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ ls -C .git
COMMIT_EDITMSG HEAD ORIG_HEAD config description hooks/ index info/ logs/ objects/ packed-refs refs/
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 73 - Содержание каталога .git

При аналогичном просмотре каталога objects можно будет увидеть множество каталогов с именами из 2 символов (рисунок 74). Имена каталогов являются первыми двумя буквами хэша.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ ls -C .git/objects/
04/ 1c/ 30/ 39/ 53/ 68/ 86/ 9a/ a5/ b6/ d0/ e6/ fb/ pack/
0b/ 21/ 33/ 4a/ 5e/ 81/ 90/ 9f/ aa/ c2/ d8/ e7/ fc/
16/ 2d/ 38/ 4b/ 66/ 82/ 93/ a1/ ae/ c3/ db/ f2/ info/
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
```

Рисунок 74 - Содержание каталога objects

При просмотре содержимого любого из каталогов будут показаны файлы, названия которых состоят из 38 символов (рисунок 75).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ ls -C .git/objects/fc
f108c76c0b08c5656d1411b25a44358c42006b
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 75 - Просмотр каталога dc

Далее требуется посмотреть файл конфигурации с помощью команды cat (рисунок 76).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 76 - Просмотр файла конфигурации

На рисунке 77 показан просмотр файлов в подкаталоге tags и веток в каталоге heads.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ ls .git/refs/tags/ && ls .git/refs/heads/
v0.9  v1.0
master
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ ls .git/refs/tags/v1.0
.git/refs/tags/v1.0
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 77 - Просмотр файлов и веток

Файл HEAD содержит ссылку на текущую ветку (рисунок 78).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ cat .git/HEAD
ref: refs/heads/master
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 78 - Содержимое файла HEAD

Работа с объектами git

Для начала необходимо посмотреть последний коммит (рисунок 17).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git hist --max-count=1
* 813e4d2 2025-06-16 | New commit (HEAD -> master) [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 79 - Последний коммит

Далее надо использовать хэш последнего коммита, используя команды `cat-file -p` и `cat-file -t` (рисунок 81) для просмотра объекта коммита. Также вместо длинных команд можно использовать сокращенные `type` и `dump`, если данные команды были заданы как алиасы.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git type 813e4d2
commit
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git dump 813e4d2
tree 3038295eab5f3fdc6788d8674c48630c03c8a4f0
parent f230c9476990643233c2664faa2c1088efb3ad5d
author Kafarov Emin <Eminkafarov22@mail.ru> 1750060252 +0300
committer Kafarov Emin <Eminkafarov22@mail.ru> 1750060252 +0300

New commit
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
```

Рисунок 81 - Использование алиасов

Для просмотра дерева каталогов необходимо использовать его хэш (рисунок 82).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git dump 3038295eab5f3fdc6788d8674c48630c03c8a4f0
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391    file1.txt
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391    file2.txt
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391    file3.txt
100644 blob 39145f1c59415eb43bdaeb67c64dc84eab09533d    index.html
040000 tree 8660bc99e1a5dd218e5a78c990ce83704cec7311    lib
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 80 - Просмотр дерева каталогов

Затем нужно посмотреть каталог lib (рисунок 83).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git dump 8660bc99e1a5dd218e5a78c990ce83704cec7311
100644 blob fcf108c76c0b08c5656d1411b25a44358c42006b    hello.html
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
```

Рисунок 81 - Просмотр каталога lib

И затем требуется вывести содержимое файла hello.html (рисунок 84).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git dump fcf108c76c0b08c5656d1411b25a44358c42006b
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
</head>
<body>
  <h1>Hello, world</h1>
</body>
</html>

<!-- Author: Emin Kafarov (Eminkafarov22@mail.ru)-->
```

Рисунок 82 - Вывод содержимого файла hello.html

Аналогичным образом можно просмотреть содержимое файла, каким оно было в самом первом коммите, как показано на рисунке 85. Для этого требуется использовать лишь нужные хэши.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git dump 0411459
tree d8a800db068656c3aafc962f7a438c987b49dd60
author Kafarov Emin <Eminkafarov22@mail.ru> 1750050351 +0300
committer Kafarov Emin <Eminkafarov22@mail.ru> 1750050351 +0300

First Commit
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 83 - Просмотр содержимого файла при первом коммите

УП_4

Работа с Git

Создание ветки

Для начала необходимо создать ветку style с помощью команды `git checkout -b style` (рисунок 86).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git checkout -b style
Switched to a new branch 'style'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ gs
On branch style
nothing to commit, working tree clean
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
```

Рисунок 84 - Создание новой ветки style

Затем нужно создать файл стилей (рисунок 87) и внести в него код, показанный на рисунке 88.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ touch lib/style.css
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ echo "h1 { color: red; }" > lib/style.css
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$
```

Рисунок 85 - Создание файла стилей

```
lib > # style.css > h1
1   h1 { color: red; }
2
```

Рисунок 86 - Код style.css

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git add lib/style.css
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git commit -m "Added stylesheet"
[style 4d8310b] Added stylesheet
1 file changed, 1 insertion(+)
create mode 100644 lib/style.css
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |
```

Рисунок 87 - Индексация и коммит нового файла

Далее требуется изменить основную страницу hello.html и закоммитить изменения (рисунки 90-91).

```
<> index.html  # style.css  <> hello.html M X
lib > <> hello.html > html > link
1   <!DOCTYPE html>
2   <html>
3   <head>
4   |   <title>My Page</title>
5   </head>
6   |   <link rel="stylesheet" href="style.css">
7   <body>
8   |   <h1>Hello, world</h1>
9   </body>
10  </html>
11
12  <!-- Author: Emin Kafarov (Eminkafarov22@mail.ru)-->
```

Рисунок 88 - Изменения в файле hello.html

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git add lib/hello.html
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git commit -m "hello.html use style.css"
[style b2704a2] hello.html use style.css
1 file changed, 1 insertion(+)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |

```

Рисунок 89 - Индексация и коммит

Далее аналогичные действия нужно осуществить с файлом index.html, как это показано на рисунках 92-93.

```

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

```

Рисунок 90 - Изменения в файле index.html

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git add index.html
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git commit -m "Linked stylesheet in index.html"
[style 3bb3e30] Linked stylesheet in index.html
1 file changed, 3 insertions(+)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |

```

Рисунок 91 - Индексация и коммит

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

Навигация по веткам

При просмотре истории, как на рисунке 94, можно увидеть, что теперь в проекте 2 ветки.

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git branch -a
master
* style
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git hist --all
* 3bb3e30 2025-06-16 | Linked stylesheet in index.html (HEAD -> style) [Kafarov Emin]
* b2704a2 2025-06-16 | hello.html use style.css [Kafarov Emin]
* 4d8310b 2025-06-16 | Added stylesheet [Kafarov Emin]
* 813e4d2 2025-06-16 | New commit (master) [Kafarov Emin]
* f230c94 2025-06-16 | Moved hello.html to lib [Kafarov Emin]
* 2d99b79 2025-06-16 | Add na author/email comment [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning

```

Рисунок 92 - Просмотр истории

Для переключения на ветку master используется команда `git checkout master` (рисунок 95). После переключения на нужную ветку при выводе файла `hello.html` можно увидеть, что изменения отсутствуют (по причине того, что они закоммичены в другой ветке).

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git checkout master
Switched to branch 'master'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ cat lib/hello.html
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
</head>
<body>
  <h1>Hello, world</h1>
</body>
</html>

<!-- Author: Emin Kafarov (Eminkafarov22@mail.ru)-->

```

Рисунок 93 - Переключение на ветку master

При переключении на ветку `style` файл `hello.html` будет иметь другое содержание (рисунок 96).


```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git co style
Switched to branch 'style'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ cat lib/hello.html && echo -e "\n"
<!DOCTYPE html>
<html>
<head>
  <title>My Page</title>
</head>
  <link rel="stylesheet" href="style.css">
<body>
  <h1>Hello, world</h1>
</body>
</html>

<!-- Author: Emin Kafarov (Eminkafarov22@mail.ru)-->

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$

```

Рисунок 94 - Переключение на ветку style

Изменения в ветке master

Необходимо переключиться на ветку master (рисунок 97) и добавить файл README (рисунки 98-99).

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git co master
Switched to branch 'master'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ |

```

Рисунок 95 - Переключение на ветку master

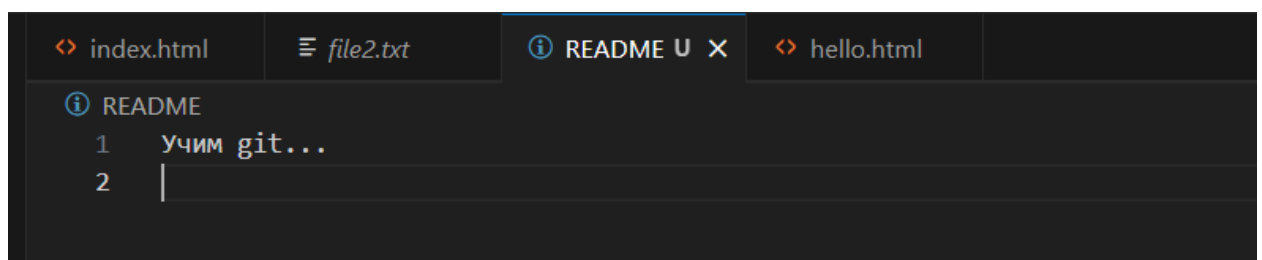


Рисунок 96 - Содержимое файла README

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git add README
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git commit -m "Added README"
[master 933fa53] Added README
1 file changed, 1 insertion(+)
create mode 100644 README
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$
```

Рисунок 97 - Индексация и коммит

Просмотр отличающихся веток

На рисунке 100 можно увидеть дерево коммитов.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git hist --all
* 933fa53 2025-06-16 | Added README (HEAD -> master) [Kafarov Emin]
| * 3bb3e30 2025-06-16 | Linked stylesheet in index.html (style) [Kafarov Emin]
| * b2704a2 2025-06-16 | hello.html use style.css [Kafarov Emin]
| * 4d8310b 2025-06-16 | Added stylesheet [Kafarov Emin]
| /
* 813e4d2 2025-06-16 | New commit [Kafarov Emin]
* f230c94 2025-06-16 | Moved hello.html to lib [Kafarov Emin]
* 2d99b79 2025-06-16 | Add na author/email comment [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
```

Рисунок 98 - 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 101-103).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git co style
Switched to branch 'style'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
```

Рисунок 99 - Переключение на ветку style

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git merge master
Merge made by the 'ort' strategy.
 README | 1 +
1 file changed, 1 insertion(+)
create mode 100644 README
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$
```

Рисунок 100 - Слияние с веткой master

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git hist --all
* 0072076 2025-06-16 | Merge branch 'master' into style (HEAD -> style) [Kafarov Emin]
|
| * 933fa53 2025-06-16 | Added README (master) [Kafarov Emin]
| * 3bb3e30 2025-06-16 | Linked stylesheet in index.html [Kafarov Emin]
| * b2704a2 2025-06-16 | hello.html use style.css [Kafarov Emin]
| * 4d8310b 2025-06-16 | Added stylesheet [Kafarov Emin]
|/
* 813e4d2 2025-06-16 | New commit [Kafarov Emin]
* f230c94 2025-06-16 | Moved hello.html to lib [Kafarov Emin]
* 2d99b79 2025-06-16 | Add na author/email comment [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$
```

Рисунок 101 - Просмотр истории

Создание конфликта

Для того, чтобы создать конфликт необходимо перейти в ветку master и внести изменения в файл hello.html (рисунки 104-106).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git co master
Switched to branch 'master'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$
```

Рисунок 102 - Переход в ветку master

```
lib > <> hello.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>My Page</title>
5  </head>
6  |   <link rel="stylesheet" href="style.css">
7  <body>
8  |   <h1>Hello, world, WORLD</h1>
9  </body>
10 </html>
11 Life is awesome!
12 <!-- Author: Emin Kafarov (Eminkafarov22@mail.ru)-->
```

Рисунок 103 - Внесение изменений

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git commit -m "Changed text in master branch"
[master dceee19] Changed text in master branch
1 file changed, 1 insertion(+), 1 deletion(-)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
```

Рисунок 104 - Индексация и коммит

После выполнения предыдущих действий при просмотре веток можно будет увидеть конфликт в виде, как на рисунке 107.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git hist --all
* a4ad6ba 2025-06-16 | Life os great! (HEAD -> master) [Kafarov Emin]
| * 0072076 2025-06-16 | Merge branch 'master' into style (style) [Kafarov Emin]
| |
| | \
| | /
| | /
| | /
| |
| * 933fa53 2025-06-16 | Added README [Kafarov Emin]
| * 3bb3e30 2025-06-16 | Linked stylesheet in index.html [Kafarov Emin]
| * b2704a2 2025-06-16 | hello.html use style.css [Kafarov Emin]
| * 4d8310b 2025-06-16 | Added stylesheet [Kafarov Emin]
| /
|
* 813e4d2 2025-06-16 | New commit [Kafarov Emin]
* f230c94 2025-06-16 | Moved hello.html to lib [Kafarov Emin]
* 2d99b79 2025-06-16 | Add na author/email comment [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
```

Рисунок 105 - Конфликт изменений

Разрешение конфликтов

При попытке объединить ветку style с master будет показана ошибка из-за конфликта, как показано на рисунке 108.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git merge master
Auto-merging lib/hello.html
CONFLICT (content): Merge conflict in lib/hello.html
Automatic merge failed; fix conflicts and then commit the result.
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
```

Рисунок 106 - Ошибка при слиянии

При открытии файла hello.html конфликт будет показан (рисунок 109).

```
lib > hello.html > ?
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>My Page</title>
5  </head>
6    <link rel="stylesheet" href="style.css">
7  <body>
8    <h1>Hello, world, WORLD</h1>
9  </body>
10 </html>
Принять текущее изменение | Принять входящее изменение | Принять оба изменения | Сравнить изменения
11 <----- HEAD (текущее изменение)
12 Life is awesome!
13 -----
14 Life is beautiful!
15 >>>>> master (входящее изменение)
16 <!-- Author: Emin Kafarov (Eminkafarov22@mail.ru)-->
```

Рисунок 107 - Просмотр файла hello.html при наличии конфликта

Чтобы решить конфликт, нужно внести изменения вручную (рисунок 110).

```
lib > hello.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>My Page</title>
5  </head>
6    <link rel="stylesheet" href="style.css">
7  <body>
8    <h1>Hello, world, WORLD</h1>
9  </body>
10 </html>
11 Life is beautiful!
12 <!-- Author: Emin Kafarov (Eminkafarov22@mail.ru)-->
```

Рисунок 108 - Решение конфликта вручную

Затем следует произвести индексацию и коммит (рисунок 111).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ gs
On branch style
All conflicts fixed but you are still merging.
(use "git commit" to conclude merge)

Changes to be committed:
  modified:   lib/hello.html

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git commit -m "Merged master fixed conflict"
[style a9fd567] Merged master fixed conflict
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
```

Рисунок 109 - Индексация и коммит

УП_5

Работа с Git

Сброс ветки style

Для сброса ветки необходимо применить команду `reset --hard` до требуемой точки (рисунки 112-113).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git hist
* a9fd567 2025-06-16 | Merged master fixed conflict (HEAD -> style) [Kafarov Emin]
| \
| * dceee19 2025-06-16 | Changed text in master branch (master) [Kafarov Emin]
* | 713f81d 2025-06-16 | Changed text in style branch [Kafarov Emin]
* | be8308b 2025-06-16 | Merge branch 'master' into style [Kafarov Emin]
| \
| * a4ad6ba 2025-06-16 | Life os great! [Kafarov Emin]
* | 0072076 2025-06-16 | Merge branch 'master' into style [Kafarov Emin]
| \
| * 933fa53 2025-06-16 | Added README [Kafarov Emin]
* | 3bb3e30 2025-06-16 | Linked stylesheet in index.html [Kafarov Emin]
* | b2704a2 2025-06-16 | hello.html use style.css [Kafarov Emin]
* | 4d8310b 2025-06-16 | Added stylesheet [Kafarov Emin]
| /
* 813e4d2 2025-06-16 | New commit [Kafarov Emin]
* f230c94 2025-06-16 | Moved hello.html to lib [Kafarov Emin]
* 2d99b79 2025-06-16 | Add na author/email comment [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git branch -a
  master
* style
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$
```

Рисунок 110 - Просмотр истории


```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git reset --hard 3bb3e30
HEAD is now at 3bb3e30 Linked stylesheet in index.html
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git hist all
fatal: ambiguous argument 'all': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git hist --all
* dceee19 2025-06-16 | Changed text in master branch (master) [Kafarov Emin]
* a4ad6ba 2025-06-16 | Life os great! [Kafarov Emin]
* 933fa53 2025-06-16 | Added README [Kafarov Emin]
* 3bb3e30 2025-06-16 | Linked stylesheet in index.html (HEAD -> style) [Kafarov Emin]
* b2704a2 2025-06-16 | hello.html use style.css [Kafarov Emin]
* 4d8310b 2025-06-16 | Added stylesheet [Kafarov Emin]
/
* 813e4d2 2025-06-16 | New commit [Kafarov Emin]
* f230c94 2025-06-16 | Moved hello.html to lib [Kafarov Emin]
* 2d99b79 2025-06-16 | Add na author/email comment [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$

```

Рисунок 111 - Сброс ветки style

Сброс ветки master

Аналогичные действия нужно произвести и для ветки master (рисунки 114-115).

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git co master
Switched to branch 'master'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git hist
* dceee19 2025-06-16 | Changed text in master branch (HEAD -> master) [Kafarov Emin]
* a4ad6ba 2025-06-16 | Life os great! [Kafarov Emin]
* 933fa53 2025-06-16 | Added README [Kafarov Emin]
* 813e4d2 2025-06-16 | New commit [Kafarov Emin]
* f230c94 2025-06-16 | Moved hello.html to lib [Kafarov Emin]
* 2d99b79 2025-06-16 | Add na author/email comment [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$

```

Рисунок 112 - Переключение на master и просмотр истории

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git reset --hard a4ad6ba
HEAD is now at a4ad6ba Life os great!
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git hist
* a4ad6ba 2025-06-16 | Life os great! (HEAD -> master) [Kafarov Emin]
* 933fa53 2025-06-16 | Added README [Kafarov Emin]
* 813e4d2 2025-06-16 | New commit [Kafarov Emin]
* f230c94 2025-06-16 | Moved hello.html to lib [Kafarov Emin]
* 2d99b79 2025-06-16 | Add na author/email comment [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$

```

Рисунок 113 - Сброс ветки master

Перебазирование

Команду rebase можно использовать вместо команды merge (рисунок 116).

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git co style
Switched to branch 'style'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git rebase master
Successfully rebased and updated refs/heads/style.
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git hist
* 4dcc6e1 2025-06-16 | Linked stylesheet in index.html (HEAD -> style) [Kafarov Emin]
* 79c2e59 2025-06-16 | hello.html use style.css [Kafarov Emin]
* 9b0cb5d 2025-06-16 | Added stylesheet [Kafarov Emin]
* a4ad6ba 2025-06-16 | Life os great! (master) [Kafarov Emin]
* 933fa53 2025-06-16 | Added README [Kafarov Emin]
* 813e4d2 2025-06-16 | New commit [Kafarov Emin]
* f230c94 2025-06-16 | Moved hello.html to lib [Kafarov Emin]
* 2d99b79 2025-06-16 | Add na author/email comment [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$

```

Рисунок 114 - Перебазирование веток

Слияние в ветку master

Далее требуется произвести слияние веток с помощью merge (рисунки 117-118).


```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (style)
$ git co master
Switched to branch 'master'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git merge style
Updating a4ad6ba..4dcc6e1
Fast-forward
 index.html      | 3 +++
 lib/hello.html  | 1 +
 lib/style.css   | 1 +
 3 files changed, 5 insertions(+)
 create mode 100644 lib/style.css
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ |

```

Рисунок 115 - Слияние веток

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git hist
* 4dcc6e1 2025-06-16 | Linked stylesheet in index.html (HEAD -> master, style) [Kafarov Emin]
* 79c2e59 2025-06-16 | hello.html use style.css [Kafarov Emin]
* 9b0cb5d 2025-06-16 | Added stylesheet [Kafarov Emin]
* a4ad6ba 2025-06-16 | Life os great! [Kafarov Emin]
* 933fa53 2025-06-16 | Added README [Kafarov Emin]
* 813e4d2 2025-06-16 | New commit [Kafarov Emin]
* f230c94 2025-06-16 | Moved hello.html to lib [Kafarov Emin]
* 2d99b79 2025-06-16 | Add na author/email comment [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$

```

Рисунок 116 - Просмотр истории

Клонирование репозиториев

Далее требуется научиться делать копии репозиториев. Для этого необходимо перейти в рабочий каталог и затем использовать команду git clone. Все данные действия показаны на рисунке 119.

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ cd ~/dev/hello/ && git clone https://EminKafarov@github.com/EminKafarov/hello-world.git cloned_learning
Cloning into 'cloned_learning'...
remote: Enumerating objects: 49, done.
remote: Counting objects: 100% (49/49), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 49 (delta 18), reused 49 (delta 18), pack-reused 0 (from 0)
Receiving objects: 100% (49/49), done.
Resolving deltas: 100% (18/18), done.

```

Рисунок 117 - Переход в рабочий каталог и его клонирование

Просмотр клонированного репозитория

После этого можно просмотреть клонированный репозиторий (рисунки 120-121).

```
Emin@kafarov22 MINGW64 ~/dev/hello ()
$ cd cloned_learning/
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ ls
README file1.txt file2.txt file3.txt index.html lib/
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$
```

Рисунок 118 - Просмотр содержимого клонированного репозитория

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git hist --all
* 4dcc6e1 2025-06-16 | Linked stylesheet in index.html (HEAD -> master, origin/master, origin/HEAD) [Kafarov Emin]
* 79c2e59 2025-06-16 | hello.html use style.css [Kafarov Emin]
* 9b0cb5d 2025-06-16 | Added stylesheet [Kafarov Emin]
* a4ad6ba 2025-06-16 | Life os great! [Kafarov Emin]
* 933fa53 2025-06-16 | Added README [Kafarov Emin]
* 813e4d2 2025-06-16 | New commit [Kafarov Emin]
* f230c94 2025-06-16 | Moved hello.html to lib [Kafarov Emin]
* 2d99b79 2025-06-16 | Add na author/email comment [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change [Kafarov Emin]
* 3339bcb 2025-06-16 | First change [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcee9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ |
```

Рисунок 119 - Просмотр логов клонированного каталога

Origin

Origin – имя по умолчанию. Просмотр данных о нем возможен с помощью команд, показанных на рисунке 122.

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git remote show origin
* remote origin
Fetch URL: https://EminKafarov@github.com/EminKafarov/hello-world.git
Push URL: https://EminKafarov@github.com/EminKafarov/hello-world.git
HEAD branch: master
Remote branch:
  master 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)
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ |
```

Рисунок 120 - Просмотр данных об origin

Удаленные ветки

Для просмотра удаленных веток используется команда `git branch -a` (рисунок 123).

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/master
```

Рисунок 121 - Просмотр удаленных веток

Уп_6

Работа с Git

Сначала необходимо внести изменения в оригинальный репозиторий. Для этого нужно перейти в данный репозиторий (рисунок 124).

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ cd ~/OneDrive/Desktop/learning/
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$
```

Рисунок 122 - Переход в оригинальный репозиторий

Далее надо внести изменения в файл README (рисунок 125) и затем произвести индексацию и коммит (рисунок 126).

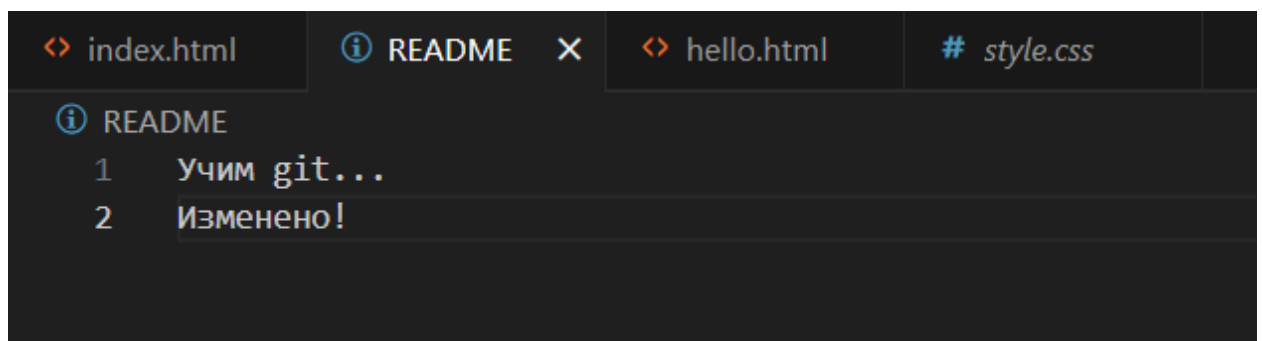


Рисунок 123 - Изменения в файле README

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git add . && git commit -m "Changed README"
[master 6835207] Changed README
1 file changed, 1 insertion(+)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ |
```

Рисунок 124 - Индексация и коммит новых изменений

Далее требуется перейти в клонированный репозиторий и извлечь изменения с помощью команды git fetch (рисунок 127) и просмотреть историю (рисунок 128).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ clon
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 288 bytes | 26.00 KiB/s, done.
From https://github.com/EminKafarov/hello-world
4dcc6e1..6835207 master -> origin/master
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ |
```

Рисунок 125 - Извлечение изменений

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git hist --all
* 6835207 2025-06-16 | Changed README (origin/master, origin/HEAD) [EminKafarov]
* 4dcc6e1 2025-06-16 | Linked stylesheet in index.html (HEAD -> master) [Kafarov Emin]
* 79c2e59 2025-06-16 | hello.html use style.css [Kafarov Emin]
* 9b0cb5d 2025-06-16 | Added stylesheet [Kafarov Emin]
* a4ad6ba 2025-06-16 | Life os great! [Kafarov Emin]
* 933fa53 2025-06-16 | Added README [Kafarov Emin]
* 813e4d2 2025-06-16 | New commit [Kafarov Emin]
* f230c94 2025-06-16 | Moved hello.html to lib [Kafarov Emin]
* 2d99b79 2025-06-16 | Add na author/email comment [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change [Kafarov Emin]
* 3339bcb 2025-06-16 | first change [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$
```

Рисунок 126 - Просмотр истории

При попытке вывести содержимое файла README можно увидеть, что изменения не были внесены (рисунок 129).

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ cat README
Учим git...
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$
```

Рисунок 127 - Вывод содержимого файла README

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

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git merge origin/master
Updating 4dcc6e1..6835207
Fast-forward
 README | 1 +
 1 file changed, 1 insertion(+)
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
```

Рисунок 128 - Слияние изменений

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

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ cat README && echo -e "\n"
Учим git...
Изменено!

Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$
```

Рисунок 129 - Вывод содержимого файла README

Также существует команда, объединяющая функции git fetch и git merge, которая показана на рисунке 132.

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git pull
Already up to date.
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ |
```

Рисунок 130 - Команда git pull

Далее требуется добавить локальную ветку, которая будет отслеживать удаленную ветку (рисунок 133).

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git fetch --all
From https://github.com/EminKafarov/hello-world
* [new branch]      style      -> origin/style
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git br -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git br --track style origin/style
branch 'style' set up to track 'origin/style'.
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git br -a
* master
  style
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ |
```

Рисунок 131 - Добавление локальной ветки

Далее необходимо создать чистый репозиторий (рисунок 134).

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning
$ cd .. && git clone --bare cloned_learning cloned_learning.git
Cloning into bare repository 'cloned_learning.git'...
done.
Emin@kafarov22 MINGW64 ~/dev/hello
$ ls cloned_learning.git/
HEAD config description hooks/ info/ objects/ packed-refs refs/
Emin@kafarov22 MINGW64 ~/dev/hello
$
```

Рисунок 132 - Создание чистого репозитория

Для добавления удаленного репозитория используется команда, показанная на рисунке 135.

```
Emin@kafarov22 MINGW64 ~/dev/hello
$ cd cloned_learning
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning
$ git remote add shared ../cloned_learning.git
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning
$
```

Рисунок 133 - Добавление удаленного репозитория

Затем требуется научиться отправлять изменения в удаленный репозиторий. Для этого сначала надо внести изменения, проиндексировать и произвести коммит (рисунок 136-137).

```
① README
1   Учим git...
2   Изменено!
3   changed in the original and pushed to shared
4   This is the Hello World example from the Git tutorial
```

Рисунок 134 - Внесение изменений в файл

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git add .
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git commit -m "Changed README in original repo"
[master 28a23b4] Changed README in original repo
1 file changed, 1 insertion(+), 3 deletions(-)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
```

Рисунок 135 - Индексация и коммит

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


```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git push shared master
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 16 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 989 bytes | 989.00 KiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0
To C:/Users/ITdiakon/dev/hello/cloned_learning.git
841eef5..dbd6ad8 master -> master
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ |
```

Рисунок 136 - Отправка изменений в общий репозиторий

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ cd ..
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$
```

Рисунок 137 - Переход в клонированный репозиторий

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git branch --track shared master
branch 'shared' set up to track 'master'.
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git pull shared master
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 9 (delta 5), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), 969 bytes | 31.00 KiB/s, done.
From ../cloned_learning
* branch master -> FETCH_HEAD
841eef5..dbd6ad8 master -> shared/master
Updating 841eef5..dbd6ad8
Fast-forward
 README | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ |
```

Рисунок 138 - 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).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop
$ git daemon --verbose --export-all --base-path=. --reuseaddr
[9948] Ready to rumble
```

Рисунок 139 - Настройка сервера

```
Emin@kafarov22 MINGW64 ~/dev/hello
$ git clone git://localhost/learning network_hello
Cloning into 'network_hello'...
remote: Enumerating objects: 64, done.
remote: Counting objects: 100% (64/64), done.
remote: Compressing objects: 100% (54/54), done.
remote: Total 64 (delta 26), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (64/64), 5.27 KiB | 2.63 MiB/s, done.
Resolving deltas: 100% (26/26), done.
Emin@kafarov22 MINGW64 ~/dev/hello
$ ls network_hello/
README file1.txt file2.txt file3.txt index.html lib/
Emin@kafarov22 MINGW64 ~/dev/hello
```

Рисунок 140 - Клонирование проекта

УП_7

Работа с Git

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ cd ~/dev/hello/ && git clone https://EminKafarov@github.com/EminKafarov/hello-world.git cloned_learning
Cloning into 'cloned_learning'...
remote: Enumerating objects: 49, done.
remote: Counting objects: 100% (49/49), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 49 (delta 18), reused 49 (delta 18), pack-reused 0 (from 0)
Receiving objects: 100% (49/49), done.
Resolving deltas: 100% (18/18), done.
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git remote
origin
shared
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$
```

Рисунок 141 – Клонирование репозитория и просмотр удаленных репозиторий

Можно указать ключ `-v`, чтобы просмотреть адреса для чтения и записи, привязанные к репозиторию (рисунок 144).

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git remote -v
origin https://EminKafarov@github.com/EminKafarov/hello-world.git (fetch)
origin https://EminKafarov@github.com/EminKafarov/hello-world.git (push)
shared ../cloned_learning.git (fetch)
shared ../cloned_learning.git (push)
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
```

Рисунок 142 - Просмотр удаленных репозиторий с ключом `-v`

Для добавления удаленного репозитория с новым именем используется команда `git remote add` (рисунок 145).


```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git remote add pbh https://EminKafarov@github.com/EminKafarov/hello-world.git
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git remote -v
origin https://EminKafarov@github.com/EminKafarov/hello-world.git (fetch)
origin https://EminKafarov@github.com/EminKafarov/hello-world.git (push)
pbh https://EminKafarov@github.com/EminKafarov/hello-world.git (fetch)
pbh https://EminKafarov@github.com/EminKafarov/hello-world.git (push)
shared C:/Users/ITdiakon/dev/hello/cloned_learning.git (fetch)
shared C:/Users/ITdiakon/dev/hello/cloned_learning.git (push)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ |
```

Рисунок 143 - Добавление удаленного репозитория

После задания имени репозиторию впоследствии его можно использовать вместо указания полного пути (рисунок 146).

```
shared C:/Users/ITdiakon/dev/hello/cloned_learning.git (push)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git fetch pbh
From https://github.com/EminKafarov/hello-world
* [new branch] master -> pbh/master
* [new branch] style -> pbh/style
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$
```

Рисунок 144 - Использование имени вместо пути

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

```
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$ git fetch
From https://github.com/EminKafarov/hello-world
6835207..dbd6ad8 master -> origin/master
Emin@kafarov22 MINGW64 ~/dev/hello/cloned_learning (master)
$
```

Рисунок 145 - Получение данных из удаленных проектов

Для отправки изменений в удаленный репозиторий используется команда git push (рисунок 148).

```
1 file changed, 2 insertions(+), 1 deletion(-)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 304 bytes | 304.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/EminKafarov/hello-world.git
dbd6ad8..b509ade master -> master
branch 'master' set up to track 'origin/master'.
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ |
```

Рисунок 146 - Отправка изменений в удаленный репозиторий

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git remote show origin
* remote origin
Fetch URL: https://EminKafarov@github.com:EminKafarov/hello-world.git
Push URL: https://EminKafarov@github.com:EminKafarov/hello-world.git
HEAD branch: master
Remote branches:
  master tracked
  style tracked
Local branch configured for 'git pull':
  master merges with remote master
Local refs configured for 'git push':
  master pushes to master (up to date)
  style pushes to style (up to date)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ |
```

Рисунок 147 - Информация об удаленном репозитории

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

```
style pushes to style (up to date)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git remote rename pbh new_name
Renaming remote references: 100% (2/2), done.
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git remote
new_name
origin
shared
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$
```

Рисунок 148 - Переименование удаленного репозитория

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

```
$ git remote remove new_name
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git remote
origin
shared
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$
```

Рисунок 149 - Удаление удаленного репозитория

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git tag
v0.9
v1.0
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ |
```

Рисунок 150 - Просмотр тегов

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git tag -a v2.0 -m "my version 2"
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git tag
v0.9
v1.0
v2.0
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$
```

Рисунок 151 - Создание аннотированной метки

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

```
v2.0
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git show v2.0
tag v2.0
Tagger: EminKafarov <Eminkafarov22@mail.ru>
Date: Mon Jun 16 17:51:15 2025 +0300

my version 2

commit b509ade05ca50c8791aa815892393970e6eb6b1e (HEAD -> master, tag: v2.0, origin/master)
Author: EminKafarov <Eminkafarov22@mail.ru>
Date: Mon Jun 16 17:46:27 2025 +0300

    NEW NEW NEW

diff --git a/README b/README
index 49b3ead..95f3b85 100644
--- a/README
+++ b/README
@@ -1,4 +1,5 @@
 Учим git...
 Изменено!
 changed in the original and pushed to shared
 -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
 +NEW NEW NEW NEW
 \ No newline at end of file
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ |
```

Рисунок 152 - 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).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git tag v3.0-wow
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git tag
v0.9
v1.0
v2.0
v3.0-wow
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$
```

Рисунок 153 - Создание легковесной метки

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git show v3.0-wow
commit b509ade05ca50c8791aa815892393970e6eb6b1e (HEAD -> master, tag: v3.0-wow, tag: v2.0, origin/master)
Author: EminKafarov <EminKafarov22@mail.ru>
Date: Mon Jun 16 17:46:27 2025 +0300

    NEW NEW NEW

diff --git a/README b/README
index 49b3ead..95f3b85 100644
--- a/README
+++ b/README
@@ -1,4 +1,5 @@
 Учим git...
 Изменено!
 changed in the original and pushed to shared
-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
+NEW NEW NEW NEW
 \ No newline at end of file
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$
```

Рисунок 154 - Просмотр данных тега

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git hist
* b509ade 2025-06-16 | NEW NEW NEW (HEAD -> master, tag: v3.0-wow, tag: v2.0, origin/master) [EminKafarov]
* dbd6ad8 2025-06-16 | Merge branch 'master' of c:/Users/ITdiakon/dev/hello/cloned_learning (shared/master) [EminKafarov]
* 841eef5 2025-06-16 | Added shared comment to README [EminKafarov]
* 28a23b4 2025-06-16 | Changed README in original repo [EminKafarov]
* ef522c9 2025-06-16 | Added shared comment to README [EminKafarov]
* 6835207 2025-06-16 | Changed README [EminKafarov]
* 4dcc6e1 2025-06-16 | Linked stylesheet in index.html (origin/style, style) [Kafarov Emin]
* 79c2e59 2025-06-16 | hello.html use style.css [Kafarov Emin]
* 9b0cb5d 2025-06-16 | Added stylesheet [Kafarov Emin]
* a4ad6ba 2025-06-16 | Life os great! [Kafarov Emin]
* 933fa53 2025-06-16 | Added README [Kafarov Emin]
* 813e4d2 2025-06-16 | New commit [Kafarov Emin]
* f230c94 2025-06-16 | Moved hello.html to lib [Kafarov Emin]
* 2d99b79 2025-06-16 | Add na author/email comment [Kafarov Emin]
* 1c72d3b 2025-06-16 | second change (tag: v1.0) [Kafarov Emin]
* 3339bcb 2025-06-16 | first change (tag: v0.9) [Kafarov Emin]
* 823e62b 2025-06-16 | Added new line [Kafarov Emin]
* 39fcce9 2025-06-16 | added file3 [Kafarov Emin]
* ae9af14 2025-06-16 | added file1 and file2 [Kafarov Emin]
* 53634d6 2025-06-16 | new commit [Kafarov Emin]
* 0411459 2025-06-16 | First Commit [Kafarov Emin]
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ AC
```

Рисунок 155 - Просмотр истории

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git tag -a v41.0 39fcce9
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$
```

Рисунок 156 - Создание тега определенному коммиту

```
MINGW64:/c/Users/ITdiakon/dev/hello/network_hello
GNU nano 7.2
new tag new tag
#
# Write a message for tag:
#   v41.0
# Lines starting with '#' will be ignored.
```

Рисунок 157 - Ввод сообщения в текстовом редакторе

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git show v41.0
tag v41.0
Tagger: EminKafarov <Eminkafarov22@mail.ru>
Date:   Mon Jun 16 17:57:49 2025 +0300

new tag new tag

commit 39fcce99447ed43b35c954a8eea05e2d832174e0 (tag: v41.0, tag: v40.0)
Author: Kafarov Emin <Eminkafarov22@mail.ru>
Date:   Mon Jun 16 08:19:06 2025 +0300

    added file3

diff --git a/file3.txt b/file3.txt
new file mode 100644
index 0000000..e69de29
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ |
```

Рисунок 158 - Просмотр данных тега

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git push shared v41.0
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 163 bytes | 81.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To C:/Users/ITdiakon/dev/hello/cloned_learning.git
 * [new tag]          v41.0 -> v41.0
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$
```

Рисунок 159 - Отправка тега на удаленный сервер

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

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git push shared --tags
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 698 bytes | 349.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
To C:/Users/ITdiakon/dev/hello/cloned_learning.git
* [new tag]          v0.9 -> v0.9
* [new tag]          v1.0 -> v1.0
* [new tag]          v2.0 -> v2.0
* [new tag]          v3.0-wow -> v3.0-wow
* [new tag]          v4.0 -> v4.0
* [new tag]          v40.0 -> v40.0
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$

```

Рисунок 160 - Отправка всех тегов на сервер

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

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git tag -d v3.0-wow
Deleted tag 'v3.0-wow' (was b509ade)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$

```

Рисунок 161 - Удаление тега

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

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$ git push shared :refs/tags/v3.0-wow
To C:/Users/ITdiakon/dev/hello/cloned_learning.git
- [deleted]          v3.0-wow
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (master)
$

```

Рисунок 162 - 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).


```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (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 b509ade NEW NEW NEW

```

Рисунок 163 - Переключение на метку

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning ((HEAD
$ git checkout -b version2 v2.1
Switched to a new branch 'version2'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (version2)
$

```

Рисунок 164 - Создание новой ветки

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

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.co checkout
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.ci commit
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.st status
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.br branch
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.hist "log --pretty=format:%h %ad | %s%d [%an]" --graph --date=short"
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.type 'cat-file -t'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ git config --global alias.dump 'cat-file -p'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning
$ |

```

Рисунок 165 - Задание алиасов

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (version2)
$ git config --global alias.unstage 'reset HEAD --'

```

Рисунок 166 - Создание псевдонима исключения файла из индекса

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (version2)
$ git unstage fileA
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (version2)
$ git reset HEAD -- fileA
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (version2)
$
```

Рисунок 167 - Использование созданного псевдонима

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (version2)
$ git config --global alias.last 'log -n 1'
```

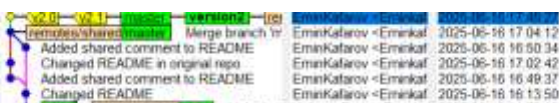
Рисунок 168 - Создание алиаса для просмотра последнего коммита

```
fatal: 'git' command not found
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (version2)
$ git last
commit b509ade05ca50c8791aa815892393970e6eb6b1e (HEAD -> version2, tag: v2.1, tag: v2.0, origin/master, master)
Author: EminKafarov <Eminkafarov22@mail.ru>
Date: Mon Jun 16 17:46:27 2025 +0300

    NEW NEW NEW
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (version2)
$
```

Рисунок 169 - Результат работы созданного алиаса

```
fatal: 'gitk' command not found
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (version2)
$ git config --global alias.visual 'gitk'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/learning (version2)
$ git visual
```



Commit	Author	Date
2025-06-16 17:04:12	EminKafarov <Eminkafarov22@mail.ru>	2025-06-16 17:04:12
2025-06-16 16:50:34	EminKafarov <Eminkafarov22@mail.ru>	2025-06-16 16:50:34
2025-06-16 17:02:42	EminKafarov <Eminkafarov22@mail.ru>	2025-06-16 17:02:42
2025-06-16 16:48:37	EminKafarov <Eminkafarov22@mail.ru>	2025-06-16 16:48:37
2025-06-16 16:13:50	EminKafarov <Eminkafarov22@mail.ru>	2025-06-16 16:13:50

Рисунок 170 - Создание псевдонима внешней команды

УП_8

Работа с Git

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


```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop ()
$ cd new_folder/
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder ()
$ touch README
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder ()
$ touch file1.txt
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder ()
$ touch file2.txt
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder ()
$ git add .
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder ()
$ git branch -a
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder ()
$ AC
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder ()
$ git commit -m "First commit"
[master (root-commit) 481e3c6] First commit
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README
create mode 100644 file1.txt
create mode 100644 file2.txt
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$

```

Рисунок 1 - Индексация и коммит 3 файлов

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

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ git co -b testing
Switched to a new branch 'testing'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (testing)
$ |

```

Рисунок 3 - Переключение на ветку testing

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

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (testing)
$ git add . && git commit -m "changed test.rb in testing"
[testing ba4e7a4] changed test.rb in testing
1 file changed, 1 insertion(+)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (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).

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (testing)
$ git co master
Switched to branch 'master'

```

Рисунок 5 - Переключение на ветку master

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ mv file1.txt test.rb
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ ls
README  file2.txt  test.rb
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ git add . && git commit -m "changed test.rb"
[master 9712cc7] changed test.rb
 2 files changed, 1 insertion(+)
 delete mode 100644 file1.txt
 create mode 100644 test.rb
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ |

```

Рисунок 6 - Еще индексация и коммит test.rb

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

```

1 file changed, 1 insertion(+)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (testing)
$ git co -b iss53
Switched to a new branch 'iss53'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (iss53)
$

```

Рисунок 7 - Создание и переключение на ветку iss53

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

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (iss53)
$ git add . && git commit -m "changed index.html in testing"
[iss53 634b6e0] changed index.html in testing
 1 file changed, 1 insertion(+)
 create mode 100644 index.html
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (iss53)
$

```

Рисунок 8 - Индексация и коммит файла index.html

Далее нужно переключить ветку на master (рисунок 9).

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (iss53)
$ git co master
Switched to branch 'master'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (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).

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (hotfix)
$ git add . && git commit -m "changed index.html in hotfix"
[hotfix eb53d53] changed index.html in hotfix
1 file changed, 1 insertion(+)
create mode 100644 index.html
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (hotfix)

```

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

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (iss53)
$ git co master
Switched to branch 'master'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ git merge hotfix
Updating 9712cc7..eb53d53
Fast-forward
 index.html | 1 +
1 file changed, 1 insertion(+)
create mode 100644 index.html
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$

```

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

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

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ git br -d hotfix
Deleted branch hotfix (was eb53d53).
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ |

```

Рисунок 12 - Удаление ветки hotfix

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

```

Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (iss53)
$ git commit --amend -m "changed index.html in iss53"
[iss53 e7388fd] changed index.html in iss53
Date: Mon Jun 16 18:36:01 2025 +0300
1 file changed, 1 insertion(+)
create mode 100644 index.html
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (iss53)
$ git co master
Switched to branch 'master'

```

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ git merge iss53
Updating 82acbce..0976d6d
Fast-forward
 index.html | 2 ++
 test.rb    | 2 ++
 2 files changed, 4 insertions(+)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ |
```

Рисунок 14 - Слияние веток

После этого ветку iss53 нужно удалить (рисунок 15).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ git br -d iss53
Deleted branch iss53 (was 0976d6d).
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ |
```

Рисунок 15 - Удаление ветки iss53

УП_9

Работа с Git

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ git branch
* master
  testing
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ |
```

Рисунок 171 - Список существующих веток

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ git branch -v
* master 7727597 fix space
  testing 0976d6d changed
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$
```

Рисунок 172 - 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).

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ git branch --merged
* master
testing
```

Рисунок 173 - Список веток слитых с текущей

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ git branch --no-merged
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$
```

Рисунок 174 - Список веток не слитых с текущей

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ git branch --d testing
Deleted branch testing (was 0976d6d).
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$
```

Рисунок 175 - Удаление ветки

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ git remote show origin
* remote origin
Fetch URL: https://github.com/EminKafarov/hello-2.git
Push URL: https://github.com/EminKafarov/hello-2.git
HEAD branch: master
Remote branch:
  master 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)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ |
```

Рисунок 176 - Просмотр удаленных веток

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/new_folder (master)
$ git push -u origin master
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (12/12), 914 bytes | 457.00 KiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/EminKafarov/hello-2.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Рисунок 177 - Отправка изменений

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (master)
$ git fetch origin
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (5/5), 373 bytes | 21.00 KiB/s, done.
From https://github.com/EminKafarov/hello-2
 7727597..6ba7b08 master    -> origin/master
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (master)
$ |
```

Рисунок 178 - Выполнение команды git fetch

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (master)
$ git co -b serv origin/testing
Switched to a new branch 'serv'
branch 'serv' set up to track 'origin/testing'.
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (serv)
$
```

Рисунок 179 - Создание ветки на основе удаленной ветки

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

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (serv)
$ git push origin :testing
To https://github.com/EminKafarov/hello-2.git
- [deleted]      testing
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (serv)
$
```

Рисунок 180 - 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.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (master)
$ git co exp
Switched to branch 'exp'
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (exp)
$ git rebase master
Successfully rebased and updated refs/heads/exp.
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (exp)
$
```

Рисунок 181 - 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.

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (cl)
$ git rebase --onto master se cl
Successfully rebased and updated refs/heads/cl.
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (cl)
$ |
```

Рисунок 182 - Перемещение изменений с параметром onto

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (cl)
$ git co master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 5 commits.
(use "git push" to publish your local commits)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (master)
$ git merge cl
Updating 98272d8..bef73b4
Fast-forward
 w.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (master)
$
```

Рисунок 183 - Слияние веток master и cl

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (master)
$ git rebase master se
Successfully rebased and updated refs/heads/se.
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (se)
$ |
```


Рисунок 184 - Перемещение изменений

```
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (se)
$ git co master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 6 commits.
  (use "git push" to publish your local commits)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (master)
$ git merge se
Updating bef73b4..8e01b9c
Fast-forward
 w.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (master)
$ |
```

Рисунок 185 - Слияние веток master и se

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

```
1 file changed, 1 insertion(+), 1 deletion(-)
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (master)
$ git br -d se
Deleted branch se (was 8e01b9c).
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (master)
$ git br -d cl
Deleted branch cl (was bef73b4).
Emin@kafarov22 MINGW64 ~/OneDrive/Desktop/repo/hello-2 (master)
$ |
```

Рисунок 186 - Удаление веток cl и se