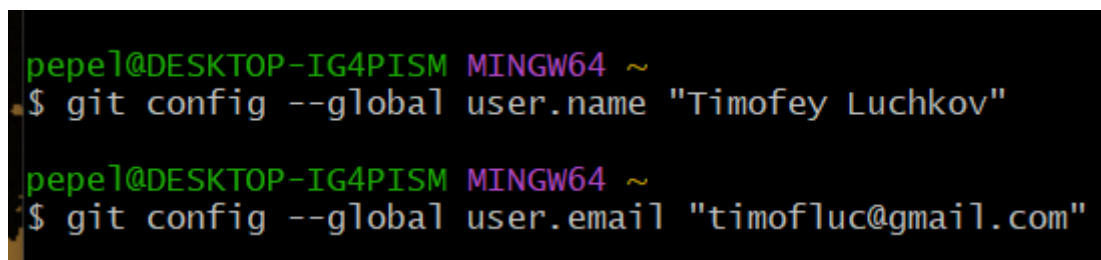


УП_1

Работа с Git

Подготовка

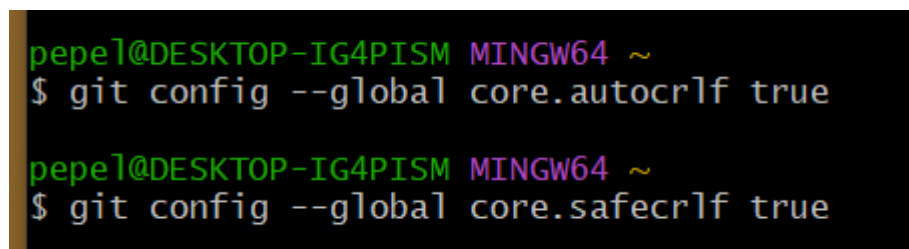
Если git никогда ранее не использовался пользователем, то необходимо установить имя и электронную почту. Для этого требуется выполнить команды, показанные на рисунке 1.



```
pepel@DESKTOP-IG4PISM MINGW64 ~  
$ git config --global user.name "Timofey Luchkov"  
  
pepel@DESKTOP-IG4PISM MINGW64 ~  
$ git config --global user.email "timofluc@gmail.com"
```

Рисунок 1 - Установка имени и электронной почты

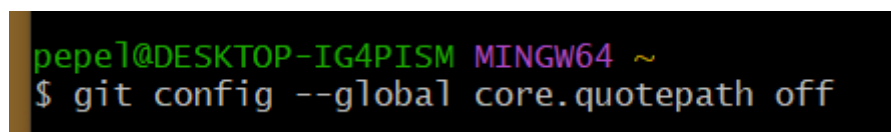
Затем нужно указать параметры установки окончаний строк (рисунок 2).



```
pepel@DESKTOP-IG4PISM MINGW64 ~  
$ git config --global core.autocrlf true  
  
pepel@DESKTOP-IG4PISM MINGW64 ~  
$ git config --global core.safecrlf true
```

Рисунок 2 - Параметры окончаний строк

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



```
pepel@DESKTOP-IG4PISM MINGW64 ~  
$ git config --global core.quotePath off
```

Рисунок 3 - Установка отображения unicode

Создание проекта

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~  
$ cd desktop  
  
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop  
$ mkdir uchebnaya.praktika  
  
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop  
$ cd uchebnaya.praktika  
  
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika  
$ touch hello.html
```

Рисунок 4 - Создание каталога и файла

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

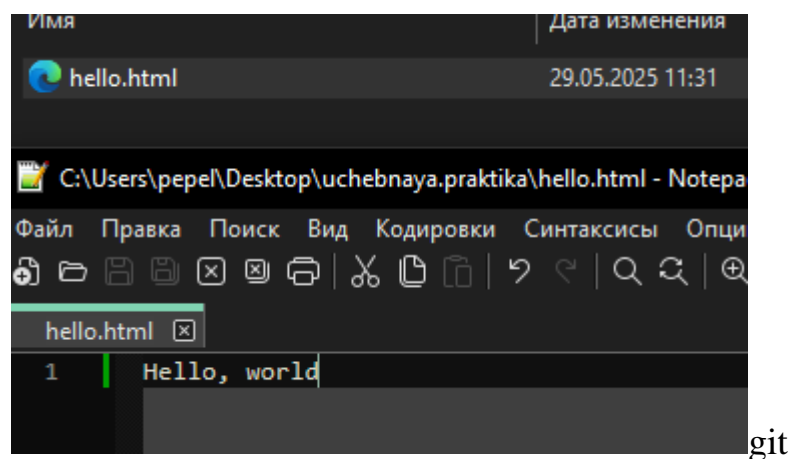


Рисунок 5 - Содержание файла

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika  
$ git init  
Initialized empty Git repository in C:/Users/pepe1/Desktop/uchebnaya.praktika/.git/
```

Рисунок 6 - Создание репозитория

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git add hello.html

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git commit -m "first commit"
[master (root-commit) 4258025] first commit
1 file changed, 1 insertion(+)
create mode 100644 hello.html
```

Рисунок 7 - Добавление в репозиторий

Проверка состояния

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Рисунок 8 - Проверка состояния репозитория

Внесение изменений

Сначала необходимо внести изменения в файл (рисунок 9).

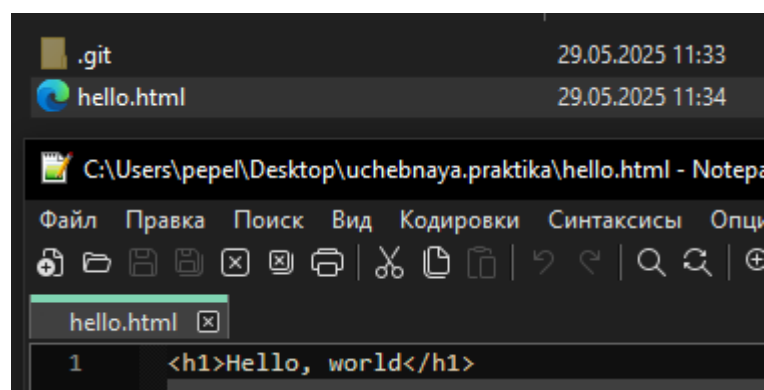


Рисунок 9 - Внесение изменений в файл

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.html

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

Рисунок 10 - Сообщение о незафиксированных изменениях

Индексация изменений

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git add hello.html

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

Рисунок 11 - Команды для индексации изменений

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

Индексация и коммит

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



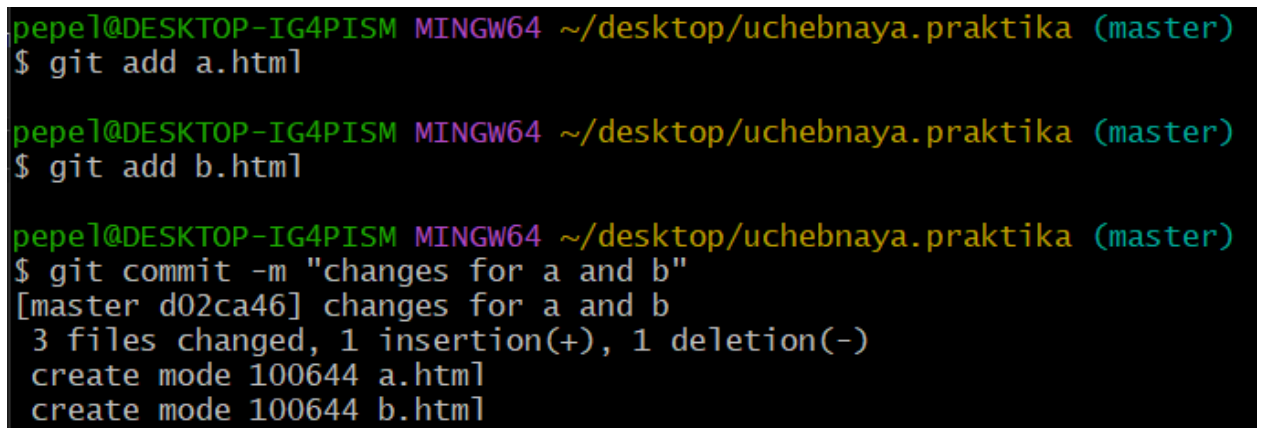
The image shows a file explorer window on the left with a list of files: .git, a.html, b.html, hello.html, and v.html. On the right, a terminal window shows the following commands and output:

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ touch a.html

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ touch b.html

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ touch v.html
```

Рисунок 12 - Создано 3 файла



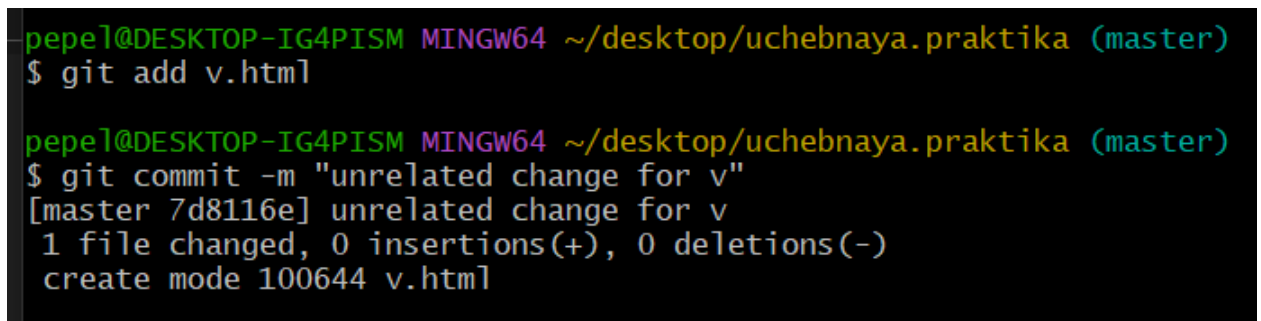
The image shows a terminal window with the following commands and output:

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git add a.html

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git add b.html

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git commit -m "changes for a and b"
[master d02ca46] changes for a and b
3 files changed, 1 insertion(+), 1 deletion(-)
create mode 100644 a.html
create mode 100644 b.html
```

Рисунок 13 - Индексация и коммит для 2 файлов



The image shows a terminal window with the following commands and output:

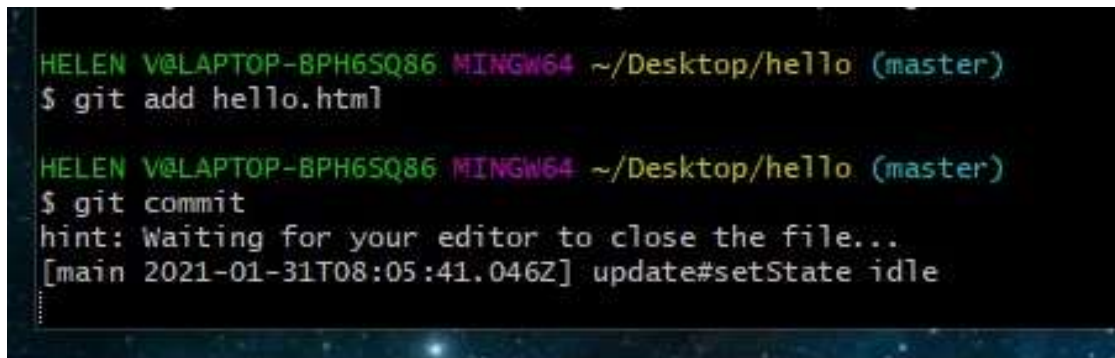
```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git add v.html

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git commit -m "unrelated change for v"
[master 7d8116e] unrelated change for v
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 v.html
```

Рисунок 14 - Индексация и коммит для третьего файла

Коммит изменений

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

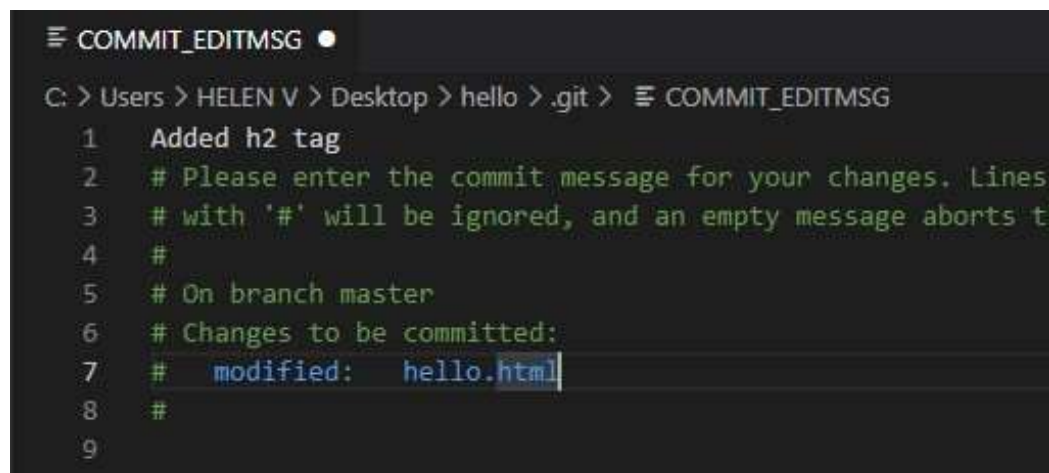


```
HELEN V@LAPTOP-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$ git add hello.html

HELEN V@LAPTOP-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$ git commit
hint: Waiting for your editor to close the file...
[main 2021-01-31T08:05:41.046Z] update#setState idle
```

Рисунок 15 - Коммит изменений

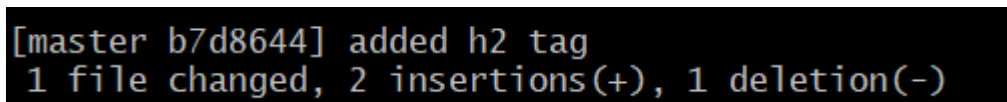
После этого будет открыт редактор, в котором необходимо на 1 строке записать комментарий коммита (рисунок 16).



```
COMMIT_EDITMSG
C: > Users > HELEN V > Desktop > hello > .git > COMMIT_EDITMSG
1 Added h2 tag
2 # Please enter the commit message for your changes. Lines
3 # with '#' will be ignored, and an empty message aborts t
4 #
5 # On branch master
6 # Changes to be committed:
7 #   modified:   hello.html
8 #
9
```

Рисунок 16 - Ввод комментария

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



```
[master b7d8644] added h2 tag
1 file changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 17 - Коммит-сообщения

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

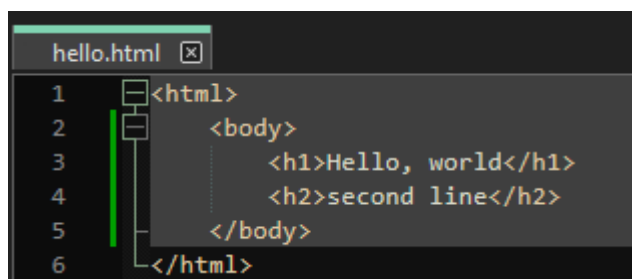
```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Рисунок 18 - Проверка состояния

Изменения, а не файлы

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

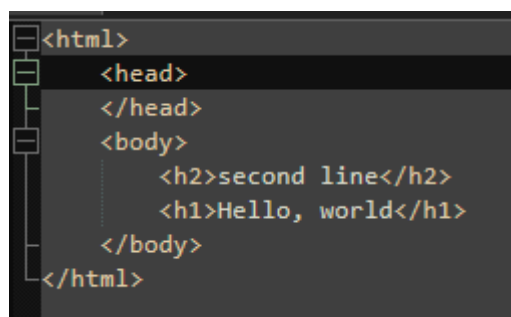
Сначала нужно в файл hello.html добавить теги html и body (рисунок 19), а затем проиндексировать изменения.



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

Рисунок 19 - Добавление тегов html и body

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



```
<html>
  <head>
  </head>
  <body>
    <h2>second line</h2>
    <h1>Hello, world</h1>
  </body>
</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.


```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

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

Рисунок 21 - Проверка состояния

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git commit -m "add standart html tags"
[master 574d9da] add standart html tags
1 file changed, 5 insertions(+), 1 deletion(-)

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.html

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

Рисунок 22 - Коммит 1 изменения и проверка состояния

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html
```

g it

Рисунок 23 - Индексация 2 изменения и проверка состояния

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git commit -m "add html header"
[master 88767cb] add html header
1 file changed, 2 insertions(+)
```

Рисунок 24 - Коммит 2 изменения

УП_2

Работа с Git

История

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git log
commit 88767cb3d6ae7a458e2d1ff2cea69eb3d1a9acbd (HEAD -> master)
Author: Timofey Luchkov <timofluc@gmail.com>
Date: Thu May 29 11:59:27 2025 +0300

    add html header

commit 574d9da356e2cac359407fc1168d79a1dbe664bc
Author: Timofey Luchkov <timofluc@gmail.com>
Date: Thu May 29 11:58:13 2025 +0300

    add standart html tags

commit 24f23e460536b5b45406b88c559d45bd5e936913
Author: Timofey Luchkov <timofluc@gmail.com>
Date: Thu May 29 11:51:27 2025 +0300

    a

commit a4d1a8baf62d474abed7dce4c5b7ccc7e1ca5717
Author: Timofey Luchkov <timofluc@gmail.com>
Date: Thu May 29 11:49:15 2025 +0300

    added html and body tag

commit b7d86448a45f4f81ce145965def12e1d7f814399
Author: Timofey Luchkov <timofluc@gmail.com>
Date: Thu May 29 11:44:43 2025 +0300

    added h2 tag

commit 46ec5b7828bbf2135e899f4acb71aecdd4e520eaa
Author: Timofey Luchkov <timofluc@gmail.com>
Date: Thu May 29 11:42:36 2025 +0300

    Added h2 tag

commit 7d8116eed054fde1fd293f9bd4a89f2b4c282bb6
Author: Timofey Luchkov <timofluc@gmail.com>
Date: Thu May 29 11:40:42 2025 +0300

    unrelated change for v

commit d02ca4653d8f83c4ecf3702bb96cee72d86076aa
Author: Timofey Luchkov <timofluc@gmail.com>
Date: Thu May 29 11:39:39 2025 +0300

    changes for a and b

commit 4258025655e18e45440c7968d1864bee47b30f41
Author: Timofey Luchkov <timofluc@gmail.com>
Date: Thu May 29 11:33:12 2025 +0300

    first commit
```

Рисунок 25 - Просмотр истории изменений

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

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git log --pretty=oneline
88767cb3d6ae7a458e2d1ff2cea69eb3d1a9acbd (HEAD -> master) add html header
574d9da356e2cac359407fc1168d79a1dbe664bc add standart html tags
24f23e460536b5b45406b88c559d45bd5e936913 a
a4d1a8baf62d474abed7dce4c5b7ccc7e1ca5717 added html and body tag
b7d86448a45f4f81ce145965def12e1d7f814399 added h2 tag
46ec5b7828bbf2135e899f4acb71aec4e520eaa Added h2 tag
7d8116eed054fde1fd293f9bd4a89f2b4c282bb6 unrelated change for v
d02ca4653d8f83c4ecf3702bb96cee72d86076aa changes for a and b
4258025655e18e45440c7968d1864bee47b30f41 first commit

```

Рисунок 26 - Однострочный формат вывода

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

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git log --pretty=oneline --max-count=2
88767cb3d6ae7a458e2d1ff2cea69eb3d1a9acbd (HEAD -> master) add html header
574d9da356e2cac359407fc1168d79a1dbe664bc add standart html tags

```

Рисунок 27 - Вывод последних 2 изменений

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git log --pretty=oneline --since='10 days ago'
88767cb3d6ae7a458e2d1ff2cea69eb3d1a9acbd (HEAD -> master) add html header
574d9da356e2cac359407fc1168d79a1dbe664bc add standart html tags
24f23e460536b5b45406b88c559d45bd5e936913 a
a4d1a8baf62d474abed7dce4c5b7ccc7e1ca5717 added html and body tag
b7d86448a45f4f81ce145965def12e1d7f814399 added h2 tag
46ec5b7828bbf2135e899f4acb71aec4e520eaa Added h2 tag
7d8116eed054fde1fd293f9bd4a89f2b4c282bb6 unrelated change for v
d02ca4653d8f83c4ecf3702bb96cee72d86076aa changes for a and b
4258025655e18e45440c7968d1864bee47b30f41 first commit

```

Рисунок 28 - Вывод изменений начиная с определенного времени

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git log --pretty=oneline --until='5 minutes ago'
574d9da356e2cac359407fc1168d79a1dbe664bc add standart html tags
24f23e460536b5b45406b88c559d45bd5e936913 a
a4d1a8baf62d474abed7dce4c5b7ccc7e1ca5717 added html and body tag
b7d86448a45f4f81ce145965def12e1d7f814399 added h2 tag
46ec5b7828bbf2135e899f4acb71aec4e520eaa Added h2 tag
7d8116eed054fde1fd293f9bd4a89f2b4c282bb6 unrelated change for v
d02ca4653d8f83c4ecf3702bb96cee72d86076aa changes for a and b
4258025655e18e45440c7968d1864bee47b30f41 first commit

```

Рисунок 29 - Вывод изменений до определенного времени

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git log --pretty=oneline --author='Timofey Luchkov'
88767cb3d6ae7a458e2d1ff2cea69eb3d1a9acbd (HEAD -> master) add html header
574d9da356e2cac359407fc1168d79a1dbe664bc add standart html tags
24f23e460536b5b45406b88c559d45bd5e936913 a
a4d1a8baf62d474abed7dce4c5b7ccc7e1ca5717 added html and body tag
b7d86448a45f4f81ce145965def12e1d7f814399 added h2 tag
46ec5b7828bbf2135e899f4acb71aec4e520eaa Added h2 tag
7d8116eed054fde1fd293f9bd4a89f2b4c282bb6 unrelated change for v
d02ca4653d8f83c4ecf3702bb96cee72d86076aa changes for a and b
4258025655e18e45440c7968d1864bee47b30f41 first commit

```

Рисунок 30 - Вывод изменений, внесенных определенным автором

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git log --pretty=oneline --all
88767cb3d6ae7a458e2d1ff2cea69eb3d1a9acbd (HEAD -> master) add html header
574d9da356e2cac359407fc1168d79a1dbe664bc add standart html tags
24f23e460536b5b45406b88c559d45bd5e936913 a
a4d1a8baf62d474abed7dce4c5b7ccc7e1ca5717 added html and body tag
b7d86448a45f4f81ce145965def12e1d7f814399 added h2 tag
46ec5b7828bbf2135e899f4acb71aec4e520eaa Added h2 tag
7d8116eed054fde1fd293f9bd4a89f2b4c282bb6 unrelated change for v
d02ca4653d8f83c4ecf3702bb96cee72d86076aa changes for a and b
4258025655e18e45440c7968d1864bee47b30f41 first commit

```

Рисунок 31 - Вывод всех изменений

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git log --all --pretty=format:"%h %cd %s (%an)" --since='12 days ago'
88767cb Thu May 29 11:59:27 2025 +0300 add html header (Timofey Luchkov)
574d9da Thu May 29 11:58:13 2025 +0300 add standart html tags (Timofey Luchkov)
24f23e4 Thu May 29 11:51:27 2025 +0300 a (Timofey Luchkov)
a4d1a8b Thu May 29 11:49:15 2025 +0300 added html and body tag (Timofey Luchkov)
b7d8644 Thu May 29 11:44:43 2025 +0300 added h2 tag (Timofey Luchkov)
46ec5b7 Thu May 29 11:42:36 2025 +0300 Added h2 tag (Timofey Luchkov)
7d8116e Thu May 29 11:40:42 2025 +0300 unrelated change for v (Timofey Luchkov)
d02ca46 Thu May 29 11:39:39 2025 +0300 changes for a and b (Timofey Luchkov)
4258025 Thu May 29 11:33:12 2025 +0300 first commit (Timofey Luchkov)

```

Рисунок 32 - Использование нескольких параметров

Алиасы

Для настройки алиасов используется команда, показанная на рисунке 9.

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git config --global alias.co checkout

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git config --global alias.ci commit

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git config --global alias.st status

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git config --global alias.br branch

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git config --global alias.hist "log --pretty=format:'%h %a | %s%d [%an]' --graph ==date=short"

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git config --global alias.type 'cat-file -t'

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git config --global alias.dump 'cat-file -p'

```

Рисунок 33 - Настройка алиасов для некоторых команд

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git config --global alias.hist "log --pretty=format:'%h %a | %s%d [%an]' --graph --date=short"

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist
* 88767cb %a | add html header (HEAD -> master) [Timofey Luchkov]
* 574d9da %a | add standart html tags [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]

```

Рисунок 34 - Выполнение алиаса hist

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ alias gs='git status'

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ gs
On branch master
nothing to commit, working tree clean

```

Рисунок 35 - Установка и выполнение алиаса gs

Получение старых версий

Для того, чтобы вернуть рабочий каталог к предыдущему состоянию, можно использовать следующий способ: для начала нужно узнать хэши

предыдущих версий, что можно сделать с помощью ранее заданного алиаса `hist` (рисунок 12).

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist
* 88767cb %a | add html header (HEAD -> master) [Timofey Luchkov]
* 574d9da %a | add standart html tags [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]
```

Рисунок 36 - Просмотр хэшей предыдущих версий

Далее нужно выполнить команду `git checkout` с номером нужного хэша (достаточно первых 7 знаков). После этого можно просмотреть содержимое файла с помощью команды `cat` (рисунок 13).

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git checkout 4258025
Note: switching to '4258025'.

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 4258025 first commit
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika ((4258025...))
$ cat hello.html
Hello, world
```

Рисунок 37 - Возвращение к нужной версии и просмотр содержимого файла

Возвращение к последней версии в ветке `master`

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


```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika ((4258025...))
$ git checkout master
Previous HEAD position was 4258025 first commit
Switched to branch 'master'

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h2>second line</h2>
    <h1>Hello, world</h1>
  </body>
</html>

```

Рисунок 38 - Возвращение к последней версии в ветке master

Создание тегов версий

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

```

HELEN V@LAPTOP-BPH65Q86 MINGW64 ~/Desktop/hello (master)
$ git tag ver1

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

```

Рисунок 39 - Задание тега

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


```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git checkout ver1^
Note: switching to 'ver1^'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false
HEAD is now at 574d9da add standart html tags
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika ((574d9da...))
$ cat hello.html
<html>
  <body>
    <h2>second line</h2>
    <h1>Hello, world</h1>
  </body>
</html>

```

Рисунок 40 – Переход к предыдущей версии с помощью тега

```

    <h1>Hello, world</h1>
  </body>
</html>
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika ((574d9da...))
$ git tag ver1-beta

```

Рисунок 41 - Задание тега предыдущей версии

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

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika ((ver1-beta))
$ git checkout ver1
Previous HEAD position was 574d9da add standart html tags
HEAD is now at 88767cb add html header
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika ((ver1))
$ git checkout ver1-beta
Previous HEAD position was 88767cb add html header
HEAD is now at 574d9da add standart html tags

```

Рисунок 42 - Переключение между версиями с помощью тегов

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika ((ver1-beta))
$ git tag
ver1
ver1-beta
```

Рисунок 43 - Просмотр тегов

Также можно просмотреть теги в логе, как показано на рисунке 20.

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika ((ver1-beta))
$ git hist master --all
* 88767cb %a | add html header (tag: ver1, master) [Timofey Luchkov]
* 574d9da %a | add standart html tags (HEAD, tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]
```

Рисунок 44 - Просмотр тегов в логе

Отмена локальных изменений (до индексации)

Сначала нужно переключиться на последний коммит master (рисунок 21).

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika ((ver1-beta))
$ git checkout master
Previous HEAD position was 574d9da add standart html tags
Switched to branch 'master'
```

Рисунок 45 - Переключение на последний коммит

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

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

Рисунок 46 - Внесение изменения в файл

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

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git st
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.html

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

```

Рисунок 47 - Не проиндексированное изменение

Для переключения в версию файла без изменений используется команда `git checkout hello.html` (рисунок 24). Команда `git status` покажет, что не было произведено изменений, не зафиксированных в каталоге.

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git checkout hello.html
Updated 1 path from the index

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git st
On branch master
nothing to commit, working tree clean

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h2>second line</h2>
    <h1>Hello, world</h1>
  </body>
</html>

```

Рисунок 48 - Возвращение к версии

Отмена проиндексированных изменений (перед коммитом)

Для того, чтобы научиться отменять проиндексированные изменения, сначала нужно внести ненужное изменение в файл (рисунок 25). После этого производится индексация (рисунок 26).

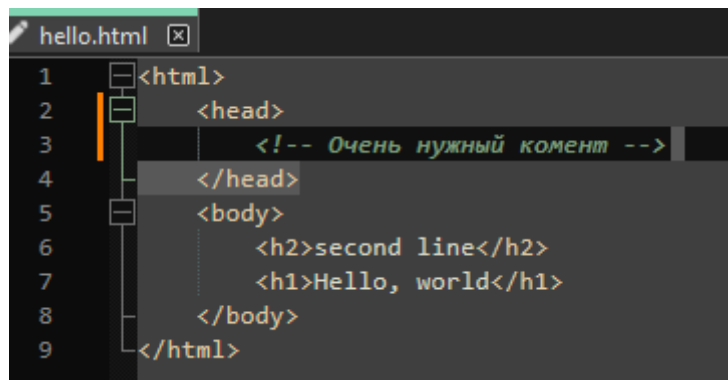


Рисунок 49 - Внесение ненужного изменения

```
</html>
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git add hello.html

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git st
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

Рисунок 50 - Индексация изменения

Для отмены индексация изменения используется команда `git reset HEAD hello.html` (рисунок 27). Команда `reset` сбрасывает буферную зону к HEAD и очищает ее от проиндексированных изменений. Но для удаления ненужного по-прежнему используется команда `git checkout` (рисунок 28).

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git reset HEAD hello.html
Unstaged changes after reset:
M      hello.html

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git st
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.html

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

Рисунок 51 - Очистка буферной зоны

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git checkout hello.html
Updated 1 path from the index

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git st
On branch master
nothing to commit, working tree clean

```

Рисунок 52 - Удаление ненужных изменений

Отмена коммитов

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

Для начала надо внести изменение, проиндексировать его и записать коммит (рисунки 29-30).

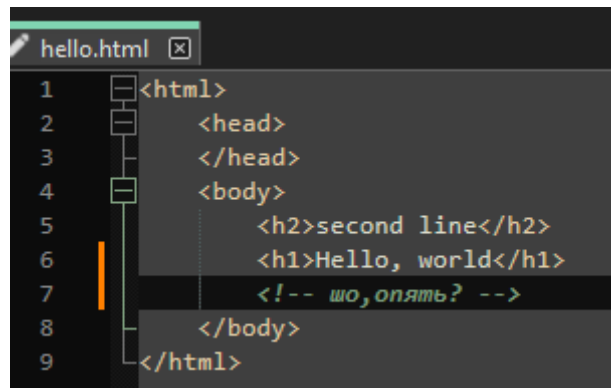


Рисунок 53 - Внесение изменения в файл

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git add hello.html

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git commit -m "bad commit"
[master d79b8d2] bad commit
1 file changed, 1 insertion(+)

```

Рисунок 54 - Индексация и коммит

Для создания коммита, который удалит ненужные изменения, используется команда `git revert HEAD` (рисунок 31). После этого будет открыт редактор, в котором можно отредактировать коммит сообщение (рисунок 32), затем надо сохранить файл и закрыть редактор (рисунок 33).

```
HELEN V@LAPTOP-BPH6SQ86 MINGW64 ~/Desktop/hello (master)
$ git revert HEAD
hint: Waiting for your editor to close the file...
(electron) Sending uncompressed crash reports is deprecated and will be removed in a future version of Electron. Set { compress: true } to opt-in to the new behavior. Crash reports will be uploaded gzipped, which most crash reporting servers support.
[main 2021-02-13T20:03:20.025Z] update#setState idle
(node:11232) electron: The default of contextIsolation is deprecated and will be changing from false to true in a future release of Electron. See https://github.com/electron/electron/issues/23506 for more information
(node:11232) electron: The default of contextIsolation is deprecated and will be changing from false to true in a future release of Electron. See https://github.com/electron/electron/issues/23506 for more information
```

Рисунок 55 - Выполнение команды git revert

```
COMMIT_EDITMSG X
C: > Users > HELEN V > Desktop > hello > .git > COMMIT_EDITMSG
1  Revert "Oops, this is bad commit"
2
3  This reverts commit ce12d52b0101f30fefaf7c66203593dbfac57ae54.
4
5  # Please enter the commit message for your changes. Lines starting
6  # with '#' will be ignored, and an empty message aborts the commit.
7  #
8  # On branch master
9  # Changes to be committed:
10 #   modified:   hello.html
11 #
12
```

Рисунок 56 - Коммит сообщение в редакторе

```
(node:11232) electron: The default of contextIsolation is deprecated and will be changing from false to true in a future release of Electron. See https://github.com/electron/electron/issues/23506 for more information
[main 2021-02-13T20:03:50.027Z] update#setState checking for updates
[main 2021-02-13T20:03:50.085Z] update#setState idle
[master e04a55f] Revert "Oops, this is bad commit"
1 file changed, 1 deletion(-)

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

Рисунок 57 - Редактор закрыт

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

Рисунок 58 - Все коммиты при просмотре лога

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist
* 5537bad %a | Revert "bad commit" (HEAD -> master, tag: oooops) [Timofey Luchkov]
* d79b8d2 %a | bad commit [Timofey Luchkov]
* 225549b %a | Revert "bad commit" [Timofey Luchkov]
* 6f77f17 %a | bad commit [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]
```

Рисунок 59 - Отметка тегом коммита

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git reset --hard ver1
HEAD is now at 88767cb add html header

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist
* 88767cb %a | add html header (HEAD -> master, tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]
```

Рисунок 60 - Сброс коммита

Но при просмотре лога с помощью команды `git hist --all` отмененные коммиты по-прежнему будут показываться, так как они всё еще находятся в репозитории (рисунок 37).


```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist --all
* 5537bad %a | Revert "bad commit" (tag: oooops) [Timofey Luchkov]
* d79b8d2 %a | bad commit [Timofey Luchkov]
* 225549b %a | Revert "bad commit" [Timofey Luchkov]
* 6f77f17 %a | bad commit [Timofey Luchkov]
* 88767cb %a | add html header (HEAD -> master, tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]

```

Рисунок 61 - Сброшенные коммиты находятся по-прежнему в репозитории

Удаление тега

Так как тег «oooops» больше не нужен, его и коммиты, на которые он указывает, можно удалить с помощью команды `git tag -d` (рисунок 38).

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git tag -d oooops
Deleted tag 'oooops' (was 5537bad)

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist --all
* 88767cb %a | add html header (HEAD -> master, tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]

```

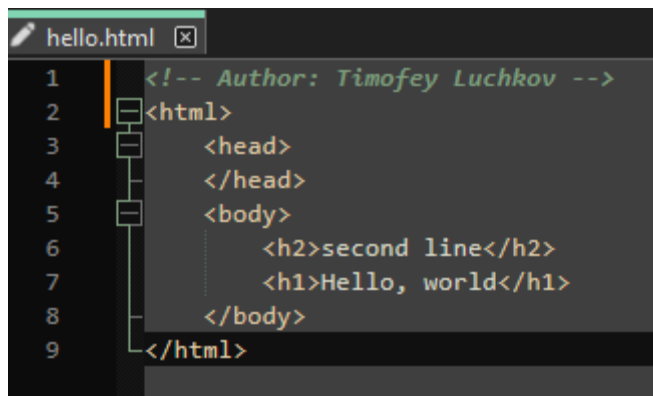
Рисунок 62 - Удаление тега

УП_3

Работа с Git

Внесение изменений в коммиты

Для начала будет создан коммит, в который позже будут внесены изменения. На рисунках 1 и 2 происходит добавление комментария в файл `hello.html` и его индексация и коммит.



```
1 <!-- Author: Timofey Luchkov -->
2 <html>
3   <head>
4   </head>
5   <body>
6     <h2>second line</h2>
7     <h1>Hello, world</h1>
8   </body>
9 </html>
```

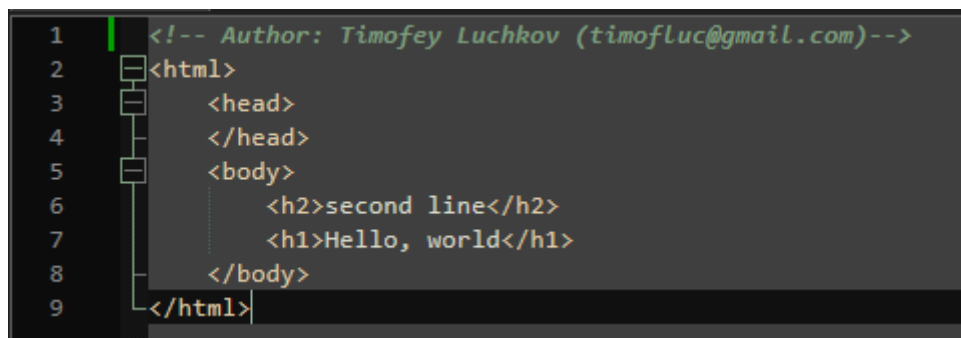
Рисунок 63 - Добавление комментария в файл

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git add hello.html

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git commit -m "Add an author comment"
[master e72f0cb] Add an author comment
1 file changed, 1 insertion(+)
```

Рисунок 64 - Индексация и коммит

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



```
1 <!-- Author: Timofey Luchkov (timofluc@gmail.com)-->
2 <html>
3   <head>
4   </head>
5   <body>
6     <h2>second line</h2>
7     <h1>Hello, world</h1>
8   </body>
9 </html>
```

Рисунок 65 - Добавление электронной почты

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git add hello.html

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git commit --amend -m "Add an author/email comment"
[master 1d7024f] Add an author/email comment
Date: Thu May 29 13:24:01 2025 +0300
1 file changed, 1 insertion(+)
```

Рисунок 66 - Индексация и изменение коммита

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist
* 1d7024f %a | Add an author/email comment (HEAD -> master) [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]
```

Рисунок 67 - Последний коммит изменен

Перемещение файлов

Для перемещения файлов в пределах репозитория используются команды, показанные на рисунке 6. После выполнения данных команды git индексирует эти изменения (удаление файла hello.html и создание файла lib/hello.html).

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ mkdir lib

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git mv hello.html lib

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:   hello.html -> lib/hello.html
```

Рисунок 68 - Перемещение файла

Далее надо осуществить коммит данного перемещения.

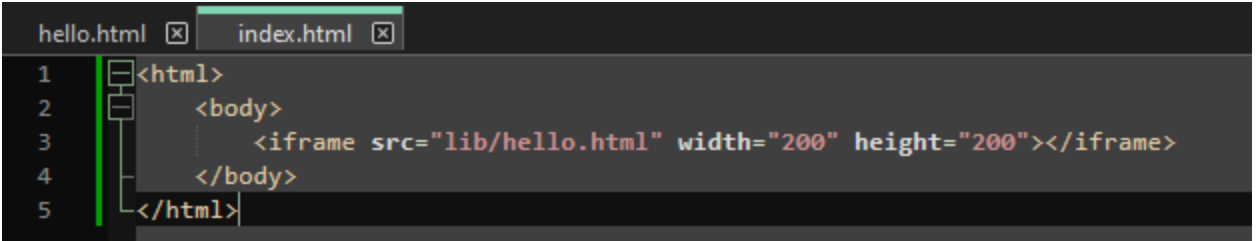
```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git commit -m "moved hello.html to lib"
[master 0e0be4b] moved hello.html to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)
```

Рисунок 69 – Коммит перемещения

[Подробнее о структуре](#)

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ touch index.html
```



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

Рисунок 70 - Содержимое файла `index.html`

Далее нужно проиндексировать и закоммитить файл (рисунок 9).

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git add index.html

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git commit -m "added index.html"
[master dc7f2db] added index.html
1 file changed, 5 insertions(+)
create mode 100644 index.html
```

Рисунок 71 - Индексация и коммит

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

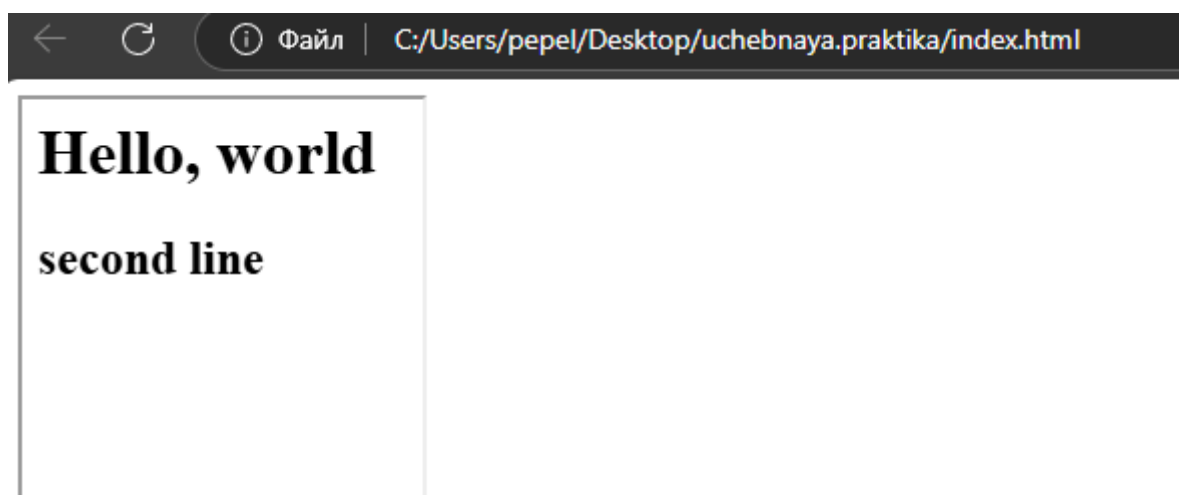


Рисунок 72 - Файл `index.html`, открытый в браузере

Каталог .git

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ ls -C .git
COMMIT_EDITMSG HEAD ORIG_HEAD config description hooks/ index info/ logs/ objects/ refs/
```

Рисунок 73 - Содержание каталога .git

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ ls -C .git/objects
05/ 0e/ 1d/ 24/ 36/ 46/ 55/ 5c/ 6e/ 77/ 7b/ 7e/ 8f/ a4/ b7/ d0/ d5/ d8/ dc/ e7/ f4/ f8/ pack/
0d/ 16/ 22/ 26/ 42/ 49/ 57/ 69/ 6f/ 78/ 7d/ 88/ 9b/ b0/ ce/ d4/ d7/ db/ e6/ f0/ f7/ info/
```

Рисунок 74 - Содержание каталога objects

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ ls -C .git/objects/8f
bb5aead8d8b0650186bf34555b5bec09dc6dc4
```

Рисунок 75 - Просмотр каталога 8d

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    ignorecase = true
```

Рисунок 76 - Просмотр файла конфигурации

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ ls .git/refs
heads/  tags/

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ ls .git/refs/heads
master

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ ls .git/refs/tags
ver1  ver1-beta

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ cat .git/refs/tags/ver1
88767cb3d6ae7a458e2d1ff2cea69eb3d1a9acbd

```

Рисунок 77 - Просмотр файлов и веток

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ cat .git/head
ref: refs/heads/master

```

Рисунок 78 - Содержимое файла HEAD

Работа с объектами git

Для начала необходимо просмотреть последний коммит (рисунок 17).

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist --max-count=1
* dc7f2db %a | added index.html (HEAD -> master) [Timofey Luchkov]

```

Рисунок 79 - Последний коммит

Далее надо использовать хэш последнего коммита, используя команды `cat-file -p` и `cat-file -t` (рисунок 18) для просмотра объекта коммита. Также вместо длинных команд можно использовать сокращенные `type` и `dump`, если данные команды были заданы как алиасы.


```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git cat-file -t dc7f2db
commit

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git cat-file -p dc7f2db
tree 0d9717544975aa67a6f46d44c14b2e68a7e81710
parent 0e0be4bf37eac0697d2b81d3cf4916346a0ab872
author Timofey Luchkov <timofluc@gmail.com> 1748514785 +0300
committer Timofey Luchkov <timofluc@gmail.com> 1748514785 +0300

added index.html

```

Рисунок 80 - Просмотр объекта коммита

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git type dc7f2db
commit

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git dump dc7f2db
tree 0d9717544975aa67a6f46d44c14b2e68a7e81710
parent 0e0be4bf37eac0697d2b81d3cf4916346a0ab872
author Timofey Luchkov <timofluc@gmail.com> 1748514785 +0300
committer Timofey Luchkov <timofluc@gmail.com> 1748514785 +0300

added index.html

```

Рисунок 81 - Использование алиасов

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git cat-file -p 0d97175
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391      a.html
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391      b.html
100644 blob d091aa28c2739117ea35a3440e582ff5a6c8edd9      index.html
040000 tree 9bd2cc72b85ee84bd7da4858ff5205e088258b6e      lib
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391      v.html

```

Рисунок 82 - Просмотр дерева каталогов

Затем нужно посмотреть каталог lib (рисунок 21).

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git cat-file -p 9bd2cc7
100644 blob 783c32015b8acb67bd767ca77d398f34272b0211      hello.html

```

Рисунок 83 - Просмотр каталога lib

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git cat-file -p 783c320
<!-- Author: Timofey Luchkov (timofluc@gmail.com)-->
<html>
  <head>
  </head>
  <body>
    <h2>second line</h2>
    <h1>Hello, world</h1>
  </body>
</html>
```

Рисунок 84 - Вывод содержимого файла hello.html

Аналогичным образом можно просмотреть содержимое файла, каким оно было в самом первом коммите, как показано на рисунке 23. Для этого требуется использовать лишь нужные хэши.

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist
* dc7f2db %a | added index.html (HEAD -> master) [Timofey Luchkov]
* 0e0be4b %a | moved hello.html to lib [Timofey Luchkov]
* 1d7024f %a | Add an author/email comment [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git cat-file -p 4258025
tree d8a800db068656c3aafc962f7a438c987b49dd60
author Timofey Luchkov <timofluc@gmail.com> 1748507592 +0300
committer Timofey Luchkov <timofluc@gmail.com> 1748507592 +0300

first commit

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git cat-file -p d8a800d
100644 blob dbe9dba55ea8fd4d5be3868b015e044be0848ec5    hello.html

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git cat-file -p dbe9dba
Hello, world
```

Рисунок 85 - Просмотр содержимого файла при первом коммите

УП_4

Работа с Git

Создание ветки

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

```
$ git checkout -b style
Switched to a new branch 'style'
```

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika/lib (style)
$ git status
On branch style
nothing to commit, working tree clean
```

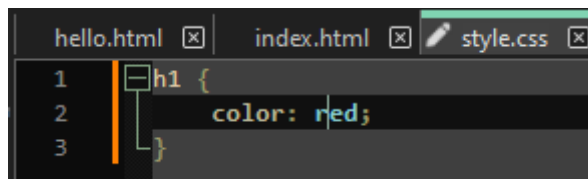
Рисунок 86 - Создание новой ветки style

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

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (style)
$ touch lib/style.css

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

Рисунок 87 - Создание файла стилей



```
1 h1 {
2   color: red;
3 }
```

Рисунок 88 - Код style.css

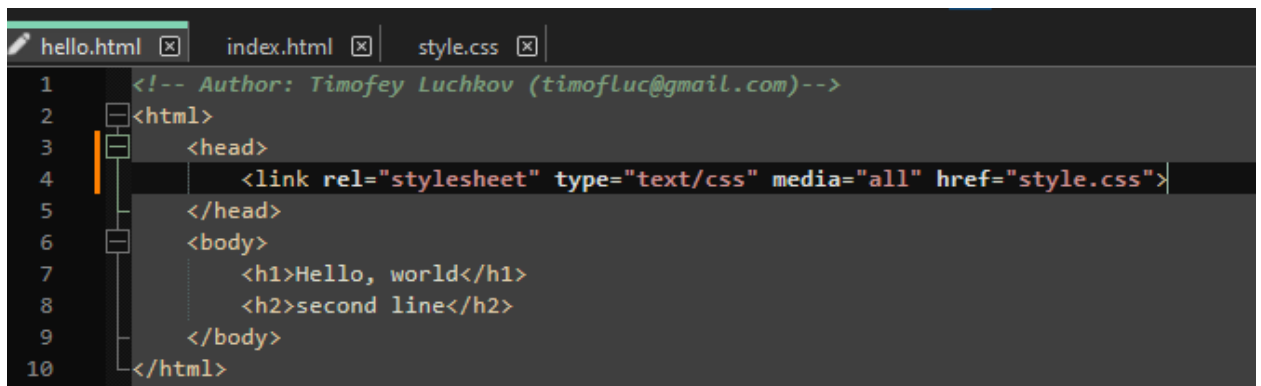
После этого надо произвести индексацию и коммит (рисунок 4).

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git add lib/style.css

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git commit -m "added css stylesheet"
[style d4b7bc8] added css stylesheet
1 file changed, 3 insertions(+)
create mode 100644 lib/style.css
```

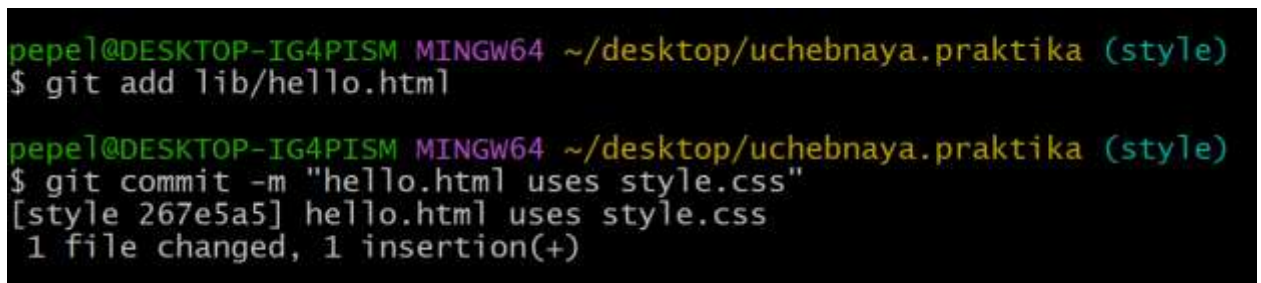
Рисунок 89 - Индексация и коммит нового файла

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



```
hello.html | index.html | style.css |
1  <!-- Author: Timofey Luchkov (timofluc@gmail.com)-->
2  <html>
3  <head>
4    <link rel="stylesheet" type="text/css" media="all" href="style.css">
5  </head>
6  <body>
7    <h1>Hello, world</h1>
8    <h2>second line</h2>
9  </body>
10 </html>
```

Рисунок 90 - Изменения в файле hello.html

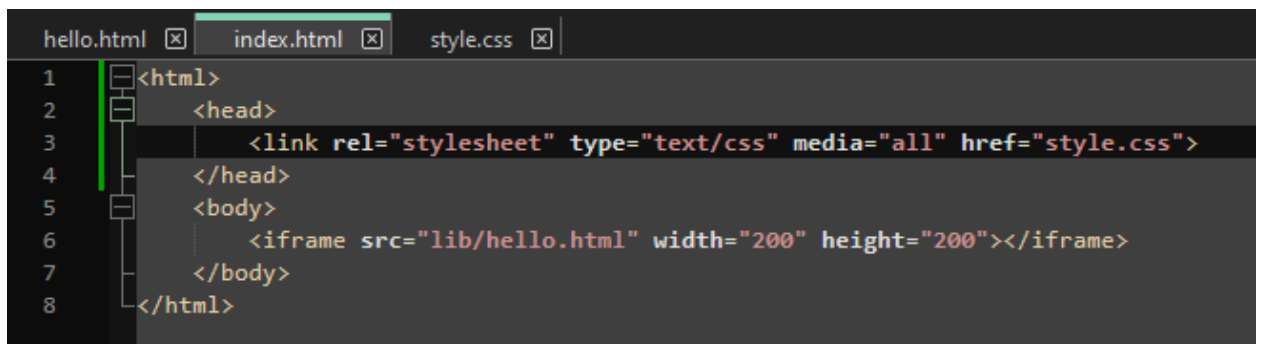


```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git add lib/hello.html

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git commit -m "hello.html uses style.css"
[style 267e5a5] hello.html uses style.css
1 file changed, 1 insertion(+)
```

Рисунок 91 - Индексация и коммит

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



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

Рисунок 92 - Изменения в файле index.html



```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git add index.html

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git commit -m "update index.html"
[style f4f077f] update index.html
1 file changed, 3 insertions(+)
```

Рисунок 93 - Индексация и коммит

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

Навигация по веткам

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git hist --all
* f4f077f %a | update index.html (HEAD -> style) [Timofey Luchkov]
* 267e5a5 %a | hello.html uses style.css [Timofey Luchkov]
* d4b7bc8 %a | added css stylesheet [Timofey Luchkov]
* 7715700 %a | idk [Timofey Luchkov]
* dc7f2db %a | added index.html (master) [Timofey Luchkov]
* 0e0be4b %a | moved hello.html to lib [Timofey Luchkov]
* 1d7024f %a | Add an author/email comment [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]
```

Рисунок 94 - Просмотр истории

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git checkout master
Switched to branch 'master'

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ cat lib/hello.html
<!-- Author: Timofey Luchkov (timofluc@gmail.com)-->
<html>
  <head>
  </head>
  <body>
    <h2>second line</h2>
    <h1>Hello, world</h1>
  </body>
</html>
```

Рисунок 95 - Переключение на ветку master

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


```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git checkout style
Switched to branch 'style'

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ cat lib/hello.html
<!-- Author: Timofey Luchkov (timofluc@gmail.com)-->
<html>
  <head>
    <link rel="stylesheet" type="text/css" media="all" href="style.css">
  </head>
  <body>
    <h1>Hello, world</h1>
    <h2>second line</h2>
  </body>
</html>

```

Рисунок 96 - Переключение на ветку style

Изменения в ветке master

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

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git checkout master
Switched to branch 'master'

```

Рисунок 97 - Переключение на ветку master

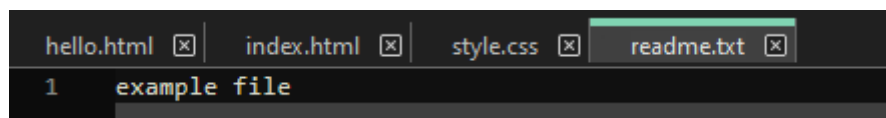


Рисунок 98 - Содержимое файла README

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git add README.txt

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git commit -m "added readme"
[master b918e68] added readme
1 file changed, 1 insertion(+)
create mode 100644 README.txt

```

Рисунок 99 - Индексация и коммит

Просмотр отличающихся веток

На рисунке 15 можно увидеть дерево коммитов.


```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist --all
* b918e68 %a | added readme (HEAD -> master) [Timofey Luchkov]
* f4f077f %a | update index.html (style) [Timofey Luchkov]
* 267e5a5 %a | hello.html uses style.css [Timofey Luchkov]
* d4b7bc8 %a | added css stylesheet [Timofey Luchkov]
* 7715700 %a | idk [Timofey Luchkov]
/
* dc7f2db %a | added index.html [Timofey Luchkov]
* 0e0be4b %a | moved hello.html to lib [Timofey Luchkov]
* 1d7024f %a | Add an author/email comment [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]

```

Рисунок 100 - Дерево коммитов

Слияние

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git checkout style
Switched to branch 'style'

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git merge master
Merge made by the 'ort' strategy.
 README.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.txt

```

Рисунок 101 - Переключение на ветку style и слияние с веткой master

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git hist --all
* 6b477c4 %a | Merge branch 'master' into style (HEAD -> style) [Timofey Luchkov]
\
* b918e68 %a | added readme (master) [Timofey Luchkov]
* f4f077f %a | update index.html [Timofey Luchkov]
* 267e5a5 %a | hello.html uses style.css [Timofey Luchkov]
* d4b7bc8 %a | added css stylesheet [Timofey Luchkov]
* 7715700 %a | idk [Timofey Luchkov]
/
* dc7f2db %a | added index.html [Timofey Luchkov]
* 0e0be4b %a | moved hello.html to lib [Timofey Luchkov]
* 1d7024f %a | Add an author/email comment [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]

```

Рисунок 17 - Просмотр истории

Создание конфликта

Для того, чтобы создать конфликт необходимо перейти в ветку master и внести изменения в файл hello.html (рисунки 18-20).

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git checkout master
Switched to branch 'master'

```

Рисунок 18 - Переход в ветку master

```

1  <!-- Author: Timofey Luchkov (timofluc@gmail.com)-->
2  <html>
3  <head>
4      <!-- no style -->
5  </head>
6  <body>
7      <h2>second line</h2>
8      <h1>Hello, world, life is beautiful</h1>
9  </body>
10 </html>

```

Рисунок 19 - Внесение изменений

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git add lib/hello.html

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git commit -m "live is beautiful"
[master 23e4a9d] live is beautiful
1 file changed, 2 insertions(+), 1 deletion(-)

```

Рисунок 20 - Индексация и коммит

После выполнения предыдущих действий при просмотре веток можно будет увидеть конфликт в виде, как на рисунке 21.

```

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist --all
* 23e4a9d %a | live is beautiful (HEAD -> master) [Timofey Luchkov]
* 6b477c4 %a | Merge branch 'master' into style (style) [Timofey Luchkov]
|
|>
|/
* b918e68 %a | added readme [Timofey Luchkov]
* f4f077f %a | update index.html [Timofey Luchkov]
* 267e5a5 %a | hello.html uses style.css [Timofey Luchkov]
* d4b7bc8 %a | added css stylesheet [Timofey Luchkov]
* 7715700 %a | idk [Timofey Luchkov]
|/
* dc7f2db %a | added index.html [Timofey Luchkov]
* 0e0be4b %a | moved hello.html to lib [Timofey Luchkov]
* 1d7024f %a | Add an author/email comment [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]

```

Рисунок 21 - Конфликт изменений

Разрешение конфликтов

При попытке объединить ветку style с master будет показана ошибка из-за конфликта, как показано на рисунке 22.

```

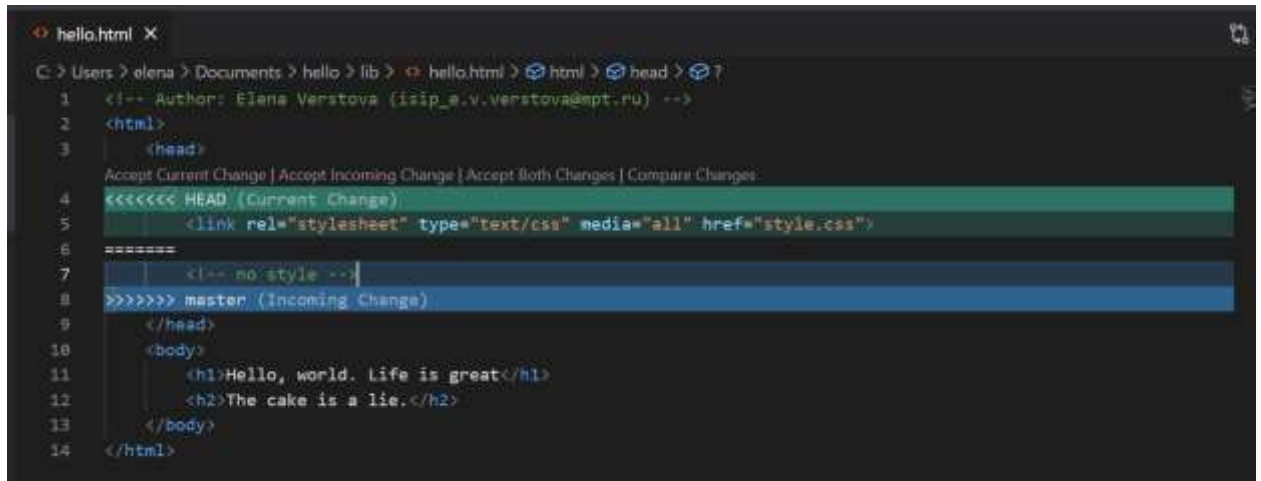
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git checkout style
Switched to branch 'style'

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git merge master
Auto-merging lib/hello.html
CONFLICT (content): Merge conflict in lib/hello.html
Automatic merge failed; fix conflicts and then commit the result.

```

Рисунок 22 - Ошибка при слиянии

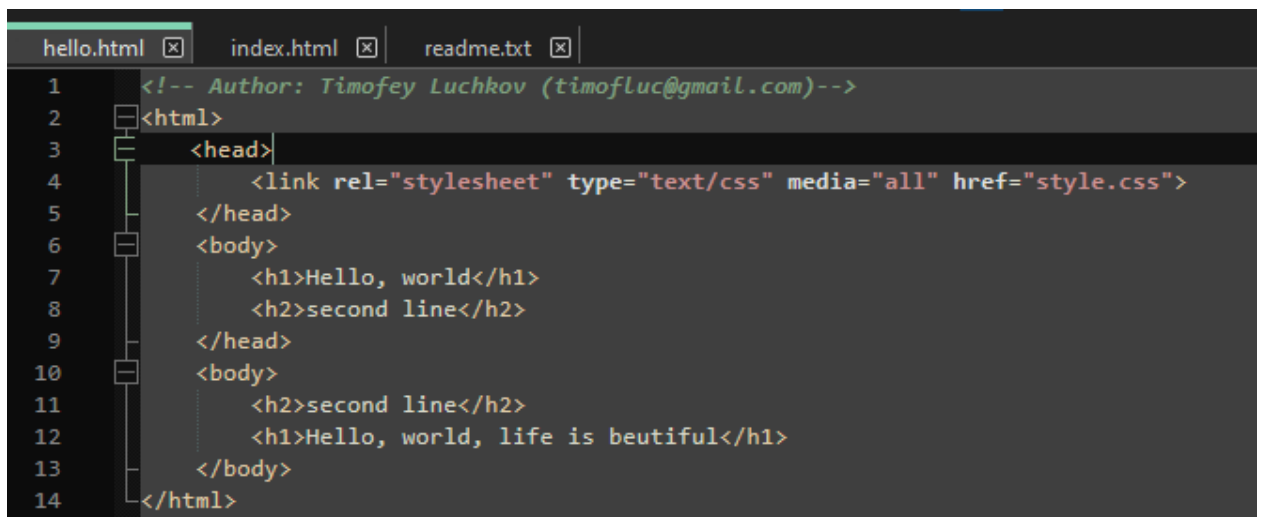
При открытии файла hello.html конфликт будет показан (рисунок 23).



```
hello.html X
C > Users > elena > Documents > hello > lib > hello.html > html > head > ?
1 <!-- Author: Elena Verstova (isip_e.v.verstova@yandex.ru) -->
2 <html>
3 <head>
4 <<<<<< HEAD (Current Change)
5 <link rel="stylesheet" type="text/css" media="all" href="style.css">
6 =====
7 <!-- no style -->
8 >>>>>> master (Incoming Change)
9 </head>
10 <body>
11 <h1>Hello, world. Life is great</h1>
12 <h2>The cake is a lie.</h2>
13 </body>
14 </html>
```

Рисунок 23 - Просмотр файла hello.html при наличии конфликта

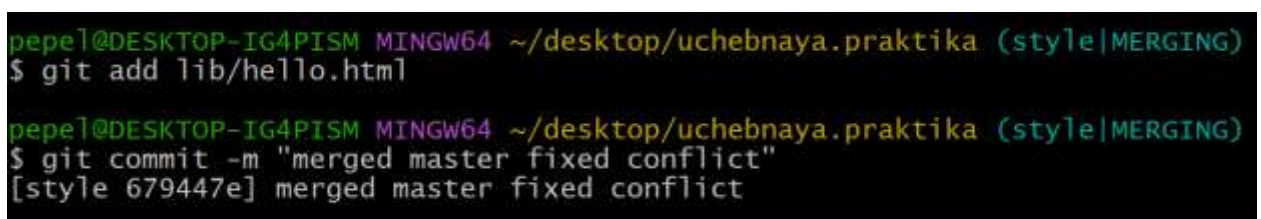
Чтобы решить конфликт, нужно внести изменения вручную (рисунок 24).



```
hello.html X index.html X readme.txt X
1 <!-- Author: Timofey Luchkov (timofluc@gmail.com) -->
2 <html>
3 <head>
4 <link rel="stylesheet" type="text/css" media="all" href="style.css">
5 </head>
6 <body>
7 <h1>Hello, world</h1>
8 <h2>second line</h2>
9 </head>
10 <body>
11 <h2>second line</h2>
12 <h1>Hello, world, life is beautiful</h1>
13 </body>
14 </html>
```

Рисунок 24 - Решение конфликта вручную

Затем следует произвести индексацию и коммит (рисунок 25).



```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style|MERGING)
$ git add lib/hello.html

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style|MERGING)
$ git commit -m "merged master fixed conflict"
[style 679447e] merged master fixed conflict
```

Рисунок 25 - Индексация и коммит

УП_5

Работа с Git

Сброс ветки style

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git checkout style
Already on 'style'

pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git hist
* 679447e %a | merged master fixed conflict (HEAD -> style) [Timofey Luchkov]
|
| * 23e4a9d %a | live is beautiful (master) [Timofey Luchkov]
| * 6b477c4 %a | Merge branch 'master' into style [Timofey Luchkov]
|
| * b918e68 %a | added readme [Timofey Luchkov]
| * f4f077f %a | update index.html [Timofey Luchkov]
| * 267e5a5 %a | hello.html uses style.css [Timofey Luchkov]
| * d4b7bc8 %a | added css stylesheet [Timofey Luchkov]
| * 7715700 %a | idk [Timofey Luchkov]
|
| * dc7f2db %a | added index.html [Timofey Luchkov]
| * 0e0be4b %a | moved hello.html to lib [Timofey Luchkov]
| * 1d7024f %a | Add an author/email comment [Timofey Luchkov]
| * 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
| * 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
| * 24f23e4 %a | a [Timofey Luchkov]
| * a4d1a8b %a | added html and body tag [Timofey Luchkov]
| * b7d8644 %a | added h2 tag [Timofey Luchkov]
| * 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
| * 7d8116e %a | unrelated change for v [Timofey Luchkov]
| * d02ca46 %a | changes for a and b [Timofey Luchkov]
| * 4258025 %a | first commit [Timofey Luchkov]
```

Рисунок 102 - Просмотр истории

```
pepel@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git hist --all
* 23e4a9d %a | live is beautiful (master) [Timofey Luchkov]
* b918e68 %a | added readme [Timofey Luchkov]
* dc7f2db %a | added index.html (HEAD -> style) [Timofey Luchkov]
* 0e0be4b %a | moved hello.html to lib [Timofey Luchkov]
* 1d7024f %a | Add an author/email comment [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]
```

Рисунок 103 - Сброс ветки style

Сброс ветки master

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git checkout master
Switched to branch 'master'

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist
* 23e4a9d %a | live is beutiful (HEAD -> master) [Timofey Luchkov]
* b918e68 %a | added readme [Timofey Luchkov]
* dc7f2db %a | added index.html (style) [Timofey Luchkov]
* 0e0be4b %a | moved hello.html to lib [Timofey Luchkov]
* 1d7024f %a | Add an author/email comment [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]
```

Рисунок 104 - Переключение на master и просмотр истории

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git reset --hard 23e4a9d
HEAD is now at 23e4a9d live is beutiful

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist --all
* 23e4a9d %a | live is beutiful (HEAD -> master) [Timofey Luchkov]
* b918e68 %a | added readme [Timofey Luchkov]
* dc7f2db %a | added index.html (style) [Timofey Luchkov]
* 0e0be4b %a | moved hello.html to lib [Timofey Luchkov]
* 1d7024f %a | Add an author/email comment [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]
```

Рисунок 105 - Сброс ветки master

Перебазирование

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

```

4258025 %a | first commit [Timofey Luchkov]
pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git checkout style
Switched to branch 'style'

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git rebase master
Successfully rebased and updated refs/heads/style.

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git hist
* 23e4a9d %a | live is beautiful (HEAD -> style, master) [Timofey Luchkov]
* b918e68 %a | added readme [Timofey Luchkov]
* dc7f2db %a | added index.html [Timofey Luchkov]
* 0e0be4b %a | moved hello.html to lib [Timofey Luchkov]
* 1d7024f %a | Add an author/email comment [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]

```

Рисунок 106 - Перебазирование веток

Слияние в ветку master

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

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (style)
$ git checkout master
Switched to branch 'master'

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git merge style
Already up to date.

```

Рисунок 107 - Слияние веток


```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ git hist
* 23e4a9d %a | live is beautiful (HEAD -> master, style) [Timofey Luchkov]
* b918e68 %a | added readme [Timofey Luchkov]
* dc7f2db %a | added index.html [Timofey Luchkov]
* 0e0be4b %a | moved hello.html to lib [Timofey Luchkov]
* 1d7024f %a | Add an author/email comment [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]

```

Рисунок 108 - Просмотр истории

Клонирование репозитория

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

```

pepe1@DESKTOP-IG4PISM MINGW64 ~/desktop/uchebnaya.praktika (master)
$ cd

pepe1@DESKTOP-IG4PISM MINGW64 ~
$ pwd
/c/Users/pepe1

pepe1@DESKTOP-IG4PISM MINGW64 ~
$ cd documents

```



```

pepe1@DESKTOP-IG4PISM MINGW64 ~/documents
$ ls
Battlefield 1/      'Escape from Tarkov Arena'/  Player/      TrailMakers/      'Мои видеозаписи'@
beamng.drive/      GroundBranch/               'Rockstar Games'/  Ubisoft/          'Моя музыка'@
'Call of Duty Black Ops Cold War'/  'My Games'/                 'Tvalle Games'/    'ml Games'/       'Настраиваемые шаблоны office'/
'Call of Duty Modern Warfare'/      MysummerCar/                Teardown/          clonedopraktika/  'Санитары Подземелий'/
Dash/               NAME/                       'The Crew 2'/      desktop.ini       'Мои рисунки'@
DayZ/               overwatch/                 TheCrewMotorfest/  dzsalauncher/
'DayZ Other Profiles'/  'Paradox Interactive'/      TheLongDrive/      steamvr/

```

Рисунок 109 - Переход в рабочий каталог и его клонирование

Просмотр клонированного репозитория

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents
$ cd clonedpraktika

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonedpraktika (master)
$ ls
README.txt  a.html  b.html  index.html  lib/  v.html

```

Рисунок 110 - Просмотр содержимого клонированного репозитория

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonedpraktika (master)
$ git hist --all
* 23e4a9d %a | live is beutifu] (HEAD -> master, origin/style, origin/master, origin/HEAD) [Timofey Luchkov]
* b918e68 %a | added readme [Timofey Luchkov]
* dc7f2db %a | added index.html [Timofey Luchkov]
* 0e0be4b %a | moved hello.html to lib [Timofey Luchkov]
* 1d7024f %a | Add an author/email comment [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]

```

Рисунок 111 - Просмотр логов клонированного каталога

Origin

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonedpraktika (master)
$ git remote
origin

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonedpraktika (master)
$ git remote show origin
* remote origin
  Fetch URL: C:/Users/pepel/Desktop/uchebnaya.praktika
  Push URL: C:/Users/pepel/Desktop/uchebnaya.praktika
  HEAD branch: master
  Remote branches:
    master tracked
    style tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (up to date)

```

Рисунок 112 - Просмотр данных об origin

Удаленные ветки

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonedpraktika (master)
$ git branch
* master

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonedpraktika (master)
$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style

```

Рисунок 113 - Просмотр удаленных веток

УП_6

Работа с Git

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonedpraktika (master)
$ cd ../uchebnaya.praktika

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/uchebnaya.praktika (master)
$ |

```

Рисунок 114 - Переход в оригинальный репозиторий

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

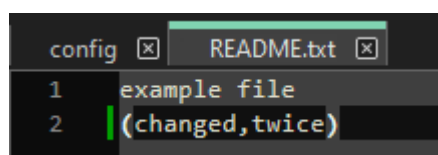


Рисунок 115 - Изменения в файле README

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/uchebnaya.praktika (master)
$ git add README.txt

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/uchebnaya.praktika (master)
$ git commit -m "changed readme in original"
[master ff7c280] changed readme in original
1 file changed, 2 insertions(+), 1 deletion(-)

```

Рисунок 116 - Индексация и коммит новых изменений

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/uchebnaya.praktika (master)
$ cd ../clonepraktika

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 296 bytes | 21.00 KiB/s, done.
From C:/Users/pepel/documents/uchebnaya.praktika
    ff7c280..6b96719  master    -> origin/master
```

Рисунок 117 - Извлечение изменений

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git hist --all
* 6b96719 %a | again changed readme (origin/master, origin/HEAD) [Timofey Luchkov]
* ff7c280 %a | changed readme in original (HEAD -> master) [Timofey Luchkov]
* 23e4a9d %a | live is beautiful (origin/style) [Timofey Luchkov]
* b918e68 %a | added readme [Timofey Luchkov]
* dc7f2db %a | added index.html [Timofey Luchkov]
* 0e0be4b %a | moved hello.html to lib [Timofey Luchkov]
* 1d7024f %a | Add an author/email comment [Timofey Luchkov]
* 88767cb %a | add html header (tag: ver1) [Timofey Luchkov]
* 574d9da %a | add standart html tags (tag: ver1-beta) [Timofey Luchkov]
* 24f23e4 %a | a [Timofey Luchkov]
* a4d1a8b %a | added html and body tag [Timofey Luchkov]
* b7d8644 %a | added h2 tag [Timofey Luchkov]
* 46ec5b7 %a | Added h2 tag [Timofey Luchkov]
* 7d8116e %a | unrelated change for v [Timofey Luchkov]
* d02ca46 %a | changes for a and b [Timofey Luchkov]
* 4258025 %a | first commit [Timofey Luchkov]
```

Рисунок 118 - Просмотр истории

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ cat README.txt
example file
(changed)
```

Рисунок 119 - Вывод содержимого файла README

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


```

(changed)
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git merge origin/master
Updating ff7c280..6b96719
Fast-forward
 README.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

```

Рисунок 120 - Слияние изменений

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ cat README.txt
example file
(changed, twice)

```

Рисунок 121 - Вывод содержимого файла README

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git pull
Already up to date.

```

Рисунок 122 - Команда git pull

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git branch --track style origin/style
branch 'style' set up to track 'origin/style'.

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git branch -a
* master
  style
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/style

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git hist --max-count=2
* 6b96719 %a | again changed readme (HEAD -> master, origin/master, origin/HEAD) [Timofey Luchkov]
* ff7c280 %a | changed readme in original [Timofey Luchkov]

```

Рисунок 123 - Добавление локальной ветки

Далее необходимо создать чистый репозиторий (рисунок 11).

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ cd ..

pepel@DESKTOP-IG4PISM MINGW64 ~/documents
$ git clone --bare uchebnaya.praktika uchebnaya.praktika.git
Cloning into bare repository 'uchebnaya.praktika.git'...
done.

pepel@DESKTOP-IG4PISM MINGW64 ~/documents
$ ls uchebnaya.praktika.git
HEAD  config  description  hooks/  info/  objects/  packed-refs  refs/

```

Рисунок 124 - Создание чистого репозитория

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents
$ cd uchebnaya.praktika

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/uchebnaya.praktika (master)
$ git remote add shared ../uchebnaya.praktika.git

```

Рисунок 125 - Добавление удаленного репозитория

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

```

≡ README ×
C: > Users > elena > Documents > hello > ≡ README
1   This is the Hello World example from the Git tutorial
2 | (Changed in the original and pushed to shared)

```

Рисунок 126 - Внесение изменений в файл

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/uchebnaya.praktika (master)
$ git checkout master
M      README.txt
Already on 'master'

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/uchebnaya.praktika (master)
$ git add README.txt

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/uchebnaya.praktika (master)
$ git commit -m "added shared comment to readme"
[master 5664b3f] added shared comment to readme
1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 127 - Индексация и коммит

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/uchebnaya.praktika (master)
$ git push shared 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), 347 bytes | 347.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
To ../uchebnaya.praktika.git
 6b96719..5664b3f  master -> master
```

Рисунок 128 - Отправка изменений в общий репозиторий

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/uchebnaya.praktika (master)
$ cd ../clonepraktika
```

Рисунок 129 - Переход в клонированный репозиторий


```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git remote add shared ../uchebnaya.praktika.git

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git branch --track shared master
branch 'shared' set up to track 'master'.

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git pull shared master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 327 bytes | 40.00 KiB/s, done.
From ../uchebnaya.praktika
 * branch            master      -> FETCH_HEAD
 * [new branch]      master      -> shared/master
Updating 6b96719..5664b3f
Fast-forward
 README.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ cat README.txt
example file
(changed in original and pushed to shared)

```

Рисунок 130 - 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).

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git daemon --verbose --export-all -- base-path=.
[2368] Ready to rumble

```

Рисунок 131 - Настройка сервера

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/uchebnaya.praktika (master)
$ git clone ../documents/uchebnaya.praktika.git network_uchebka
Cloning into 'network_uchebka'...
done.

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/uchebnaya.praktika (master)
$ cd network_uchebka

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/uchebnaya.praktika/network_uchebka (master)
$ ls
README.txt  a.html  b.html  index.html  lib/  v.html

```

Рисунок 132 - Клонирование проекта

УП_7

Работа с Git

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents
$ git clone https://github.com/schacon/ticgit
Cloning into 'ticgit'...
remote: Enumerating objects: 1857, done.
remote: Total 1857 (delta 0), reused 0 (delta 0), pack-reused 1857 (from 1)
Receiving objects: 100% (1857/1857), 334.06 KiB | 2.23 MiB/s, done.
Resolving deltas: 100% (837/837), done.

pepel@DESKTOP-IG4PISM MINGW64 ~/documents
$ cd ticgit

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/ticgit (master)
$ git remote
origin
```

Рисунок 133 – Клонирование репозитория и просмотр удаленных репозиторий

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)
```

Рисунок 134 - Просмотр удаленных репозиторий с ключом `-v`

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/ticgit (master)
$ git remote add pb https://github.com/paulboone/ticgit

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)
pb https://github.com/paulboone/ticgit (fetch)
pb https://github.com/paulboone/ticgit (push)
```

Рисунок 135 - Добавление удаленного репозитория

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/ticgit (master)
$ git fetch pb
remote: Enumerating objects: 43, done.
remote: Counting objects: 100% (22/22), done.
remote: Total 43 (delta 22), reused 22 (delta 22), pack-reused 21 (from 1)
Unpacking objects: 100% (43/43), 5.99 KiB | 42.00 KiB/s, done.
From https://github.com/paulboone/ticgit
* [new branch]      master    -> pb/master
* [new branch]      ticgit    -> pb/ticgit

```

Рисунок 136 - Использование имени вместо пути

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/ticgit (master)
$ git fetch ~/documents/clonepraktika
remote: Enumerating objects: 52, done.
remote: Counting objects: 100% (52/52), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 52 (delta 17), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (52/52), 4.39 KiB | 26.00 KiB/s, done.
From C:/Users/pepel/documents/clonepraktika
* branch            HEAD      -> FETCH_HEAD

```

Рисунок 137 - Получение данных из удаленных проектов

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

```

* [new branch]      master    -> pb/master
* [new branch]      ticgit    -> pb/ticgit

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/ticgit (master)
$ git fetch ~/documents/clonepraktika
remote: Enumerating objects: 52, done.
remote: Counting objects: 100% (52/52), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 52 (delta 17), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (52/52), 4.39 KiB | 26.00 KiB/s, done.
From C:/Users/pepel/documents/clonepraktika
* branch            HEAD      -> FETCH_HEAD

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/ticgit (master)
$ git push origin master

```




Рисунок 138 - Отправка изменений в удаленный репозиторий

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/documents/ticgit (master)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/schacon/ticgit
  Push URL: https://github.com/schacon/ticgit
  HEAD branch: master
  Remote branches:
    master tracked
    ticgit tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (up to date)
```

Рисунок 139 - Информация об удаленном репозитории

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/documents/ticgit (master)
$ git remote rename pb paul
Renaming remote references: 100% (4/4), done.

pepe1@DESKTOP-IG4PISM MINGW64 ~/documents/ticgit (master)
$ git remote
origin
paul
```

Рисунок 140 - Переименование удаленного репозитория

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/documents/ticgit (master)
$ git remote remove paul

pepe1@DESKTOP-IG4PISM MINGW64 ~/documents/ticgit (master)
$ git remote
origin
```

Рисунок 141 - Удаление удаленного репозитория

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


```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git tag
ver1
ver1-beta
```

Рисунок 142 - Просмотр тегов

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git tag -a v2.1 -m "my version 2.1"

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git tag
v2.1
ver1
ver1-beta
```

Рисунок 143 - Создание аннотированной метки

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git show v2.1
tag v2.1
Tagger: Timofey Luchkov <timofluc@gmail.com>
Date:   Fri Jun 6 12:17:58 2025 +0300

my version 2.1

commit 5664b3fcd2a726b300fd1216080ae2543eb42ab8 (HEAD -> master, tag: v2.1, shared/master)
Author: Timofey Luchkov <timofluc@gmail.com>
Date:   Fri May 30 13:10:22 2025 +0300

    added shared comment to readme

diff --git a/README.txt b/README.txt
index f5a1530..366a61a 100644
--- a/README.txt
+++ b/README.txt
@@ -1,2 +1,2 @@
 example file
-(changed, twice)
 \ No newline at end of file
+(changed in original and pushed to shared)
 \ No newline at end of file
```

Рисунок 144 - 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).

```
eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git tag v2.1-lw

eIena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello (master)
$ git tag
v2.1
v2.1-lw
ver1
ver1-beta
ver2

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

Рисунок 145 - Создание легковесной метки

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git show v2.1-lw
commit 5664b3fcd2a726b300fd1216080ae2543eb42ab8 (HEAD -> master, tag: v2.1-lw, tag: v2.1, shared/master)
Author: Timofey Luchkov <timofluc@gmail.com>
Date: Fri May 30 13:10:22 2025 +0300

    added shared comment to readme

diff --git a/README.txt b/README.txt
index f5a1530..366a61a 100644
--- a/README.txt
+++ b/README.txt
@@ -1,2 +1,2 @@
 example file
-(changed, twice)
 \ No newline at end of file
+(changed in original and pushed to shared)
 \ No newline at end of file
```

Рисунок 146 - Просмотр данных тега

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git log --pretty=oneline
5664b3fcd2a726b300fd1216080ae2543eb42ab8 (HEAD -> master, tag: v2.1-lw, tag: v2.1, shared/master) added shared comment to readme
6b96719d002de3c5e8e8fa5000bf91c9b8782b6be (origin/master, origin/HEAD, shared) again changed readme
ff7c280e411fde56ebf93bc81a7cc45c94f4e156 changed readme in original
23e4a9d5c071cedf9fecb95c1c37f8f75f034da2 (origin/style, style) live is beutiful
b918e68d181afa404cb08dc173d5b2128f7235d3 added readme
dc7f2dbe49f807ddeb42d55573ce8a60b9b69bc added index.html
0e0be4bf37eac0697d2b81d3cf4916346a0ab872 moved hello.html to lib
1d7024f8a673f45c34463cf98cc62ce77894b604 Add an author/email comment
88767cb3d6ae7a458e2d1ff2cea69eb3d1a9acbd (tag: ver1) add html header
374d9da150e2cac359407fc1168d79a1dbe664bc (tag: ver1-beta) add standart html tags
24f23e460536b5b45406b88c559d45bd5e936913 a
a4d1a8baf67d474abed7dce4c5b7ccc7e1ca5717 added html and body tag
b7d86448a45f4f81ce145965def12e1d7f814399 added h2 tag
46ec5b7828bbf2135e899f4ac671aec4e520eaa Added h2 tag
7d8116eed054fde1fd293f9bd4a89f2b4c282bb6 unrelated change for v
d02ca4653d8f83c4ecf3702bb96cee72d86076aa changes for a and b
425802565e18e45440c7968d1864bee47b30f41 First commit
```

Рисунок 147 - Просмотр истории

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git tag -a ver3 ff7c280
```

Рисунок 148 - Создание тега определенному коммиту

```
≡ TAG_EDITMSG ●
C: > Users > elena > Documents > hello > .git > ≡ TAG_EDITMSG
1 Tag message1
2 #
3 # Write a message for tag:
4 # ver3
5 # Lines starting with '#' will be ignored.
6
```

Рисунок 149 - Ввод сообщения в текстовом редакторе

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git show ver3
tag ver3
Tagger: Timofey Luchkov <timofluc@gmail.com>
Date: Fri Jun 6 12:21:22 2025 +0300

tag msg1

commit ff7c280e411fdc56ebf93bc81a7cc45c94f4e156 (tag: ver3)
Author: Timofey Luchkov <timofluc@gmail.com>
Date: Fri May 30 12:30:19 2025 +0300

    changed readme in original

diff --git a/README.txt b/README.txt
index 7897add..983d568 100644
--- a/README.txt
+++ b/README.txt
@@ -1,2 @@
-example file
\ No newline at end of file
+example file
+(changed)
\ No newline at end of file
```

Рисунок 150 - Просмотр данных тега

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


```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git push shared v2.1
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 169 bytes | 169.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To ../uchebnaya.praktika.git
 * [new tag]          v2.1 -> v2.1

```

Рисунок 151 - Отправка тега на удаленный сервер

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git push shared --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 164 bytes | 164.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To ../uchebnaya.praktika.git
 * [new tag]          v2.1-lw -> v2.1-lw
 * [new tag]          ver3 -> ver3

```

Рисунок 152 - Отправка всех тегов на сервер

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git tag -d v2.1-lw
Deleted tag 'v2.1-lw' (was 5664b3f)

```

Рисунок 153 - Удаление тега

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

```

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (master)
$ git push shared :refs/tags/2.1-lw
remote: warning: deleting a non-existent ref
To ../uchebnaya.praktika.git
 - [deleted]          2.1-lw

```

Рисунок 154 - 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).

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (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 5664b3f added shared comment to readme
```

Рисунок 155 - Переключение на метку

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika ((v2.1))
$ git checkout -b version2 v2.1
Switched to a new branch 'version2'

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (version2)
```

Рисунок 156 - Создание новой ветки

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika ((v2.1))
$ git checkout -b version2 v2.1
Switched to a new branch 'version2'

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (version2)
$ git config --global alias.co checkout

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (version2)
$ git config --global alias.br branch

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (version2)
$ git config --global alias.ci commit

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (version2)
$ git config --global alias.st status
```

Рисунок 157 - Задание алиасов

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (version2)
$ git config --global alias.unstage 'reset HEAD --'
```

Рисунок 158 - Создание псевдонима исключения файла из индекса

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (version2)
$ git unstage fileA
```

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (version2)
$ git reset HEAD -- fileA
```

Рисунок 159 - Использование созданного псевдонима

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (version2)
$ git config --global alias.last 'log -1 HEAD'
```

Рисунок 160 - Создание алиаса для просмотра последнего коммита

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (version2)
$ git last
commit 5664b3fcd2a726b300fd1216080ae2543eb42ab8 (HEAD -> version2, tag: v2.1, shared/master, master)
Author: Timofey Luchkov <timofluc@gmail.com>
Date: Fri May 30 13:10:22 2025 +0300

    added shared comment to readme
```

Рисунок 161 - Результат работы созданного алиаса

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/clonepraktika (version2)
$ git config --global alias.visual '!gitk'
```

Рисунок 162 - Создание псевдонима внешней команды

УП_8

Работа с Git

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git add README test.rb LICENSE

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git commit -m 'inita1 commit'
[master (root-commit) 41204cc] inita1 commit
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 LICENSE
create mode 100644 README
create mode 100644 test.rb
```

Рисунок 1 - Индексация и коммит 3 файлов

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git branch testing
```

Рисунок 2 - Создание ветки testing

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git checkout testing
Switched to branch 'testing'

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (testing)
```

Рисунок 3 - Переключение на ветку testing

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (testing)
$ git add test.rb

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (testing)
$ git commit -m 'new changes'
[testing db0de58] new changes
1 file changed, 1 insertion(+)
```

Рисунок 4 - Индексация и коммит файла test.rb

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (testing)
$ git checkout master
Switched to branch 'master'
```

Рисунок 5 - Переключение на ветку master


```
pepe1@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git add test.rb

pepe1@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git commit -a -m 'another commit'
[master c1c0e94] another commit
1 file changed, 1 insertion(+)
```

Рисунок 6 - Еще индексация и коммит test.rb

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git checkout -b iss53
Switched to a new branch 'iss53'
```

Рисунок 7 - Создание и переключение на ветку iss53

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

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (iss53)
$ git add index.html

pepe1@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (iss53)
$ git commit -m 'added html'
[iss53 591607f] added html
1 file changed, 1 insertion(+)
create mode 100644 index.html
```

Рисунок 8 - Индексация и коммит файла index.html

Далее нужно переключить ветку на master (рисунок 9).

```
pepe1@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (iss53)
$ git checkout master
Switched to branch '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).

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (hotfix)
$ git add index.html

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (hotfix)
$ git commit -m 'fixed broken email'
[hotfix 89834a1] fixed broken email
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.html
```

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (hotfix)
$ git checkout master
Switched to branch 'master'

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git merge hotfix
Updating c1c0e94..024614c
Fast-forward
 index.html | 1 +
1 file changed, 1 insertion(+)
create mode 100644 index.html
```

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

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git branch -d hotfix
Deleted branch hotfix (was 024614c).
```

Рисунок 12 - Удаление ветки hotfix

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (iss53)
$ git add index.html

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (iss53)
$ git commit -m 'finished new footer'
[iss53 caa5e10] finished new footer
1 file changed, 1 insertion(+), 1 deletion(-)
```

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


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

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

Рисунок 14 - Слияние веток

После этого ветку iss53 нужно удалить (рисунок 15).

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git branch -d iss53
Deleted branch iss53 (was caa5e10).
```

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).

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git branch
* master
  testing
```

Рисунок 163 - Список существующих веток

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git branch -v
* master 7c973b6 1
testing db0de58 new changes
```

Рисунок 164 - 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).

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git branch --merged
* master
```

Рисунок 165 - Список веток слитых с текущей

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git branch --no-merged
testing
```

Рисунок 166 - Список веток не слитых с текущей

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

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git branch -D testing
Deleted branch testing (was db0de58).
```

Рисунок 167 - Удаление ветки

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

```
elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two/repo1 (main)
$ git remote show origin
* remote origin
Fetch URL: https://github.com/Eternity-blip/repo1
Push URL: https://github.com/Eternity-blip/repo1
HEAD branch: main
Remote branch:
main tracked
Local branches configured for 'git pull':
main merges with remote main
serv merges with remote main
Local ref configured for 'git push':
main pushes to main (local out of date)
```

Рисунок 168 - Просмотр удаленных веток

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

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two/repo1 (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 285 bytes | 95.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Eternity-blip/repo1
6cdd680..f6b94bc main -> main
```

Рисунок 169 - Отправка изменений

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

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two/repo1 (main)
$ git fetch origin
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 693 bytes | 18.00 KiB/s, done.
From https://github.com/Eternity-blip/repo1
f6b94bc..68cbf5f main -> origin/main
```

Рисунок 170 - Выполнение команды `git fetch`

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

```
e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two/repo1 (main)
$ git checkout -b serv origin/main
Switched to a new branch 'serv'
Branch 'serv' set up to track remote branch 'main' from 'origin'.
```

Рисунок 171 - Создание ветки на основе удаленной ветки

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

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

Рисунок 172 - 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.

```
pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (master)
$ git checkout exp
Switched to branch 'exp'

pepel@DESKTOP-IG4PISM MINGW64 ~/documents/praktika_dva (exp)
$ git rebase master
Current branch exp is up to date.
```

Рисунок 173 - 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.

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

Рисунок 174 - Перемещение изменений с параметром `onto`

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

elena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git merge cl
Updating fafe213..08d256f
Fast-forward
 w.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 w.txt
```

Рисунок 175 - Слияние веток master и cl

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

Рисунок 176 - Перемещение изменений

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

e1ena@LAPTOP-I3I3QM5I MINGW64 ~/Documents/hello_two (master)
$ git merge se
Updating 08d256f..4a98db2
Fast-forward
 e.txt | 0
 q.txt | 0
 repo1 | 1 +
3 files changed, 1 insertion(+)
create mode 100644 e.txt
create mode 100644 q.txt
create mode 160000 repo1
```

Рисунок 177 - Слияние веток master и se

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

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

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

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

Рисунок 178 - Удаление веток cl и se