

Лабораторная работа №1

Работа с git

Махорин Иван Сергеевич

Содержание

1 Цель работы	7
2 Выполнение лабораторной работы	8
3 Выводы	72

Список иллюстраций

2.1	Параметры установки окончаний строк	8
2.2	Установка отображения unicode	9
2.3	Создание страницы «Hello, World»	10
2.4	Создание репозитория	11
2.5	Добавление файла в репозиторий	11
2.6	Проверка состояния репозитория	12
2.7	Изменение содержимого файла hello.html	12
2.8	Проверка состояния рабочего каталога	13
2.9	Индексация изменений	13
2.10	Добавление комментария, сохранение файла и выход из редактора	14
2.11	Повторная проверка состояния	14
2.12	Изменение страницы «Hello, World»	15
2.13	Добавление изменения в индекс git	15
2.14	Добавление заголовков HTML к странице «Hello, World»	16
2.15	Добавление стандартных тегов страницы	17
2.16	История	18
2.17	Получение старых версий	19
2.18	Создание тегов версий	20
2.19	Переключение по имени тега	21
2.20	Просмотр тегов с помощью команды tag	22
2.21	Переключитесь на ветку master	22
2.22	Изменение hello.html	23
2.23	Проверка состояния	23
2.24	Отмена изменений в рабочем каталоге	24
2.25	Внесение изменения в файл hello.html	24
2.26	Индексация изменения	25
2.27	Проверка состояния	25
2.28	Выполнение сброса буферной зоны	25
2.29	Переключение на версию коммита	26
2.30	Изменение файла hello.html	26
2.31	Коммит	27
2.32	Создание коммита с новыми изменениями, отменяющими предыдущие	27
2.33	Проверка лога	28
2.34	Проверка нашей истории	29
2.35	Отметка этой ветки для начала	30
2.36	Сброс коммитов к предшествующим коммиту Oops	30

2.37 Просмотр всех коммитов	31
2.38 Удаление тега oops	32
2.39 Добавление в страницу комментария автора	33
2.40 Коммит	33
2.41 Добавление email	34
2.42 Изменение предыдущего коммита	34
2.43 Просмотр истории	35
2.44 Перемещение файла hello.html в каталог lib	36
2.45 Коммит в новом каталоге	36
2.46 Добавление index.html	37
2.47 Каталог .git	37
2.48 Просмотр базы данных объектов	37
2.49 Углубление в базу данных объектов	38
2.50 Просмотр Config File	38
2.51 Изучение веток и тегов	39
2.52 Файл HEAD	39
2.53 Поиск последнего коммита	40
2.54 Вывод последнего коммита с помощью SHA1 хэша	40
2.55 Поиск дерева	41
2.56 Вывод каталога lib	41
2.57 Вывод файла hello.html	42
2.58 Создание ветки	42
2.59 Создание файла css	43
2.60 Добавление содержимого в файл	43
2.61 Коммит	44
2.62 Обновление файла hello.html	44
2.63 Коммит	45
2.64 Обновление файла index.html	45
2.65 Коммит	45
2.66 Навигация по веткам	46
2.67 Переключение на ветку master	47
2.68 Переключение на ветку style	47
2.69 Создание файла README.md	48
2.70 Коммит изменений README.md в ветку master	48
2.71 Просмотр текущих веток	49
2.72 Слияние веток	50
2.73 Возвращение в ветку master	51
2.74 Внесение изменений в файл	51
2.75 Коммит	52
2.76 Просмотр веток	53
2.77 Попытка объединения	54
2.78 Решение конфликта	54
2.79 Коммит решения конфликта	55
2.80 Возвращение на ветку style	56

2.81 Сброс ветки style к нужному коммиту	56
2.82 Проверка ветки	58
2.83 Сброс ветки master	59
2.84 Перебазирование	60
2.85 Слияние style в master	60
2.86 Просмотр логов	61
2.87 Клонирование репозиториев	62
2.88 Создание клона репозитория hello	62
2.89 Просмотр клонированного репозитория	63
2.90 Просмотр истории репозитория	63
2.91 Изучение origin	64
2.92 Просмотр доступных веток в клонированном репозитории	64
2.93 Просмотр всех веток	65
2.94 Переход в репозиторий hello	65
2.95 Внесение изменения в файл README.md	65
2.96 Добавление изменения и создание коммита	66
2.97 Извлечение изменений	66
2.98 Проверка README.md	67
2.99 Слив извлеченных изменений в локальную ветку master	67
2.100 Объединение fetch и merge в одну команду	67
2.101 Добавление локальной ветки, которая отслеживает удаленную ветку	68
2.102 Создание чистого репозитория	68
2.103 Добавление удаленного репозитория	69
2.104 Редактирование файла README.md	69
2.105 Коммит и отправка изменений в общий репозиторий	70
2.106 Извлечение общих изменений	71

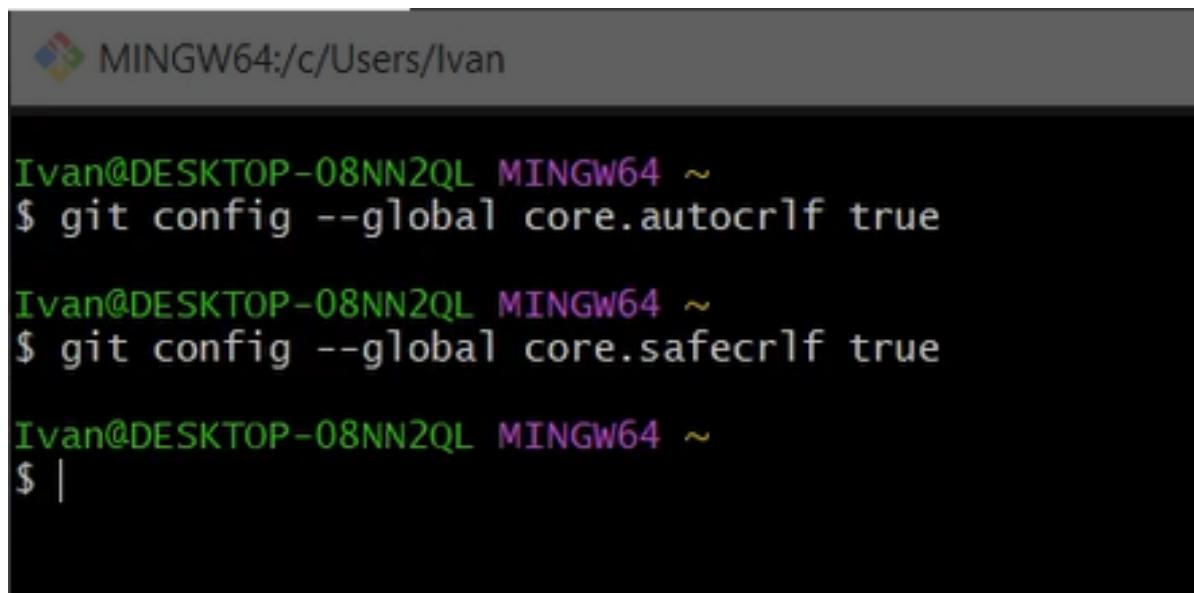
Список таблиц

1 Цель работы

Научиться полному взаимодействию с git и разобрать основные его особенности.

2 Выполнение лабораторной работы

Для начала установим параметры окончаний строк (рис. 2.1).



The screenshot shows a terminal window titled 'MINGW64:/c/Users/Ivan'. It contains three commands entered by the user:

```
Ivan@DESKTOP-08NN2QL MINGW64 ~
$ git config --global core.autocrlf true

Ivan@DESKTOP-08NN2QL MINGW64 ~
$ git config --global core.safecrlf true

Ivan@DESKTOP-08NN2QL MINGW64 ~
$ |
```

Рис. 2.1: Параметры установки окончаний строк

По умолчанию, git будет печатать не-ASCII символы в именах файлов в виде восьмеричных последовательностей. Что бы избежать нечитаемых строк, установим соответствующий флаг (рис. 2.2).



```
MINGW64:/c/Users/Ivan  
Ivan@DESKTOP-08NN2QL MINGW64 ~  
$ git config --global core.quotepath off  
Ivan@DESKTOP-08NN2QL MINGW64 ~  
$ |
```

Рис. 2.2: Установка отображения unicode

Начнём работу в пустом рабочем каталоге с создания пустого каталога с именем hello, затем войдём в него и создадим там файл с именем hello.html (рис. 2.3).

```
MINGW64:/c/Users/Ivan/MD/lab01/hello
Ivan@DESKTOP-08NN2QL MINGW64 ~
$ mkdir MD

Ivan@DESKTOP-08NN2QL MINGW64 ~
$ cd MD

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD
$ mkdir lab01

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ cd lab01

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ ls

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ mkdir hello

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ cd hello

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello
$ touch hello.html

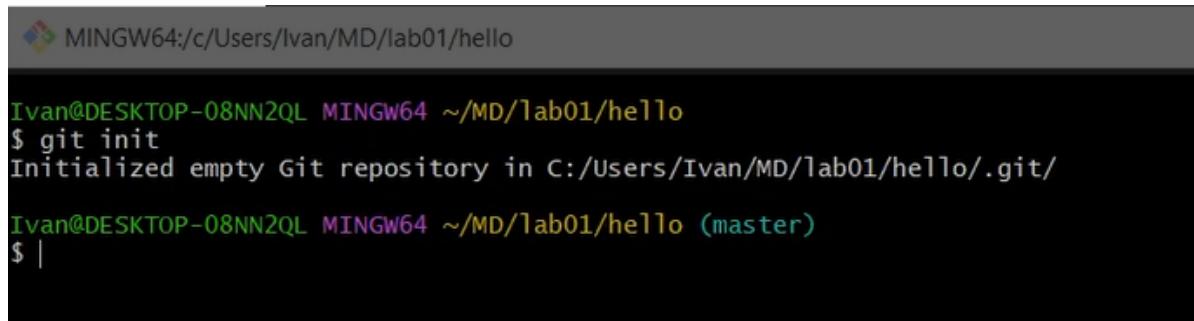
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello
$ echo "Hello, world!" > hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello
$ |
```

Рис. 2.3: Создание страницы «Hello, World»

Чтобы создать git репозиторий из этого каталога, выполним команду git init

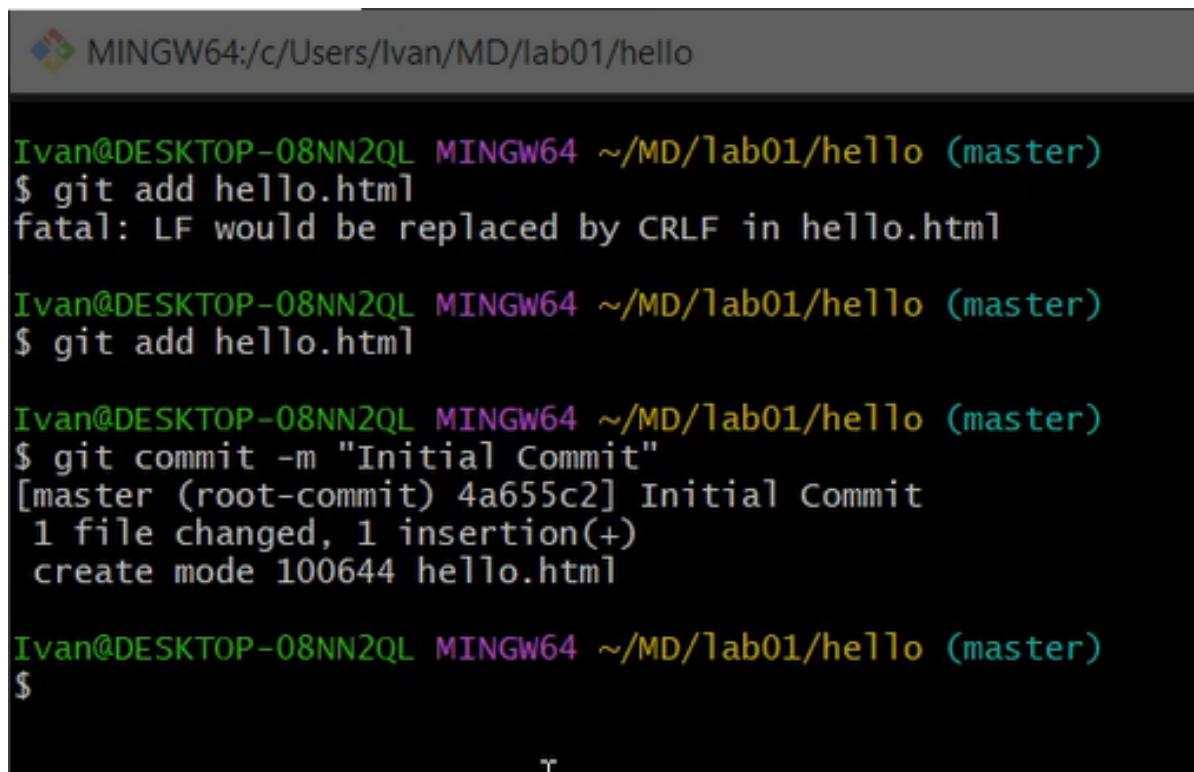
(рис. 2.4).



```
MINGW64:/c/Users/Ivan/MD/lab01/hello
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello
$ git init
Initialized empty Git repository in C:/Users/Ivan/MD/lab01/hello/.git/
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ |
```

Рис. 2.4: Создание репозитория

Добавим файл в репозиторий (рис. 2.5).



```
MINGW64:/c/Users/Ivan/MD/lab01/hello
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add hello.html
fatal: LF would be replaced by CRLF in hello.html

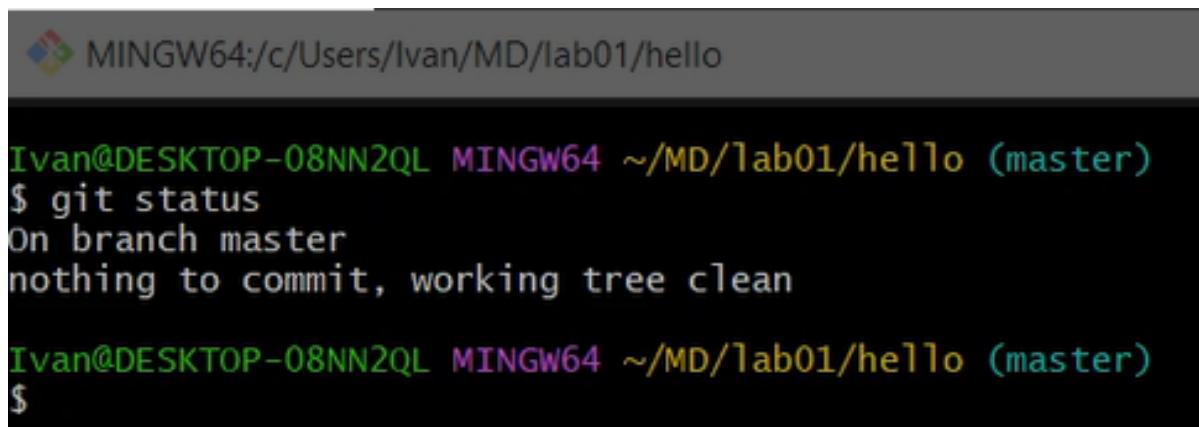
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit -m "Initial Commit"
[master (root-commit) 4a655c2] Initial Commit
 1 file changed, 1 insertion(+)
 create mode 100644 hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ |
```

Рис. 2.5: Добавление файла в репозиторий

Используем команду `git status`, чтобы проверить текущее состояние репозитория (рис. 2.6).



```
MINGW64:/c/Users/Ivan/MD/lab01/hello
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git status
On branch master
nothing to commit, working tree clean

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.6: Проверка состояния репозитория

Добавим кое-какие HTML-теги к нашему приветствию. Изменим содержимое файла hello.html (рис. 2.7).

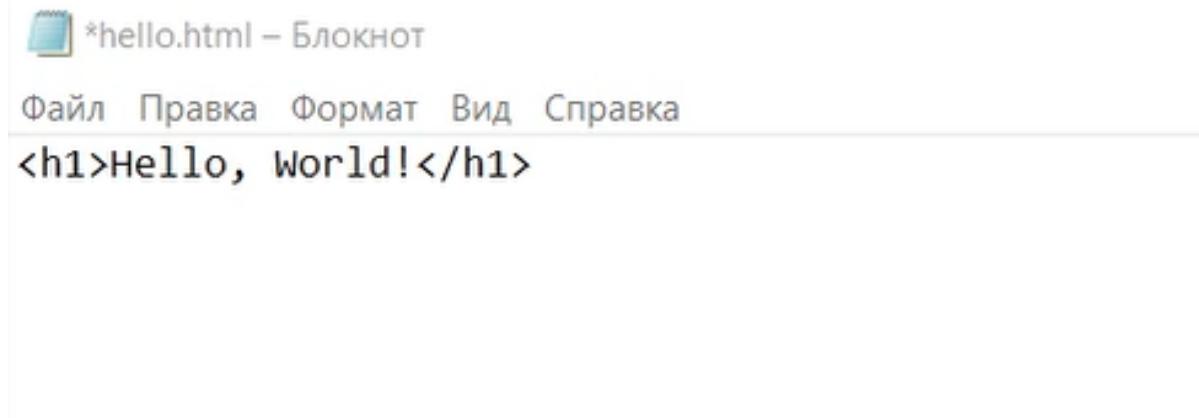


Рис. 2.7: Изменение содержимого файла hello.html

Проверим состояние рабочего каталога (рис. 2.8).

```
MINGW64:/c/Users/Ivan/MD/lab01/hello
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      modified:   hello.html

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

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ |
```

Рис. 2.8: Проверка состояния рабочего каталога

Теперь выполним команду git, чтобы проиндексировать изменения. Проверим состояние (рис. 2.9).

```
MINGW64:/c/Users/Ivan/MD/lab01/hello
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ |
```

Рис. 2.9: Индексация изменений

Сделаем коммит и проверим состояние. Откроется редактор. В первой строке введём комментарий: «Added h1 tag». Сохраним файл и выйдем из редактора (рис. 2.10).

```
MINGW64:/c/Users/Ivan/MD/lab01/hello
«Added h1 tag»
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#       modified:   hello.html
#
```

Рис. 2.10: Добавление комментария, сохранение файла и выход из редактора

Теперь еще раз проверим состояние (рис. 2.11).

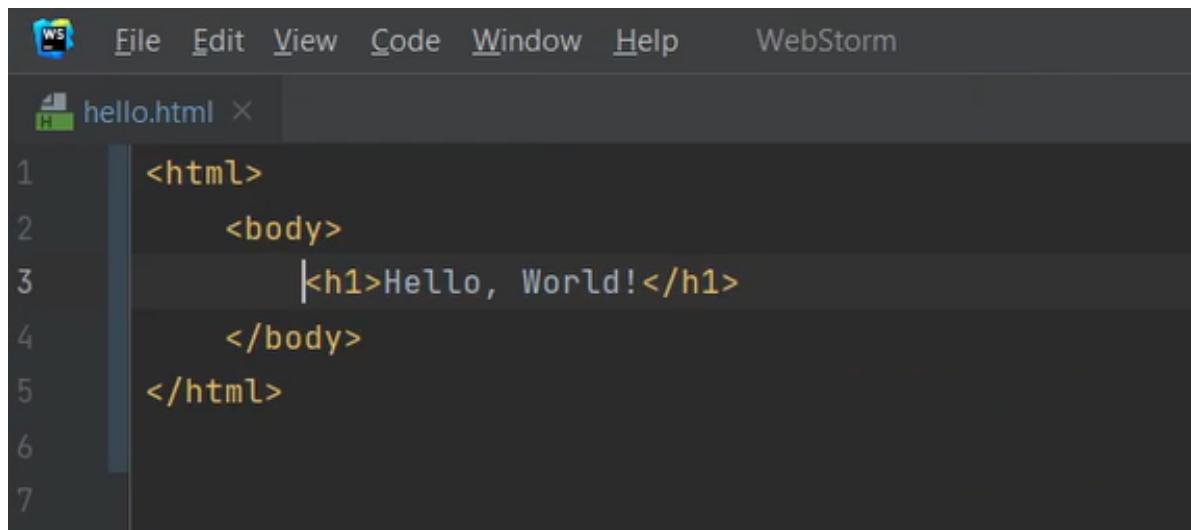
```
MINGW64:/c/Users/Ivan/MD/lab01/hello
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit
[master 6e27f33] «Added h1 tag»
  1 file changed, 1 insertion(+), 1 deletion(-)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git status
On branch master
nothing to commit, working tree clean

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ |
```

Рис. 2.11: Повторная проверка состояния

Изменим страницу «Hello, World», чтобы она содержала стандартные теги и (рис. 2.12).



```
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Рис. 2.12: Изменение страницы «Hello, World»

Теперь добавим это изменение в индекс git (рис. 2.13).

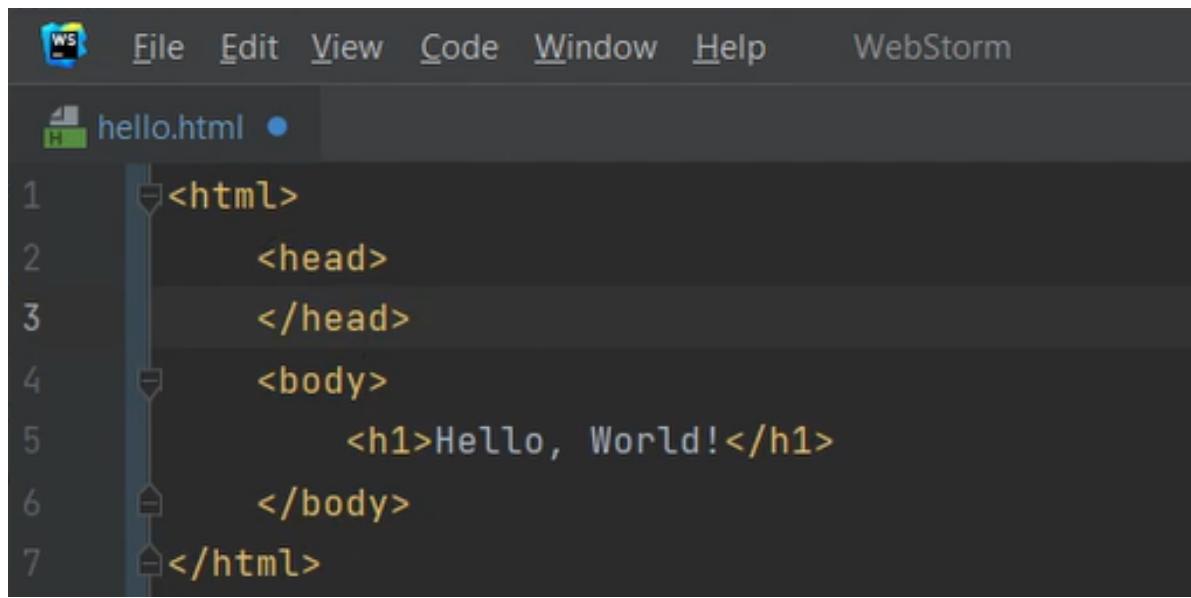


```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.13: Добавление изменения в индекс git

Теперь добавим заголовки HTML (секцию) к странице «Hello, World» (рис. 2.14).



```
1 <html>
2   <head>
3   </head>
4   <body>
5     <h1>Hello, World!</h1>
6   </body>
7 </html>
```

Рис. 2.14: Добавление заголовков HTML к странице «Hello, World»

Проверим текущий статус. Произведём коммит проиндексированного изменения (значение по умолчанию), а затем еще раз проверим состояние. Далее добавим второе изменение в индекс, а затем проверим состояние с помощью команды `git status` и сделаем коммит второго изменения (рис. 2.15).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (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

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit -m "Added standard HTML page tags"
[master 866efe5] Added standard HTML page tags
 1 file changed, 6 insertions(+), 1 deletion(-)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.html

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

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add .

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit -m "Added HTML header"
[master e9e7ae5] Added HTML header
 1 file changed, 2 insertions(+)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.15: Добавление стандартных тегов страницы

Получим список произведенных изменений различными способами (рис. 2.16).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (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

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit -m "Added standard HTML page tags"
[master 866efe5] Added standard HTML page tags
 1 file changed, 6 insertions(+), 1 deletion(-)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.html

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

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add .

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit -m "Added HTML header"
[master e9e7ae5] Added HTML header
 1 file changed, 2 insertions(+)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.16: История

Получим хэши предыдущих версий. Затем проверим содержимое файла hello.html и вернёмся к последней версии в ветке master (рис. 2.17).

```

Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:02:23 2024 +0300

    «Added h1 tag»

commit 4a655c202e8ab1fbbc0bca945f888b540bf10b20
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:00:11 2024 +0300

Initial Commit

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git checkout 4a655c2
Note: switching to '4a655c2'.

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 4a655c2 Initial Commit

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello ((4a655c2...))
$ cat hello.html
Hello, World!

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello ((4a655c2...))
$ git checkout master
Previous HEAD position was 4a655c2 Initial Commit
Switched to branch 'master'

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```

Рис. 2.17: Получение старых версий

Создадим тег первой версии. Далее переключимся на предыдущую версию.
Это версия с тегами

и

, но еще пока без

. Давайте сделаем ее версией v1-beta (рис. 2.18).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git checkout v1^
Note: switching to 'v1^'.

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 866efe5 Added standard HTML page tags

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello ((866efe5...))
$ cat hello.html
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello ((866efe5...))
$ git tag v1-beta

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello ((v1-beta))
$
```

Рис. 2.18: Создание тегов версий

Теперь попробуем попреключаться между двумя отмеченными версиями (рис. 2.19).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello ((v1-beta))
$ git checkout v1
Previous HEAD position was 866efe5 Added standard HTML page tags
HEAD is now at e9e7ae5 Added HTML header

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello ((v1))
$ git checkout v1-beta
Previous HEAD position was e9e7ae5 Added HTML header
HEAD is now at 866efe5 Added standard HTML page tags

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello ((v1-beta))
$ |
```

Рис. 2.19: Переключение по имени тега

Мы можем увидеть, какие теги доступны, используя команду `git tag` и также можем посмотреть теги в логе (рис. 2.20).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello ((v1-beta))
$ git tag
v1
v1-beta

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello ((v1-beta))
$ git log master --all
commit e9e7ae5e45ced4b85256116d4467ee7b0e86c4e3 (tag: v1, master)
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:06:02 2024 +0300

    Added HTML header

commit 866efe5a303de974263c42fdbbe695c6f349e0c1b (HEAD, tag: v1-beta)
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:05:27 2024 +0300

    Added standard HTML page tags

commit 6e27f33880a0bf50ff296626e508f84dafc31ee4
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:02:23 2024 +0300

    «Added h1 tag»

commit 4a655c202e8ab1fbcc0bca945f888b540bf10b20
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:00:11 2024 +0300

    Initial Commit

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello ((v1-beta))
$
```

Рис. 2.20: Просмотр тегов с помощью команды tag

Убедимся, что мы находимся на последнем коммите ветки master, прежде чем продолжить работу (рис. 2.21).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello ((v1-beta))
$ git checkout master
Previous HEAD position was 866efe5 Added standard HTML page tags
Switched to branch 'master'

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.21: Переключитесь на ветку master

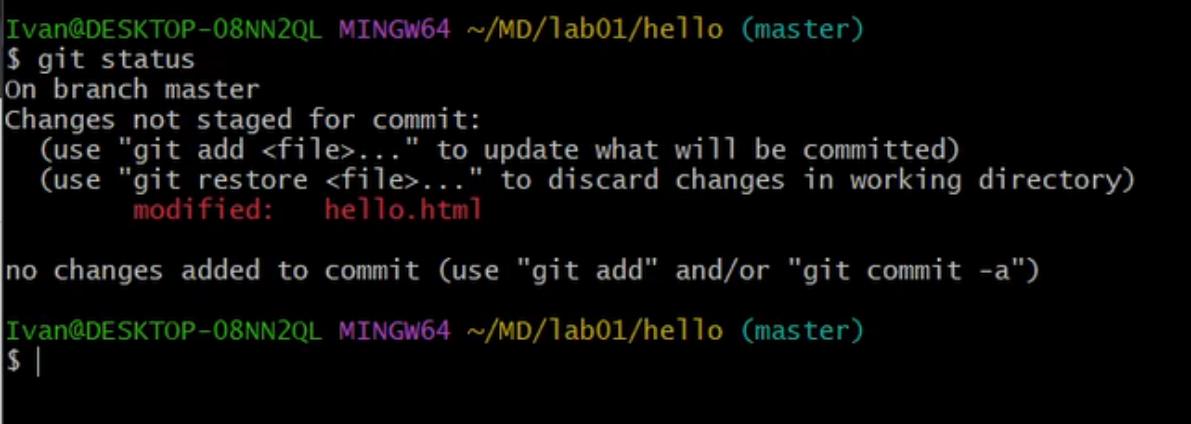
Внесём изменение в файл hello.html в виде нежелательного комментария (рис. 2.22).



```
hello.html
1 <html>
2   <head>
3     </head>
4     <body>
5       <h1>Hello, World!</h1>
6       |<!-- This is a bad comment. We want to revert it. --&gt;
7     &lt;/body&gt;
8 &lt;/html&gt;</pre>
```

Рис. 2.22: Изменение hello.html

Сначала проверим состояние рабочего каталога (рис. 2.23).



```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.html

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

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ |
```

Рис. 2.23: Проверка состояния

Используем команду git checkout для переключения версии файла hello.html в репозитории (рис. 2.24).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git checkout hello.html
Updated 1 path from the index

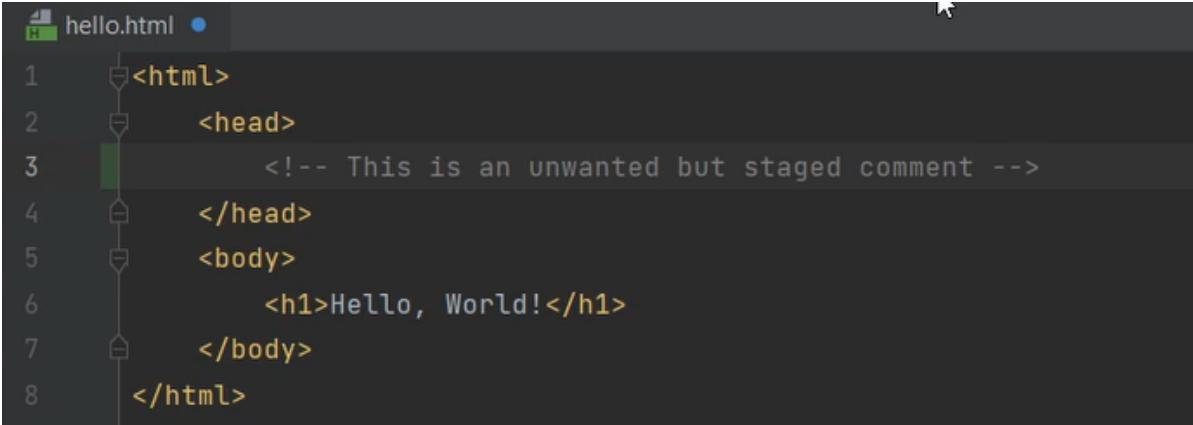
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git status
On branch master
nothing to commit, working tree clean

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ |
```

Рис. 2.24: Отмена изменений в рабочем каталоге

Внесём изменение в файл hello.html в виде нежелательного комментария (рис. 2.25).



The screenshot shows a code editor window with the file 'hello.html' open. The code is:

```
<html>
  <head>
    <!-- This is an unwanted but staged comment -->
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

The third line, which contains the comment, is highlighted with a green background, indicating it is staged for commit.

Рис. 2.25: Внесение изменения в файл hello.html

Проиндексируем это изменение (рис. 2.26).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.26: Индексация изменения

Проверим состояние нежелательного изменения (рис. 2.27).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ |
```

Рис. 2.27: Проверка состояния

К счастью, вывод состояния показывает нам именно то, что мы должны сделать для отмены индексации изменения (рис. 2.28).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git reset HEAD hello.html
Unstaged changes after reset:
M       hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ |
```

Рис. 2.28: Выполнениеброса буферной зоны

Переключимся на версию коммита (рис. 2.29).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git checkout hello.html
Updated 1 path from the index

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Рис. 2.29: Переключение на версию коммита

Изменим файл hello.html (рис. 2.30).

```
hello.html •
1 <html>
2   <head>
3     </head>
4   <body>
5     <h1>Hello, World!</h1>
6     <!-- This is an unwanted but committed change -->
7   </body>
8 </html>
```

Рис. 2.30: Изменение файла hello.html

Далее сделаем коммит (рис. 2.31).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit -m "Oops, we didn't want this commit"
[master 565fc35] Oops, we didn't want this commit
 1 file changed, 1 insertion(+)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.31: Коммит

Чтобы отменить коммит, нам необходимо сделать коммит, который удаляет изменения, сохраненные нежелательным коммитом (рис. 2.32).

```
Revert "Oops, we didn't want this commit"
This reverts commit 565fc350291180ff43455ee58a96e123d915130a.

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

Рис. 2.32: Создание коммита с новыми изменениями, отменяющими предыдущие

Проверка лога показывает нежелательные и отмененные коммиты в наш репозиторий (рис. 2.33).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git log
commit 2186257364056cd83d183bebe216f24c9c1713ea (HEAD -> master)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:15:05 2024 +0300

    Revert "Ooops, we didn't want this commit"
    This reverts commit 565fc350291180ff43455ee58a96e123d915130a.

commit 565fc350291180ff43455ee58a96e123d915130a
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:14:53 2024 +0300

    Oops, we didn't want this commit

commit e9e7ae5e45ced4b85256116d4467ee7b0e86c4e3 (tag: v1)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:06:02 2024 +0300

    Added HTML header

commit 866efe5a303de974263c42fbe695c6f349e0c1b (tag: v1-beta)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:05:27 2024 +0300

    Added standard HTML page tags

commit 6e27f33880a0bf50ff296626e508f84dafc31ee4
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:02:23 2024 +0300

    «Added h1 tag»

commit 4a655c202e8ab1fbcc0bca945f888b540bf10b20
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:00:11 2024 +0300

    Initial Commit

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
```

Рис. 2.33: Проверка лога

Сделаем быструю проверку нашей истории коммитов (рис. 2.34).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git log
commit 2186257364056cd83dbebe216f24c9c1713ea (HEAD -> master)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:15:05 2024 +0300

    Revert "Ooops, we didn't want this commit"
    This reverts commit 565fc350291180ff43455ee58a96e123d915130a.

commit 565fc350291180ff43455ee58a96e123d915130a
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:14:53 2024 +0300

    Oops, we didn't want this commit

commit e9e7ae5e45ced4b85256116d4467ee7b0e86c4e3 (tag: v1)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:06:02 2024 +0300

    Added HTML header

commit 866efe5a303de974263c42fdb695c6f349e0c1b (tag: v1-beta)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:05:27 2024 +0300

    Added standard HTML page tags

commit 6e27f33880a0bf50ff296626e508f84dafc31ee4
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:02:23 2024 +0300

    «Added h1 tag»

commit 4a655c202e8ab1fbcb0bca945f888b540bf10b20
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:00:11 2024 +0300

    Initial Commit

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
```

Рис. 2.34: Проверка нашей истории

Но прежде чем удалить коммиты, давайте отметим последний коммит тегом, чтобы потом можно было его найти (рис. 2.35).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ git tag oops

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ |
```

Рис. 2.35: Отметка этой ветки для начала

Глядя на историю лога, мы видим, что коммит с тегом «v1» является коммитом, предшествующим ошибочному коммиту. Давайте сбросим ветку до этой точки. Поскольку ветка имеет тег, мы можем использовать имя тега в команде сброса (если она не имеет тега, мы можем использовать хэш-значение) (рис. 2.36).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git reset --hard v1
HEAD is now at e9e7ae5 Added HTML header

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git log
commit e9e7ae5e45ced4b85256116d4467ee7b0e86c4e3 (HEAD -> master, tag: v1)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:06:02 2024 +0300

    Added HTML header

commit 866efe5a303de974263c42fdbe695c6f349e0c1b (tag: v1-beta)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:05:27 2024 +0300
```

Рис. 2.36: Сброс коммитов к предшествующим коммиту Oops

Что же случается с ошибочными коммитами? Оказывается, что коммиты все еще находятся в репозитории. На самом деле, мы все еще можем на них ссылаться. Помните, в начале этого урока мы создали для отмененного коммита тег «oops». Давайте посмотрим на все коммиты (рис. 2.37).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git log --all
commit 2186257364056cd83d183bebe216f24c9c1713ea (tag: oops)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:15:05 2024 +0300

    Revert "Ooops, we didn't want this commit"
    This reverts commit 565fc350291180ff43455ee58a96e123d915130a.

commit 565fc350291180ff43455ee58a96e123d915130a
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:14:53 2024 +0300

    Oops, we didn't want this commit

commit e9e7ae5e45ced4b85256116d4467ee7b0e86c4e3 (HEAD -> master, tag: v1)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:06:02 2024 +0300

    Added HTML header

commit 866efe5a303de974263c42fdbbe695c6f349e0c1b (tag: v1-beta)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:05:27 2024 +0300

    Added standard HTML page tags

commit 6e27f33880a0bf50ff296626e508f84dafc31ee4
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:02:23 2024 +0300

    «Added h1 tag»

commit 4a655c202e8ab1fbcc0bca945f888b540bf10b20
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:00:11 2024 +0300

    Initial Commit

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
```

Рис. 2.37: Просмотр всех коммитов

Тег `oops` свою функцию выполнил. Давайте удалим его и коммиты, на которые он ссылался, сборщиком мусора (рис. 2.38).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git tag -d oops
Deleted tag 'oops' (was 2186257)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git log --all
commit e9e7ae5e45ced4b85256116d4467ee7b0e86c4e3 (HEAD -> master, tag: v1)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:06:02 2024 +0300

    Added HTML header

commit 866efe5a303de974263c42fdbbe695c6f349e0c1b (tag: v1-beta)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:05:27 2024 +0300

    Added standard HTML page tags

commit 6e27f33880a0bf50ff296626e508f84dafc31ee4
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:02:23 2024 +0300

    «Added h1 tag»

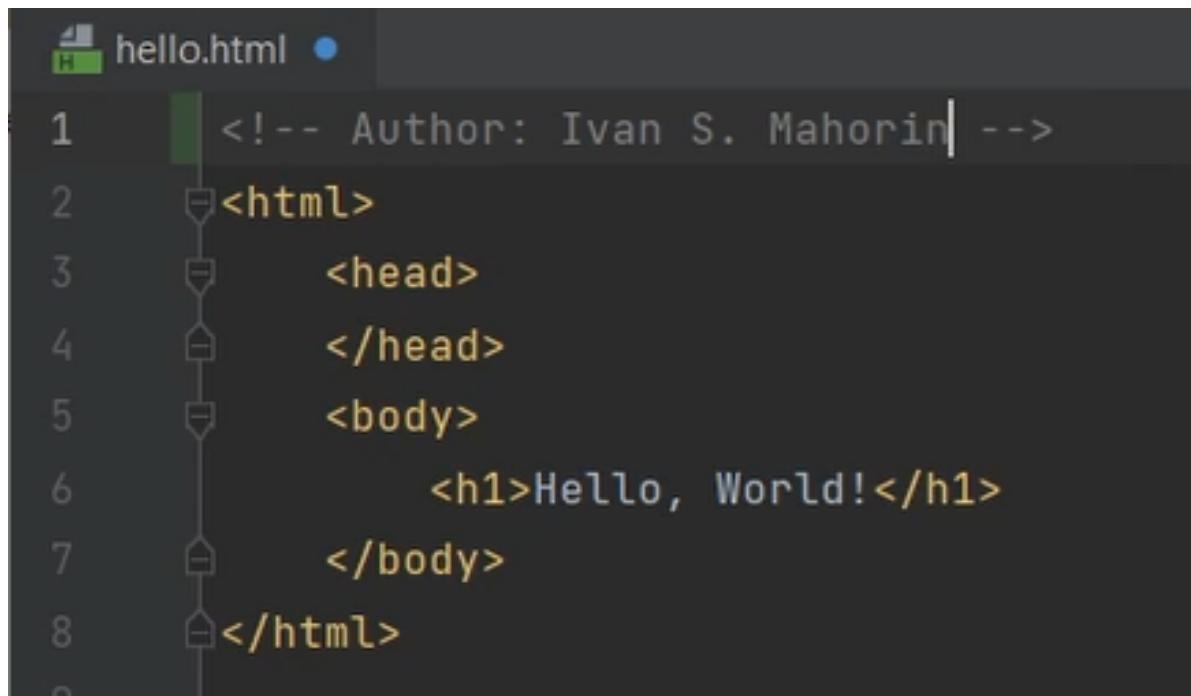
commit 4a655c202e8ab1fbcc0bca945f888b540bf10b20
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:00:11 2024 +0300

    Initial Commit

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.38: Удаление тега oops

Добавим в страницу комментарий автора (вставим свою фамилию) (рис. 2.39).



The screenshot shows a code editor window with the file 'hello.html' open. The content of the file is:

```
1 <!-- Author: Ivan S. Mahorin -->
2 <html>
3   <head>
4     </head>
5   <body>
6     <h1>Hello, World!</h1>
7   </body>
8 </html>
```

Рис. 2.39: Добавление в страницу комментария автора

После чего выполним коммит (рис. 2.40).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit -m "Add an author comment"
[master 184beed] Add an author comment
 1 file changed, 1 insertion(+)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.40: Коммит

После совершения коммита мы понимаем, что любой хороший комментарий должен включать электронную почту автора. Обновим страницу hello, включив в нее email (рис. 2.41).

```
1 <!-- Author: Ivan S. Mahorin (ivan.mahorin@yandex.ru) -->
2 <html>
3 <head>
4 </head>
5 <body>
6 <h1>Hello, World!</h1>
7 </body>
8 </html>
```

Рис. 2.41: Добавление email

Мы действительно не хотим создавать отдельный коммит только ради электронной почты. Давайте изменим предыдущий коммит, включив в него адрес электронной почты (рис. 2.42).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit --amend -m "Add an author/email comment"
[master ef99883] Add an author/email comment
  Date: Sat Feb 10 08:20:43 2024 +0300
  1 file changed, 1 insertion(+)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ |
```

Рис. 2.42: Изменение предыдущего коммита

Просмотрим историю (рис. 2.43).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git log
commit ef99883e2934a668eb2705b88302bc7dfe54b5dd (HEAD -> master)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:20:43 2024 +0300

    Add an author/email comment

commit e9e7ae5e45ced4b85256116d4467ee7b0e86c4e3 (tag: v1)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:06:02 2024 +0300

    Added HTML header

commit 866efe5a303de974263c42fbe695c6f349e0c1b (tag: v1-beta)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:05:27 2024 +0300

    Added standard HTML page tags

commit 6e27f33880a0bf50ff296626e508f84dafc31ee4
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:02:23 2024 +0300

    «Added h1 tag»

commit 4a655c202e8ab1fbcc0bca945f888b540bf10b20
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:00:11 2024 +0300

    Initial Commit

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.43: Просмотр истории

Сейчас мы собираемся создать структуру нашего репозитория. Давайте перенесем страницу в каталог lib (рис. 2.44).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ mkdir lib

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git mv hello.html lib

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:    hello.html -> lib/hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.44: Перемещение файла hello.html в каталог lib

Давайте сделаем коммит этого перемещения (рис. 2.45).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit -m "Moved hello.html to lib"
[master dfef41c] Moved hello.html to lib
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename hello.html => lib/hello.html (100%)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.45: Коммит в новом каталоге

Добавим файл index.html в наш репозиторий, добавим содержимое в файл и сделаем коммит (рис. 2.46).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ touch index.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add index.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit -m "Added index.html."
[master aa60c28] Added index.html.
 1 file changed, 5 insertions(+)
 create mode 100644 index.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.46: Добавление index.html

Просмотрим каталог, в котором хранится вся информация git (рис. 2.47).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ cd .git

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ ls
COMMIT_EDITMSG  ORIG_HEAD  description  index  logs/  packed-refs
HEAD          config     hooks/      info/   objects/ refs/
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$
```

Рис. 2.47: Каталог .git

Далее просмотрим набор каталогов, имена которых состоят из 2 символов. Имена каталогов являются первыми двумя буквами хэша sha1 объекта, хранящегося в git (рис. 2.48).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ ls objects
0a/  17/  1f/  3a/  52/  61/  6e/  86/  aa/  b8/  ca/  df/  ef/  f7/  pack/
0e/  18/  21/  4a/  56/  62/  7f/  8a/  b4/  b9/  d2/  e9/  f3/  info/
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$
```

Рис. 2.48: Просмотр базы данных объектов

Осмотрим один из каталогов с именем из 2 букв. Мы видим файлы с именами из 38 символов. Это файлы, содержащие объекты, хранящиеся в git. Они сжаты и закодированы, поэтому просмотр их содержимого нам мало чем поможет (рис. 2.49).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ ls objects/0a
cf493dc314e50480cd41ec306b7eb64e48a5bc
```

Рис. 2.49: Углубление в базу данных объектов

Изучим файл конфигурации, создающийся для каждого конкретного проекта. Записи в этом файле будут перезаписывать записи в файле .gitconfig нашего главного каталога, по крайней мере в рамках этого проекта (рис. 2.50).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ cat config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ |
```

Рис. 2.50: Просмотр Config File

Выполним набор команд для папки .git (рис. 2.51).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ ls refs
heads/ tags/

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ ls refs/heads
master

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ ls refs/tags
v1 v1-beta

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ cat refs/tags/v1
e9e7ae5e45ced4b85256116d4467ee7b0e86c4e3

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ |
```

Рис. 2.51: Изучение веток и тегов

Файл HEAD содержит ссылку на текущую ветку, в данный момент это должна быть ветка master (рис. 2.52).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ cat HEAD
ref: refs/heads/master

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$
```

Рис. 2.52: Файл HEAD

Эта команда показывает последний коммит в репозиторий (рис. 2.53).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ git log --max-count=1
commit aa60c28a292591cd94484ec7f92482ed0e68e31e (HEAD -> master)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:26:24 2024 +0300
```

```
    Added index.html.
```

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$
```

Рис. 2.53: Поиск последнего коммита

Выведем последний коммит с помощью SHA1 хэша (рис. 2.54).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ git cat-file -t aa60c28a
commit

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$ git cat-file -p aa60c28a
tree d258ffb922ff63ca6284dd57fd46683a1232728b
parent dfef41cf40f8b3b62cf6a239f85c81721a33008c
author Ivan-Mahorin <ivan.mahorin@yandex.ru> 1707542784 +0300
committer Ivan-Mahorin <ivan.mahorin@yandex.ru> 1707542784 +0300

Added index.html.
```

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
$
```

Рис. 2.54: Вывод последнего коммита с помощью SHA1 хэша

Мы можем вывести дерево каталогов, ссылка на который идет в коммите. Это должно быть описание файлов (верхнего уровня) в нашем проекте (для конкретного коммита). Используем SHA1 хеш из строки «дерева», из списка выше (рис. 2.55).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
```

```
$ git cat-file -p d258ffb922
```

```
100644 blob 8a3df81da2627a7b77bae3a87ae886120c9fb021 index.html
```

```
040000 tree b431c448abe9f899cd21c0598317b9502922a851 lib
```

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
```

```
$
```

Рис. 2.55: Поиск дерева

Выведем каталог lib (рис. 2.56).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
```

```
$ git cat-file -p b431c448abe
```

```
100644 blob 7f3d585f31fbf2f37cb7876e0b534fb4fe6060c hello.html
```

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello/.git (GIT_DIR!)
```

```
$
```

Рис. 2.56: Вывод каталога lib

Выведем файл hello.html (рис. 2.57).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ git cat-file -p 7f3d585f
<!-- Author: Ivan S. Mahorin (ivan.maho
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ |
```

Рис. 2.57: Вывод файла hello.html

Давайте назовем нашу новую ветку «style» (рис. 2.58).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git checkout -b style
Switched to a new branch 'style'

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git status
On branch style
nothing to commit, working tree clean

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$
```

Рис. 2.58: Создание ветки

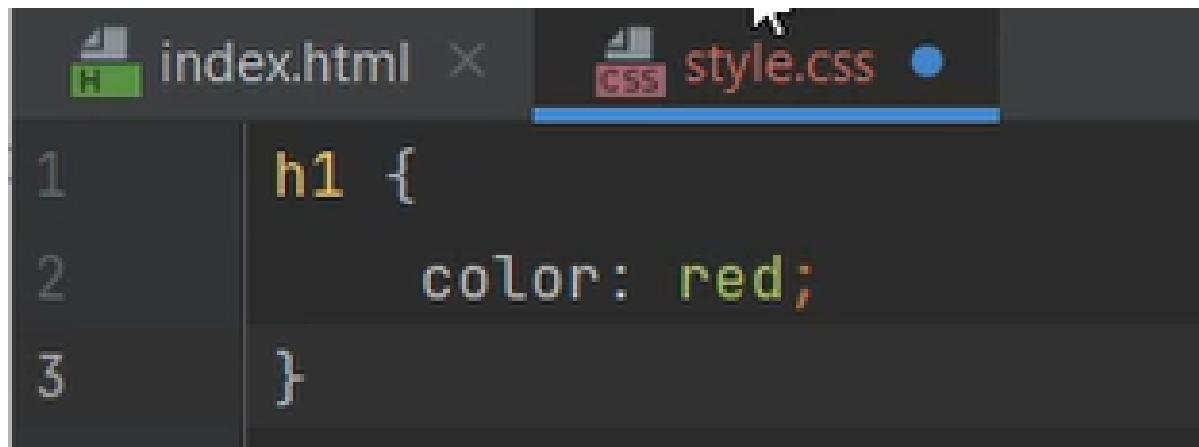
Создадим файл css (рис. 2.59).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ touch lib/style.css
```

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$
```

Рис. 2.59: Создание файла css

Добавим содержимое в файл (рис. 2.60).



```
index.html × style.css ●
1 h1 {
2     color: red;
3 }
```

Рис. 2.60: Добавление содержимого в файл

Выполним коммит (рис. 2.61).

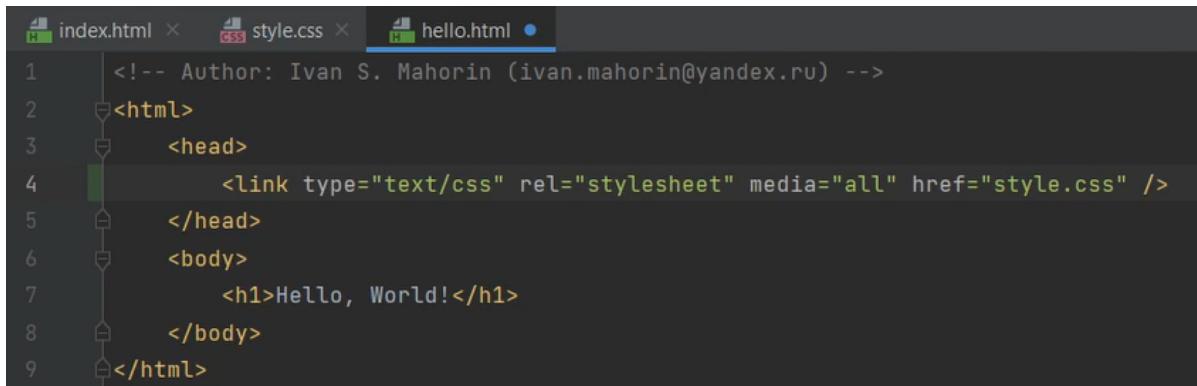
```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git add lib/style.css

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git commit -m "Added css stylesheet"
[style 91bc742] Added css stylesheet
 1 file changed, 3 insertions(+)
 create mode 100644 lib/style.css

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$
```

Рис. 2.61: Коммит

Обновим файл hello.html, чтобы использовать стили style.css (рис. 2.62).



```
index.html × style.css × hello.html •
1 <!-- Author: Ivan S. Mahorin (ivan.mahorin@yandex.ru) -->
2 <html>
3   <head>
4     <link type="text/css" rel="stylesheet" media="all" href="style.css" />
5   </head>
6   <body>
7     <h1>Hello, World!</h1>
8   </body>
9 </html>
```

Рис. 2.62: Обновление файла hello.html

Выполним коммит (рис. 2.63).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git add lib/hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git commit -m "Hello uses style.css"
[style 9fd3d10] Hello uses style.css
 1 file changed, 1 insertion(+)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$
```

Рис. 2.63: Коммит

Обновим файл index.html, чтобы он тоже использовал style.css (рис. 2.64).



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

Рис. 2.64: Обновление файла index.html

Выполним коммит (рис. 2.65).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git add index.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git commit -m "Updated index.html"
[style 429b594] Updated index.html
 1 file changed, 3 insertions(+)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$
```

Рис. 2.65: Коммит

Теперь в нашем проекте есть две ветки (рис. 2.66).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git log --all
commit 429b59467712a57b6e8a487b5a3a645aafe429d5 (HEAD -> style)
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:36:36 2024 +0300
        Updated index.html
I

commit 9fd3d1044e344c7b66b54e1c96443d5b52ba5e39
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:35:34 2024 +0300
        Hello uses style.css

commit 91bc742955e40067933109a44dc9f1eac193523a
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:34:17 2024 +0300
        Added css stylesheet

commit aa60c28a292591cd94484ec7f92482ed0e68e31e (master)
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:26:24 2024 +0300
        Added index.html.

commit dfef41cf40f8b3b62cf6a239f85c81721a33008c
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:23:04 2024 +0300
        Moved hello.html to lib

commit ef99883e2934a668eb2705b88302bc7dfe54b5dd
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:20:43 2024 +0300
        Add an author/email comment

commit e9e7ae5e45ced4b85256116d4467ee7b0e86c4e3 (tag: v1)
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:06:02 2024 +0300
```

Рис. 2.66: Навигация по веткам

Используем команду git checkout для переключения между ветками (рис. 2.67).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git checkout master
Switched to branch 'master'

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ cat lib/hello.html
<!-- Author: Ivan S. Mahorin (ivan.mahorin@yandex.ru) -->
<html>
    <head>
    </head>
    <body>
        <h1>Hello, world!</h1>
    </body>
</html>

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.67: Переключение на ветку master

Вернемся к ветке style (рис. 2.68).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git checkout style
Switched to branch 'style'

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ cat lib/hello.html
<!-- Author: Ivan S. Mahorin (ivan.mahorin@yandex.ru) -->
<html>
    <head>
        <link type="text/css" rel="stylesheet" media="all" href="style.css" />
    </head>
    <body>
        <h1>Hello, world!</h1>
    </body>
</html>

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$
```

Рис. 2.68: Переключение на ветку style

Переключимся на ветку master и создадим файл README.md (рис. 2.69).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git checkout master
Switched to branch 'master'

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ echo "This is the Hello World example from the git tutorial." > README.md

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.69: Создание файла README.md

Сделаем коммит изменений README.md в ветку master (рис. 2.70).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add README.md

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit -m "Added README"
[master 209871a] Added README
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.70: Коммит изменений README.md в ветку master

Просмотрим текущие ветки (рис. 2.71).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git log --graph --all
* commit 209871a5ef3ed5ecc108f2361a92bc4f80dba2df (HEAD -> master)
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:38:38 2024 +0300
|
|     Added README
|
* commit 429b59467712a57b6e8a487b5a3a645aafe429d5 (style)
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:36:36 2024 +0300
|
|     Updated index.html
|
* commit 9fd3d1044e344c7b66b54e1c96443d5b52ba5e39
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:35:34 2024 +0300
|
|     Hello uses style.css
|
* commit 91bc742955e40067933109a44dc9f1eac193523a
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:34:17 2024 +0300
|
|     Added css stylesheet
|
* commit aa60c28a292591cd94484ec7f92482ed0e68e31e
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:26:24 2024 +0300
|
|     Added index.html.
|
* commit dfef41cf40f8b3b62cf6a239f85c81721a33008c
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:23:04 2024 +0300
|
|     Moved hello.html to lib
|
* commit ef99883e2934a668eb2705b88302bc7dfe54b5dd
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
```

Рис. 2.71: Просмотр текущих веток

Слияние переносит изменения из двух веток в одну. Давайте вернемся к ветке style и сольем master c style (рис. 2.72).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git merge master
Merge made by the 'ort' strategy.
 README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git log --graph --all
* commit e311b7e3cfb3a0e5f51b34ff148ab95be64c896b (HEAD -> style)
| Merge: 429b594 209871a
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:39:16 2024 +0300
|
|       Merge branch 'master' into style
|
* commit 209871a5ef3ed5ecc108f2361a92bc4f80dba2df (master)
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:38:38 2024 +0300
|
|       Added README
|
* commit 429b59467712a57b6e8a487b5a3a645aafe429d5
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:36:36 2024 +0300
|
|       Updated index.html
|
* commit 9fd3d1044e344c7b66b54e1c96443d5b52ba5e39
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:35:34 2024 +0300
|
|       Hello uses style.css
|
* commit 91bc742955e40067933109a44dc9f1eac193523a
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:34:17 2024 +0300
|
|       Added css stylesheet
|
* commit aa60c28a292591cd94484ec7f92482ed0e68e31e
```

Рис. 2.72: Слияние веток

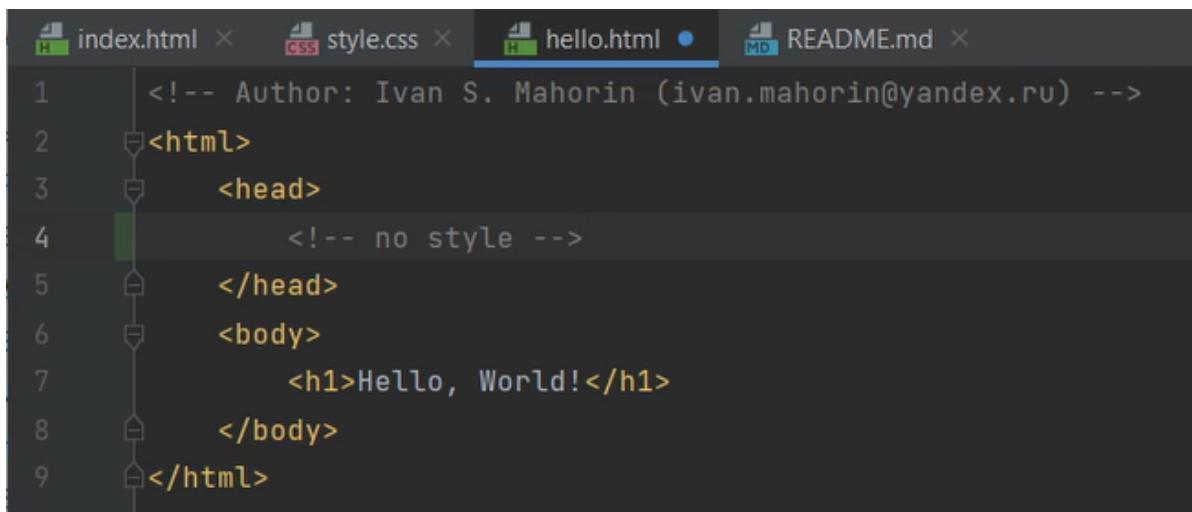
Вернёмся в ветку master (рис. 2.73).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git checkout master
Switched to branch 'master'

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.73: Возвращение в ветку master

Внесём следующие изменения в файл (рис. 2.74).



```
index.html × style.css × hello.html ● README.md ×
1 <!-- Author: Ivan S. Mahorin (ivan.mahorin@yandex.ru) -->
2 <html>
3   <head>
4     <!-- no style -->
5   </head>
6   <body>
7     <h1>Hello, World!</h1>
8   </body>
9 </html>
```

Рис. 2.74: Внесение изменений в файл

Выполним коммит (рис. 2.75).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git checkout master
Switched to branch 'master'

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add lib/hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit -m 'Life is great'
[master 01aab6] Life is great
 1 file changed, 1 insertion(+)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.75: Коммит

Просмотрим ветки (рис. 2.76).

```
* commit 01aabd6128e50c2bbebbe9f4877e8ect97bb6350 (HEAD -> master)
  Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
  Date:   Sat Feb 10 08:40:59 2024 +0300

    Life is great

* commit e311b7e3cfb3a0e5f51b34ff148ab95be64c896b (style)
  Merge: 429b594 209871a
  Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
  Date:   Sat Feb 10 08:39:16 2024 +0300

    Merge branch 'master' into style

* commit 209871a5ef3ed5ecc108f2361a92bc4f80dba2df
  Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
  Date:   Sat Feb 10 08:38:38 2024 +0300

    Added README

* commit 429b59467712a57b6e8a487b5a3a645aafe429d5
  Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
  Date:   Sat Feb 10 08:36:36 2024 +0300

    Updated index.html

* commit 9fd3d1044e344c7b66b54e1c96443d5b52ba5e39
  Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
  Date:   Sat Feb 10 08:35:34 2024 +0300

    Hello uses style.css

* commit 91bc742955e40067933109a44dc9f1eac193523a
  Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
  Date:   Sat Feb 10 08:34:17 2024 +0300

    Added css stylesheet

* commit aa60c28a292591cd94484ec7f92482ed0e68e31e
  Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
  Date:   Sat Feb 10 08:26:24 2024 +0300
```

Рис. 2.76: Просмотр веток

Теперь вернемся к ветке `style` и попытаемся объединить ее с новой веткой `master` (рис. 2.77).

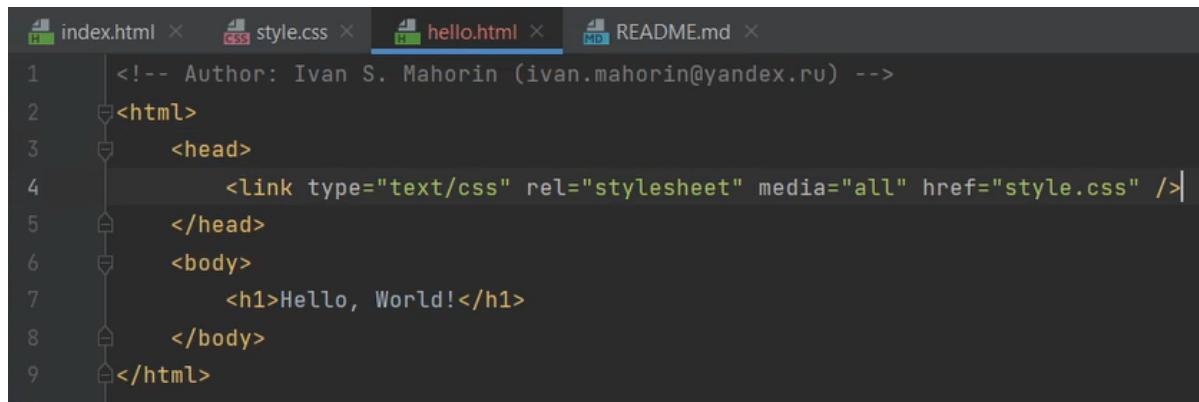
```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git checkout style
Switched to branch 'style'

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git merge master
Auto-merging lib/hello.html
CONFLICT (content): Merge conflict in lib/hello.html
Automatic merge failed; fix conflicts and then commit the result.

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style|MERGING)
$ |
```

Рис. 2.77: Попытка объединения

Нам необходимо вручную разрешить конфликт. Внесём изменения в lib/hello.html для достижения следующего результата (рис. 2.78).



The screenshot shows a code editor with four tabs at the top: index.html, style.css, hello.html, and README.md. The hello.html tab is active. The code in the editor is:

```
<!-- Author: Ivan S. Mahorin (ivan.mahorin@yandex.ru) -->
<html>
  <head>
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Рис. 2.78: Решение конфликта

Сделаем коммит решения конфликта (рис. 2.79).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style|MERGING)
$ git add lib/hello.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style|MERGING)
$ git commit -m "Merged master fixed conflict."
[style cd51a6c] Merged master fixed conflict.

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ |
```

Рис. 2.79: Коммит решения конфликта

Вернемся на ветку style к точке перед тем, как мы слили ее с веткой master. Мы можембросить ветку к любому коммиту. По сути, это изменение указателя ветки на любую точку дерева коммитов. В этом случае мы хотим вернуться в ветке style в точку перед слиянием с master. Нам необходимо найти последний коммит перед слиянием (рис. 2.80).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git checkout style
Already on 'style'

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git log --graph
* commit cd51a6c3836e94ff604259bc45a78755e79fba49 (HEAD -> style)
|\ Merge: e311b7e 01aabd6
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:43:29 2024 +0300
|
|     Merged master fixed conflict.

* commit 01aabbd6128e50c2bbebbe9f4877e8ecf97bb6350 (master)
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:40:59 2024 +0300
|
|     Life is great

* commit e311b7e3cfb3a0e5f51b34ff148ab95be64c896b
|\ Merge: 429b594 209871a
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:39:16 2024 +0300
|
|     Merge branch 'master' into style

* commit 209871a5ef3ed5ecc108f2361a92bc4f80dba2df
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:38:38 2024 +0300
|
|     Added README

* commit 429b59467712a57b6e8a487b5a3a645aafe429d5
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:36:36 2024 +0300
|
|     Updated index.html

* commit 9fd3d1044e344c7b66b54e1c96443d5b52ba5e39
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
```

Рис. 2.80: Возвращение на ветку style

Сбросим ветку style к нужному коммиту (рис. 2.81).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git reset --hard 429b59467712a57b6e8a487b5a3a645aafe429d5
HEAD is now at 429b594 Updated index.html
```

Рис. 2.81: Сброс ветки style к нужному коммиту

Проверим лог ветки style. У нас в истории больше нет коммитов слияний. (рис. 2.82).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ git log --graph --all
* commit 01aabd6128e50c2bbebbe9f4877e8e
| Author: Ivan-Mahorin <ivan.mahorin@ya
| Date:   Sat Feb 10 08:40:59 2024 +030
|
|     Life is great
|
* commit 209871a5ef3ed5ecc108f2361a92bc
| Author: Ivan-Mahorin <ivan.mahorin@ya
| Date:   Sat Feb 10 08:38:38 2024 +030
|
|     Added README
|
* commit 429b59467712a57b6e8a487b5a3a
| Author: Ivan-Mahorin <ivan.mahorin@ya
| Date:   Sat Feb 10 08:36:36 2024 +030
|
|     Updated index.html
|
* commit 9fd3d1044e344c7b66b54e1c9644
| Author: Ivan-Mahorin <ivan.mahorin@ya
| Date:   Sat Feb 10 08:35:34 2024 +030
|
|     Hello uses style.css
|
* commit 91bc742955e40067933109a44dc9
| Author: Ivan-Mahorin <ivan.mahorin@ya
| Date:   Sat Feb 10 08:34:17 2024 +030
|
|     Added css stylesheet
|
* commit aa60c28a292591cd94484ec7f92482
| Author: Ivan-Mahorin <ivan.mahorin@ya
| Date:   Sat Feb 10 08:26:24 2024 +030
```

Рис. 2.82: Проверка ветки

Сбросим ветку master (рис. 2.83).

```
* commit 209871a5ef3ed5ecc108f2361a92bc4f80dba2df (HEAD -> master)
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:38:38 2024 +0300
|
|     Added README
|
* commit 429b59467712a57b6e8a487b5a3a645aafe429d5 (style)
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:36:36 2024 +0300
|
|     Updated index.html
|
* commit 9fd3d1044e344c7b66b54e1c96443d5b52ba5e39
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:35:34 2024 +0300
|
|     Hello uses style.css
|
* commit 91bc742955e40067933109a44dc9f1eac193523a
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:34:17 2024 +0300
|
|     Added css stylesheet
|
* commit aa60c28a292591cd94484ec7f92482ed0e68e31e
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:26:24 2024 +0300
|
|     Added index.html.
|
* commit dfef41cf40f8b3b62cf6a239f85c81721a33008c
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:23:04 2024 +0300
|
|     Moved hello.html to lib
|
* commit ef99883e2934a668eb2705b88302bc7dfe54b5dd
| Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
| Date:   Sat Feb 10 08:20:43 2024 +0300
```

Рис. 2.83: Сброс ветки master

Используем команду rebase вместо команды merge. Мы вернулись в точку до первого слияния и хотим перенести изменения из ветки master в нашу ветку style. На этот раз для переноса изменений из ветки master мы будем использо-

вать команду git rebase вместо слияния (рис. 2.84).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git checkout style
Switched to branch 'style'

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git rebase master
Successfully rebased and updated refs/heads/style.

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ |
```

Рис. 2.84: Перебазирование

Выполним слияние style в master (рис. 2.85).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (style)
$ git checkout master
Switched to branch 'master'

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git merge style
Updating 209871a..4cc9ddd
Fast-forward
 index.html      | 3 +++
 lib/hello.html  | 1 +
 lib/style.css   | 3 +++
 3 files changed, 7 insertions(+)
 create mode 100644 lib/style.css

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.85: Слияние style в master

Просмотрим логи (рис. 2.86).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git log
commit 4cc9ddd70e4ccea08f7996411bf4a3f8831dcc6c (HEAD -> master, style)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:36:36 2024 +0300

    Updated index.html

commit 31adfc75eff26b4921a4caadb4487758f967f5e
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:35:34 2024 +0300

    Hello uses style.css

commit ba5d66da1d65fab7416559e73e403ed4e02ca940
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:34:17 2024 +0300

    Added css stylesheet

commit 209871a5ef3ed5ecc108f2361a92bc4f80dba2df
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:38:38 2024 +0300

    Added README

commit aa60c28a292591cd94484ec7f92482ed0e68e31e
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:26:24 2024 +0300

    Added index.html.

commit dfef41cf40f8b3b62cf6a239f85c81721a33008c
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
```

Рис. 2.86: Просмотр логов

Перейдём в рабочий каталог и сделаем клон нашего репозитория hello (рис. 2.87).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ cd

Ivan@DESKTOP-08NN2QL MINGW64 ~
$ cd MD/lab01

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ ls
hello/

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ pwd
/c/Users/Ivan/MD/lab01

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ |
```

Рис. 2.87: Клонирование репозиториев

Создадим клон репозитория (рис. 2.88).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ git clone hello cloned_hello
Cloning into 'cloned_hello'...
done.

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ ls
cloned_hello/  hello/

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ |
```

Рис. 2.88: Создание клона репозитория hello

Давайте взглянем на клонированный репозиторий (рис. 2.89).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ cd cloned_hello

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ ls
README.md index.html lib/
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ |
```

Рис. 2.89: Просмотр клонированного репозитория

Просмотрим историю репозитория (рис. 2.90).

```
commit 4cc9ddd/0e4ccea08f/996411bf4a3f8831dcc6c (HEAD -> master, origin/style, origin/master, origin/HEAD)
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:36:36 2024 +0300

    Updated index.html

commit 31adfc75eff26b4921a4caadb4487758f967f5e
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:35:34 2024 +0300

    Hello uses style.css

commit ba5d66da1d65fab7416559e73e403ed4e02ca940
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:34:17 2024 +0300

    Added css stylesheet

commit 209871a5ef3ed5ecc108f2361a92bc4f80dba2df
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:38:38 2024 +0300

    Added README

commit aa60c28a292591cd94484ec7f92482ed0e68e31e
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:26:24 2024 +0300

    Added index.html.

commit dfef41cf40f8b3b62cf6a239f85c81721a33008c
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:23:04 2024 +0300

    Moved hello.html to lib

commit ef99883e2934a668eb2705b88302bc7dfe54b5dd
Author: Ivan-Mahorin <iwan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:20:43 2024 +0300
```

Рис. 2.90: Просмотр истории репозитория

Мы видим, что клонированный репозиторий знает об имени по умолчанию удаленного репозитория. Давайте посмотрим, можем ли мы получить более по-

дробную информацию об имени по умолчанию (рис. 2.91).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ git remote
origin

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ git remote show origin
* remote origin
  Fetch URL: C:/Users/Ivan/MD/lab01/hello
  Push  URL: C:/Users/Ivan/MD/lab01/hello
  HEAD branch: master
  Remote branches:
    master tracked
    style tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (up to date)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ |
```

Рис. 2.91: Изучение origin

Давайте посмотрим на ветки, доступные в нашем клонированном репозитории (рис. 2.92).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ git branch
* master

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ |
```

Рис. 2.92: Просмотр доступных веток в клонированном репозитории

Для того, чтобы увидеть все ветки, попробуем следующую команду (рис. 2.93).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ |
```

Рис. 2.93: Просмотр всех веток

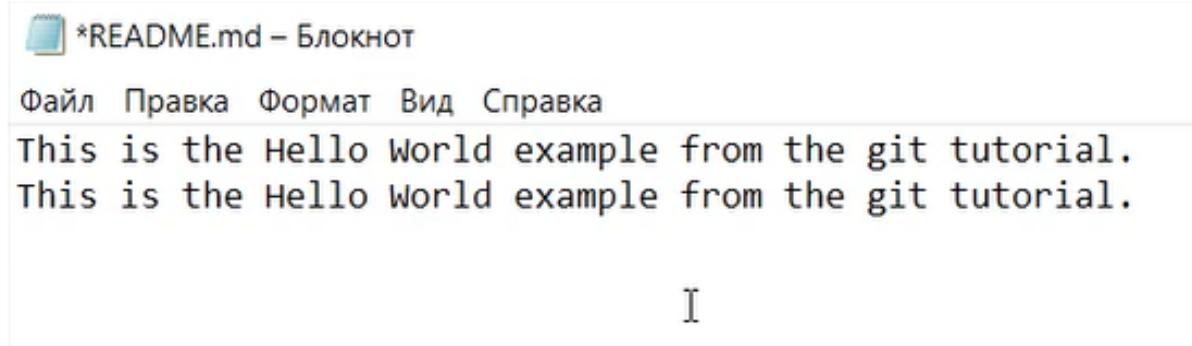
Перейдём в репозиторий hello (рис. 2.94).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ cd /c/Users/Ivan/MD/lab01/hello

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ |
```

Рис. 2.94: Переход в репозиторий hello

Внесём изменения в файл README.md (рис. 2.95).



```
*README.md – Блокнот
Файл Правка Формат Вид Справка
This is the Hello World example from the git tutorial.
This is the Hello World example from the git tutorial.
```

Рис. 2.95: Внесение изменения в файл README.md

Добавим это изменение и сделаем коммит (рис. 2.96).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add README.md

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit -m "Changed README in original repo"
[master 09a7f97] Changed README in original repo
 1 file changed, 1 insertion(+)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ |
```

Рис. 2.96: Добавление изменения и создание коммита

Научимся извлекать изменения из удаленного репозитория (рис. 2.97).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ cd /c/Users/Ivan/MD/lab01/cloned_hello

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 364 bytes | 19.00 KiB/s, done.
From C:/Users/Ivan/MD/lab01/hello
  4cc9ddd..09a7f97  master      -> origin/master

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ |
```

Рис. 2.97: Извлечение изменений

Мы можем продемонстрировать, что клонированный файл README.md не изменился (рис. 2.98).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ cat README.md
This is the Hello World example from the git tutorial.
```

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$
```

Рис. 2.98: Проверка README.md

Сольём извлеченные изменения в локальную ветку master (рис. 2.99).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ git merge origin/master
Updating 4cc9ddd..09a7f97
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
```

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$
```

Рис. 2.99: Слив извлеченных изменений в локальную ветку master

Теперь давайте рассмотрим объединение fetch и merge в одну команду (рис. 2.100).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ cat README.md
This is the Hello World example from the git tutorial.
This is the Hello World example from the git tutorial.
```

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ git pull
Already up to date.
```

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$
```

Рис. 2.100: Объединение fetch и merge в одну команду

Добавим локальную ветку, которая отслеживает удаленную ветку (рис. 2.101).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ git branch --track style origin/style
branch 'style' set up to track 'origin/style'.

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ git branch -a
* master
  style
    remotes/origin/HEAD -> origin/master
    remotes/origin/master
    remotes/origin/style

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ git log --max-count=2
commit 09a7f97a4944632d5766f32ba2caed6dac86be9b (HEAD -> master, origin/master, origin/HEAD)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:51:57 2024 +0300

    Changed README in original repo

commit 4cc9ddd70e4cceea08f7996411bf4a3f8831dcc6c (origin/style, style)
Author: Ivan-Mahorin <ivan.mahorin@yandex.ru>
Date:   Sat Feb 10 08:36:36 2024 +0300

    Updated index.html

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$
```

Рис. 2.101: Добавление локальной ветки, которая отслеживает удаленную ветку

Создадим чистый репозиторий (рис. 2.102).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ cd /c/Users/Ivan/MD/lab01

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ ls
cloned_hello/ hello/

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ git clone --bare hello hello.git
Cloning into bare repository 'hello.git'...
done.

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ ls
cloned_hello/ hello/ hello.git/

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ ls hello.git
HEAD config description hooks/ info/ objects/ packed-refs refs/
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$
```

Рис. 2.102: Создание чистого репозитория

Добавим удаленный репозиторий (рис. 2.103).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01
$ cd hello

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git remote add shared /c/Users/Ivan/MD/lab01/hello.git

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$
```

Рис. 2.103: Добавление удаленного репозитория

Отредактируем файл README.md (рис. 2.104).

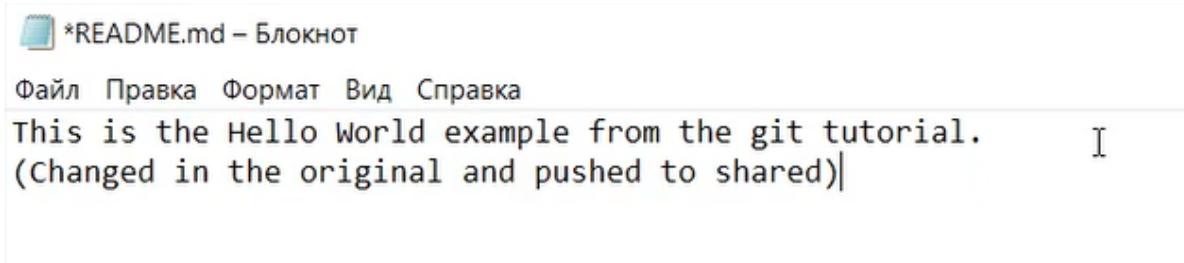


Рис. 2.104: Редактирование файла README.md

Сделаем коммит и отправим изменения в общий репозиторий (рис. 2.105).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git checkout master
Already on 'master'
M      README.md

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add README
fatal: pathspec 'README' did not match any files

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git add README.md

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git commit -m "Added shared comment to readme"
[master 8a566b6] Added shared comment to readme
 1 file changed, 1 insertion(+), 1 deletion(-)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ git push shared master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 407 bytes | 407.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To C:/Users/Ivan/MD/lab01/hello.git
 09a7f97..8a566b6  master -> master

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ |
```

Рис. 2.105: Коммит и отправка изменений в общий репозиторий

Научимся извлекать изменения из общего репозитория. Быстро переключимся в клонированный репозиторий и извлечём изменения, только что отправленные в общий репозиторий (рис. 2.106).

```
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/hello (master)
$ cd /c/Users/Ivan/MD/lab01/cloned_hello

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ git remote add shared /c/Users/Ivan/MD/lab01/hello.git

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ git branch --track shared master
branch 'shared' set up to track 'master'.

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ git pull shared master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 387 bytes | 11.00 KiB/s, done.
From C:/Users/Ivan/MD/lab01/hello
 * branch           master      -> FETCH_HEAD
 * [new branch]     master      -> shared/master
Updating 09a7f97..8a566b6
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ cat README.md
This is the Hello World example from the git tutorial.
(Changed in the original and pushed to shared)
Ivan@DESKTOP-08NN2QL MINGW64 ~/MD/lab01/cloned_hello (master)
$ |
```

Рис. 2.106: Извлечение общих изменений

3 Выводы

В результате выполнения лабораторной работы мы научились полному взаимодействию с git и разобрали основные его особенности.