

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКИЙ УНИВЕРСИТЕТ
ДРУЖБЫ НАРОДОВ**

Факультет физико-математических и естественных наук

Кафедра информационных технологий

ОТЧЕТ по лабораторной работе 10

ТЕМА «Работа с git »

по дисциплине «Математическое моделирование»

Выполнил:

Студент группы НПИбд-02-21

Студенческий билет № 1032205641

Сатлихана Петрити

Цель работы

Изучение работы с Git и знакомство с его утилитами.

Последовательность выполнения работы

1. Работа с git

1.1 Подготовка

1.1.1 Установка имени и электронной почты

Если вы никогда ранее не использовали git, для начала вам необходимо осуществить установку. Выполните следующие команды, чтобы git узнал ваше имя и электронную почту. Если git уже установлен, можете переходить к разделу окончания строк.

```
git config --global user.name "Your Name"
```

```
git config --global user.email "your_email@whatever.com"
```

```
Microsoft Windows [Version 10.0.22631.2506]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Acer>git config --global user.name "SatlihanaPetriti"

C:\Users\Acer>git config --global user.email "satlihanapetriti@gmail.com"

C:\Users\Acer>git config --global core.autocrlf true

C:\Users\Acer>git config --global core.safecrlf true

C:\Users\Acer>git config --global core.quotepath off

C:\Users\Acer>
```

рис.1- имя и электронную почту gita

1.1.2 Параметры установки окончаний строк

`core.autocrlf` configures newline character handling in Git:

`true`: Converts CRLF to LF on commit and vice versa on checkout (Windows)

`input`: Converts CRLF to LF on commit only (Mac/Linux).

`core.safecrlf`:

- `true`: Rejects irreversible LF <-> CRLF conversion.
- `warn`: Warns but accepts irreversible conversion.

Для пользователей Windows:

```
git config --global core.autocrlf true
```

```
git config --global core.safecrlf true
```

(посмотрите рис 1)

1.1.3 Установка отображения unicode

По умолчанию, git будет печатать не-ASCII символов в именах файлов в виде восьмеричных последовательностей \nnn. Что бы избежать нечитаемых строк, установите соответствующий флаг.

```
git config --global core.quotepath off
```

(посмотрите рис 1)

2. 1.2 Создание проекта

1.2.1 Создайте страницу «hello, world»

Начните работу в пустом рабочем каталоге с создания пустого каталога с именем hello, затем войдите в него и создайте там файл с именем hello.html

```
mkdir hello
```

```
cd hello
```

```
touch
```

```
hello.html
```

```
echo "hello, world!" > hello.html
```

```

C:\Users\Acer>cd C:\Users\Acer\Downloads\ADM\Mm
C:\Users\Acer\Downloads\ADM\Mm>mkdir hello
C:\Users\Acer\Downloads\ADM\Mm>ls
hello
C:\Users\Acer\Downloads\ADM\Mm>cd hello
C:\Users\Acer\Downloads\ADM\Mm\hello>touch hello.html
C:\Users\Acer\Downloads\ADM\Mm\hello>echo "Hello, World!" > hello.html
C:\Users\Acer\Downloads\ADM\Mm\hello>

```

рис.2- создание каталог, html файл

1.2.2 Создание репозитория

Чтобы создать git репозиторий из этого каталога, выполните команду git init.

```
git init
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>git init
Initialized empty Git repository in C:/Users/Acer/Downloads/ADM/Mm/hello/.git/
C:\Users\Acer\Downloads\ADM\Mm\hello>git add hello.html
C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Initial Commit"
[master (root-commit) 19ed51b] Initial Commit
1 file changed, 1 insertion(+)
create mode 100644 hello.html
C:\Users\Acer\Downloads\ADM\Mm\hello>git status
On branch master
nothing to commit, working tree clean
C:\Users\Acer\Downloads\ADM\Mm\hello>

```

рис.3- создать git репозиторий, добавление файла, проверка

1.2.3 Добавление файла в репозиторий

Добавим файл в репозиторий.

```
git add hello.html
```

```
git commit -m "Initial Commit"
```

(посмотрите рис 3)

1.2.4 Проверка состояние репозитория. Используйте команду git status чтобы проверить текущее состояние репозитория.

```
git status
```

(посмотрите рис 3)

3. 1.3 Внесение изменений

1.3.1 Измените страницу «Hello, World»

Добавим кое-какие HTML-теги к нашему приветствию. Измените содержимое файла hello.html на:

Hello, World!

Проверьте состояние рабочего каталога.

```
git status
```

4. 1.4 Индексация изменений

Теперь выполните команду git, чтобы проиндексировать изменения. Проверьте состояние.

```
git add hello.html
```

```
git status
```

1.4.1 Коммит изменений

Сделайте коммит и проверьте состояние.

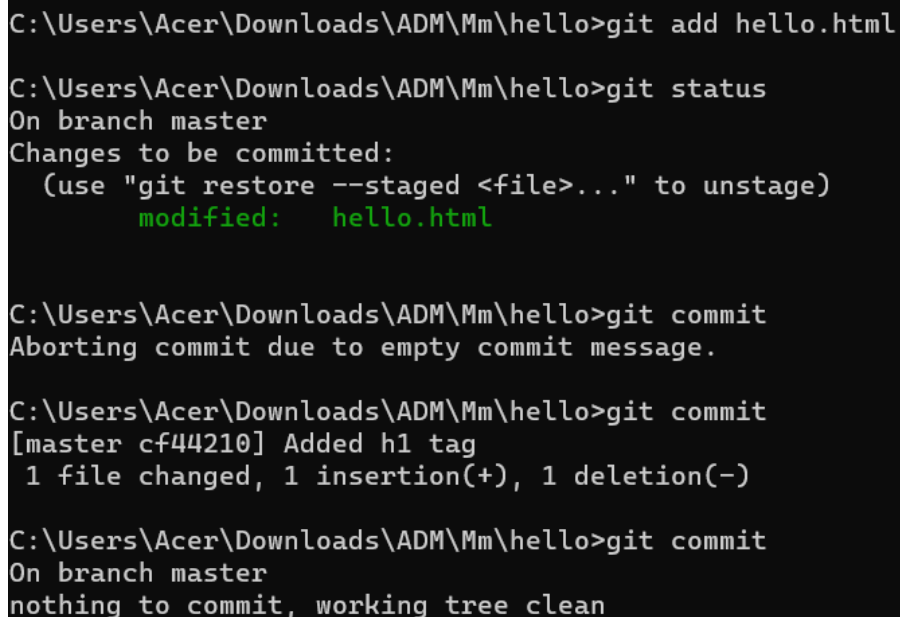
```
git commit
```

Откроется редактор. В первой строке введите комментарий: «Added h1 tag». Сохраните файл и выйдите из редактора (для этого в редакторе по-умолчанию (Vim) вам нужно нажать клавишу ESC, ввести :wq и нажать Enter).

Теперь еще раз проверим состояние.

```
git status
```

Рабочий каталог чистый, можно продолжить работу.



```
C:\Users\Acer\Downloads\ADM\Mm\hello>git add hello.html

C:\Users\Acer\Downloads\ADM\Mm\hello>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit
Aborting commit due to empty commit message.

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit
[master cf44210] Added h1 tag
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit
On branch master
nothing to commit, working tree clean
```

рис.4- Индексация изменений

1.4.2 Добавьте стандартные теги страницы

Измените страницу «Hello, World», чтобы она содержала стандартные теги и .

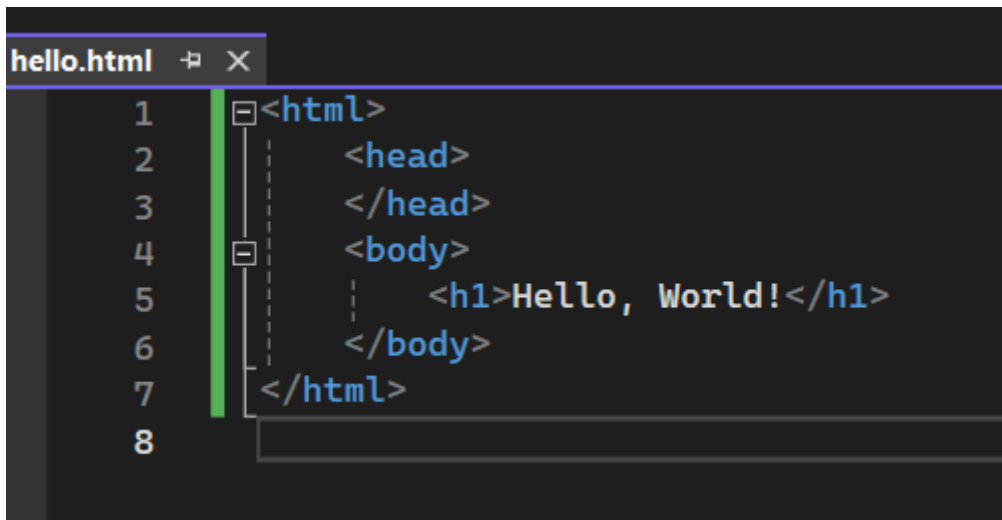


рис.5- добавим скрипт

Теперь добавьте это изменение в индекс git.

```
git add hello.html
```

Теперь добавьте заголовки HTML (секцию) к странице «Hello, World».

(посмотрите рис 5)

Проверьте текущий статус:

```
git status
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git add hello.html

C:\Users\Acer\Downloads\ADM\Mm\hello>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

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Added standard HTML page tags"
[master 145e701] Added standard HTML page tags
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Acer\Downloads\ADM\Mm\hello>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")

C:\Users\Acer\Downloads\ADM\Mm\hello>git add .

C:\Users\Acer\Downloads\ADM\Mm\hello>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Added HTML header"
[master c22d10a] Added HTML header
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Acer\Downloads\ADM\Mm\hello>git status
On branch master
nothing to commit, working tree clean

C:\Users\Acer\Downloads\ADM\Mm\hello>
```

рис.6- добавьте изменение в git.Проверьте статус.

Произведите коммит проиндексированного изменения (значение по умолчанию), а затем еще раз проверьте состояние.

```
git commit -m "Added standard HTML page tags"
```

```
git status
```

(посмотрите рис 6)

Теперь добавьте второе изменение в индекс, а затем проверьте состояние с помощью команды `git status`

```
git add .
```

```
git status
```

(посмотрите рис 6)

Второе изменение было проиндексировано и готово к коммиту. Сделайте коммит второго изменения

```
git commit -m "Added HTML header"
```

(посмотрите рис 6)

1.4.3 История

Получим список произведенных изменений:

```
git log
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git log
commit c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (HEAD -> master)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 23:19:05 2024 +0100

    Added HTML header

commit 145e701e1d01c0e40e11f79e3f9c13284ca51c59
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 23:18:13 2024 +0100

    Added standard HTML page tags

commit cf44210518c23990dce379d135ae8480030bdbbd
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 23:05:26 2024 +0100

    Added h1 tag

commit 19ed51bf0c44ff5b32155412cd0ea3db22ad27c8
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 22:50:43 2024 +0100

    Initial Commit

C:\Users\Acer\Downloads\ADM\Mm\hello>
```

рис.7- список произведенных изменений

Однострочный формат истории:

```
git log --pretty=oneline
```

Есть много вариантов отображения лога.

```
git log --pretty=oneline --max-count=2
```

```
git log --pretty=oneline --since='5 minutes ago'
```

```
git log --pretty=oneline --until='5 minutes ago'
```

```
git log --pretty=oneline --author=<your name>
```

```
git log --pretty=oneline --all
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git log --pretty=oneline
c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (HEAD -> master) Added HTML header
145e701e1d01c0e40e11f79e3f9c13284ca51c59 Added standard HTML page tags
cf44210518c23990dce379d135ae8480030bdbbd Added h1 tag
19ed51bf0c44ff5b32155412cd0ea3db22ad27c8 Initial Commit

C:\Users\Acer\Downloads\ADM\Mm\hello>git log --pretty=oneline --max-count=2
c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (HEAD -> master) Added HTML header
145e701e1d01c0e40e11f79e3f9c13284ca51c59 Added standard HTML page tags
```

рис.8- вариантов отображения лога(1)

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git log --pretty=oneline --since='11:35'
c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (HEAD -> master) Added HTML header
145e701e1d01c0e40e11f79e3f9c13284ca51c59 Added standard HTML page tags
cf44210518c23990dce379d135ae8480030bdbbd Added h1 tag
19ed51bf0c44ff5b32155412cd0ea3db22ad27c8 Initial Commit
```

рис.9- вариантов отображения лога(2)

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git log --pretty=oneline --all
c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (HEAD -> master) Added HTML header
145e701e1d01c0e40e11f79e3f9c13284ca51c59 Added standard HTML page tags
cf44210518c23990dce379d135ae8480030bdbbd Added h1 tag
19ed51bf0c44ff5b32155412cd0ea3db22ad27c8 Initial Commit
```

рис.10- вариантов отображения лога(3)

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git log --pretty=oneline --author=SatlihanaPetriti
c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (HEAD -> master) Added HTML header
145e701e1d01c0e40e11f79e3f9c13284ca51c59 Added standard HTML page tags
cf44210518c23990dce379d135ae8480030bdbbd Added h1 tag
19ed51bf0c44ff5b32155412cd0ea3db22ad27c8 Initial Commit
```

рис.11- вариантов отображения лога(4)

Справочная страница:

`man git-log`

Инструмент `gitk` полезен в изучении истории изменений.

The screenshot shows the `gitk` graphical user interface. On the left, a terminal window displays the command `man git-log` and its output, which states that `'man'` is not recognized. The main window shows the commit history as a list of commits with their SHA1 IDs and descriptions. The commit `145e701e1d01c0e40e11f79e3f9c13284ca51c59` is selected, and the right pane shows a diff view of the changes made in this commit. The diff shows the addition of standard HTML page tags and an h1 tag to the `hello.html` file. The terminal window also shows the command `git log --help` and the command `gitk` being executed.

рис.12- Инструмент `gitk`

1.4.4 Получение старых версий

Возвращаться назад в историю очень просто. Команда `checkout` скопирует любой снимок из репозитория в рабочий каталог.

Получите хэши предыдущих версий

```
git log
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git log
commit c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (HEAD -> master)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 23:19:05 2024 +0100

    Added HTML header

commit 145e701e1d01c0e40e11f79e3f9c13284ca51c59
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 23:18:13 2024 +0100

    Added standard HTML page tags

commit cf44210518c23990dce379d135ae8480030bdbbd
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 23:05:26 2024 +0100

    Added h1 tag

commit 19ed51bf0c44ff5b32155412cd0ea3db22ad27c8
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 22:50:43 2024 +0100

    Initial Commit
```

рис.13- хэши предыдущих версий

Затем проверьте содержимое файла `hello.html`.

```
git checkout <hash>
cat hello.html
```

Вернитесь к последней версии в ветке `master`

```
git checkout master
cat hello.html
```

`master` — имя ветки по умолчанию. Переключая имена веток, вы попадаете на последнюю версию выбранной ветки.

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout cf44210518c23990dce379d135ae8480030bdbbd
Previous HEAD position was 19ed51b Initial Commit
HEAD is now at cf44210 Added h1 tag

C:\Users\Acer\Downloads\ADM\Mm\hello>cat hello.html
"<h1>Hello, World!</h1>"

C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout master
Previous HEAD position was cf44210 Added h1 tag
Switched to branch 'master'

C:\Users\Acer\Downloads\ADM\Mm\hello>cat hello.html
</html>>ody>>Hello, World!</h1>

C:\Users\Acer\Downloads\ADM\Mm\hello>
```

рис.14- проверьте содержимое файла `hello.html`, вернитесь к ветке `master`

1.4.5 Создание тегов версий

Давайте назовем текущую версию страницы hello первой (v1).Создайте тег первой версии

```
git tag v1
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git tag v1  
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout v1^  
More?
```

рис.15- Создание тегов версий

Теперь текущая версия страницы называется v1.

Теги для предыдущих версий Давайте создадим тег для версии, которая идет перед текущей версией и назовем его v1-beta. В первую очередь нам надо переключиться на предыдущую версию. Вместо поиска до хэш, мы будем использовать ^, обозначающее «родитель v1». Вместо обозначения v1^ можно использовать v1~1. Это обозначение можно определить как «первую версию предшествующую v1».

```
git checkout v1^`
```

```
`cat hello.html
```

Это версия с тегами и , но еще пока без . Давайте сделаем ее версией v1-beta.

```
git tag v1-beta
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout v1~1  
Note: switching to 'v1~1'.  
  
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 145e701 Added standard HTML page tags  
C:\Users\Acer\Downloads\ADM\Mm\hello>cat hello.html  
</html>>Hello, World!</h1>  
C:\Users\Acer\Downloads\ADM\Mm\hello>git tag v1-beta
```

рис.16-Создание тегов версий, beta, v1

1.4.6 Переключение по имени тега

Теперь попробуйте попереключаться между двумя отмеченными версиями.

```
git checkout v1
```

```
git checkout v1-beta
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout v1  
Previous HEAD position was 145e701 Added standard HTML page tags  
HEAD is now at c22d10a Added HTML header  
  
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout v1-beta  
Previous HEAD position was c22d10a Added HTML header  
HEAD is now at 145e701 Added standard HTML page tags
```

рис.17-Переключение по имени тега

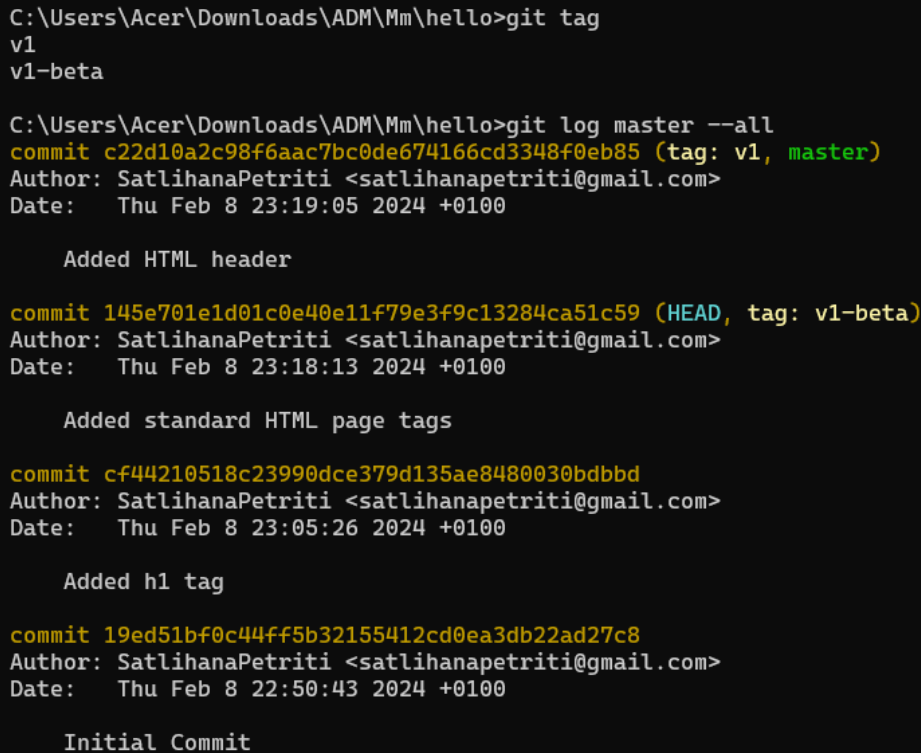
1.4.7 Просмотр тегов с помощью команды tag

Вы можете увидеть, какие теги доступны, используя команду `git tag`.

```
git tag
```

Вы также можете посмотреть теги в логе.

```
git log master --all
```



```
C:\Users\Acer\Downloads\ADM\Mm\hello>git tag
v1
v1-beta

C:\Users\Acer\Downloads\ADM\Mm\hello>git log master --all
commit c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (tag: v1, master)
Author: SatlihanPetriti <satlihanapetrिति@gmail.com>
Date: Thu Feb 8 23:19:05 2024 +0100

    Added HTML header

commit 145e701e1d01c0e40e11f79e3f9c13284ca51c59 (HEAD, tag: v1-beta)
Author: SatlihanPetriti <satlihanapetrिति@gmail.com>
Date: Thu Feb 8 23:18:13 2024 +0100

    Added standard HTML page tags

commit cf44210518c23990dce379d135ae8480030bdbbd
Author: SatlihanPetriti <satlihanapetrिति@gmail.com>
Date: Thu Feb 8 23:05:26 2024 +0100

    Added h1 tag

commit 19ed51bf0c44ff5b32155412cd0ea3db22ad27c8
Author: SatlihanPetriti <satlihanapetrिति@gmail.com>
Date: Thu Feb 8 22:50:43 2024 +0100

    Initial Commit
```

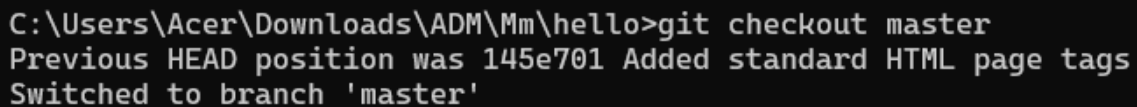
рис.18-Просмотр тегов с помощью команды tag

5. 1.5 Отмена локальных изменений (до индексации)

1.5.1 Переключитесь на ветку master

Убедитесь, что вы находитесь на последнем коммите ветки master, прежде чем продолжить работу.

```
git checkout master
```



```
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout master
Previous HEAD position was 145e701 Added standard HTML page tags
Switched to branch 'master'
```

рис.19-Переключитесь на ветку master

1.5.2 Измените hello.html

Иногда случается, что вы изменили файл в рабочем каталоге, и хотите отменить последние коммиты. С этим справится команда `git checkout`. Внесите изменение в файл `hello.html` в виде нежелательного комментария.

```
<html>``
```

```
``<body><h1>hello, world!</h1>
```

```
<!-- This is a bad comment. We want to revert it. -->
```

```
</body>
```

```
</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. -->
7      </body>
8  </html>
9

```

рис.20-Измените hello.html

1.5.3 Проверьте состояние

Сначала проверьте состояние рабочего каталога.

```
git status
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>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")

```

рис.21-Проверьте состояние рабочего каталога

Мы видим, что файл hello.html был изменен, но еще не проиндексирован.

1.5.4 Отмена изменений в рабочем каталоге

Используйте команду git checkout для переключения версии файла hello.html в репозитории.

```
git checkout hello.html
```

```
git status
```

```
cat hello.html
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout hello.html
Updated 1 path from the index

C:\Users\Acer\Downloads\ADM\Mm\hello>git status
On branch master
nothing to commit, working tree clean

C:\Users\Acer\Downloads\ADM\Mm\hello>cat hello.html
</html>ody>>Hello, World!</h1>

C:\Users\Acer\Downloads\ADM\Mm\hello>

```

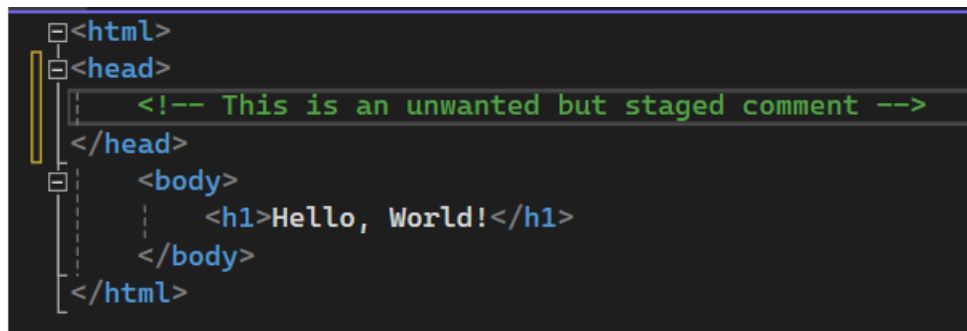
рис.22-Отмена изменений в рабочем каталоге

Команда git status показывает нам, что не было произведено никаких изменений, не зафиксированных в рабочем каталоге.

6. 1.6 Отмена проиндексированных изменений (перед коммитом)

1.6.1 Измените файл и проиндексируйте изменения

Внесите изменение в файл hello.html в виде нежелательного комментария



```

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

```

рис.23 изменение в файл hello.html

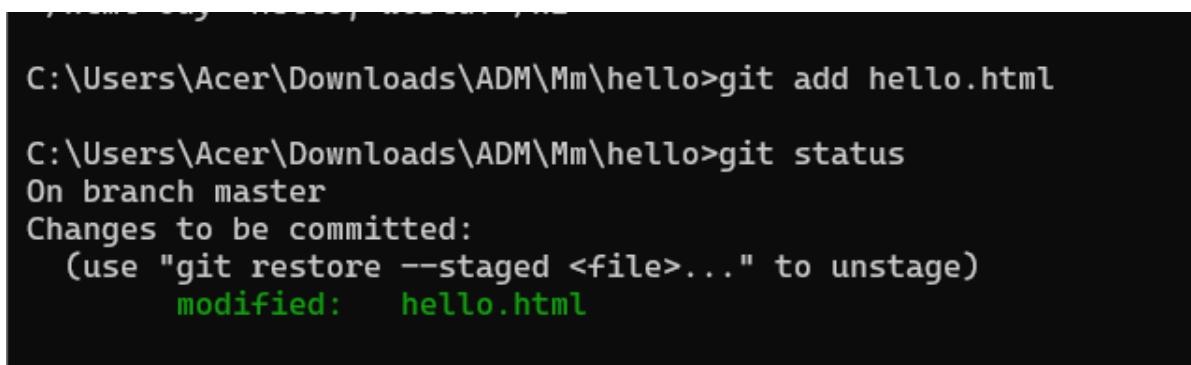
Проиндексируйте это изменение.

```
git add hello.html
```

1.6.2 Проверьте состояние

Проверьте состояние нежелательного изменения.

```
git status
```



```

C:\Users\Acer\Downloads\ADM\Mm\hello>git add hello.html

C:\Users\Acer\Downloads\ADM\Mm\hello>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

```

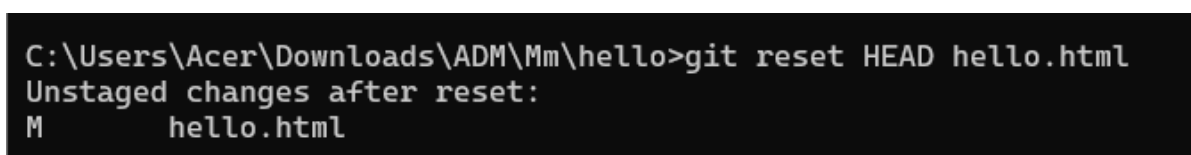
рис.24 индексируем это изменение и проверь состояние

Состояние показывает, что изменение было проиндексировано и готово к коммиту.

1.6.3 Выполните сброс буферной зоны

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

```
git reset HEAD hello.html
```



```

C:\Users\Acer\Downloads\ADM\Mm\hello>git reset HEAD hello.html
Unstaged changes after reset:
M      hello.html

```

рис.25 сброс буферной зоны

Команда `git reset` сбрасывает буферную зону к HEAD.

Команда `git reset` (по умолчанию) не изменяет рабочий каталог. Поэтому рабочий каталог все еще содержит нежелательный комментарий.

Мы можем использовать команду `git checkout`, чтобы удалить нежелательные изменения в рабочем каталоге.

1.6.4 Переключитесь на версию коммита

```
git checkout hello.html
```

```
git status
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout hello.html
Updated 1 path from the index

C:\Users\Acer\Downloads\ADM\Mm\hello>git status
On branch master
nothing to commit, working tree clean
```

рис.26 Переключат на версию коммита

Наш рабочий каталог опять чист.

7. 1.7 Отмена коммитов

1.7.2 Измените файл и сделайте коммит

Измените файл hello.html на следующий.

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

рис.27 Измените файл hello.html

Выполните:

```
git add hello.html
```

```
git commit -m "Oops, we didn't want this commit"
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git add hello.html

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Oops, we didn't want this commit"
[master 3a91dcb] Oops, we didn't want this commit
1 file changed, 1 insertion(+), 1 deletion(-)
```

рис.28 Выполним команды

1.7.3 Сделайте коммит с новыми изменениями, отменяющими предыдущие

Чтобы отменить коммит, нам необходимо сделать коммит, который удаляет изменения, сохраненные нежелательным коммитом.

```
git revert HEAD
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git revert HEAD
[master 5e4821b] Revert "Oops, we didn't want this commit"
1 file changed, 1 insertion(+), 1 deletion(-)
```

рис.29 git revert HEAD

Перейдите в редактор, где вы можете отредактировать коммит-сообщение по умолчанию или оставить все как есть. Сохраните и закройте файл.

```
Revert "Oops, we didn't want this commit"
```

```
This reverts commit 3a91dcbb9d8ecfa0d1f13757d001b974b5cd5f18.
```

```
# 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.30 редактор

Так как мы отменили самый последний произведенный коммит, мы смогли использовать HEAD в качестве аргумента для отмены. Мы можем отменить любой произвольный коммит в истории, указав его хэш-значение.

1.7.4 Проверьте лог

Проверка лога показывает нежелательные и отмененные коммиты в наш репозиторий.

git log

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git log
commit 5e4821be630ebd5d9657630fd0a80a6786ba24b9 (HEAD -> master)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 00:43:01 2024 +0100

    Revert "Oops, we didn't want this commit"

    This reverts commit 3a91dcbb9d8ecfa0d1f13757d001b974b5cd5f18.

commit 3a91dcbb9d8ecfa0d1f13757d001b974b5cd5f18
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 00:42:34 2024 +0100

    Oops, we didn't want this commit

commit c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (tag: v1)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 23:19:05 2024 +0100

    Added HTML header

commit 145e701e1d01c0e40e11f79e3f9c13284ca51c59 (tag: v1-beta)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 23:18:13 2024 +0100

    Added standard HTML page tags

commit cf44210518c23990dce379d135ae8480030bdbbd
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 23:05:26 2024 +0100

    Added h1 tag

commit 19ed51bf0c44ff5b32155412cd0ea3db22ad27c8
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 22:50:43 2024 +0100

    Initial Commit
```

рис.31 нежелательные и отмененные коммиты

Эта техника будет работать с любым коммитом

8. 1.8 Удаление коммитов из ветки

git revert является мощной командой, которая позволяет отменить любые коммиты в репозиторий. Однако, и оригинальный и «отмененный» коммиты видны в истории ветки (при использовании команды git log).

1.8.1 Команда git reset

При получении ссылки на коммит (т.е. хэш, ветка или имя тега), команда gitreset:

- перепишет текущую ветку, чтобы она указывала на нужный коммит;
- опционально сбросит буферную зону для соответствия с указанным коммитом;
- опционально сбросит рабочий каталог для соответствия с указанным коммитом.

1.8.2 Проверьте нашу историю

Давайте сделаем быструю проверку нашей истории коммитов. Выполните:

```
git log
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git log
commit 5e4821be630ebd5d9657630fd0a80a6786ba24b9 (HEAD -> master)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 00:43:01 2024 +0100

    Revert "Oops, we didn't want this commit"

    This reverts commit 3a91dcbb9d8ecfa0d1f13757d001b974b5cd5f18.

commit 3a91dcbb9d8ecfa0d1f13757d001b974b5cd5f18
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 00:42:34 2024 +0100

    Oops, we didn't want this commit

commit c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (tag: v1)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 23:19:05 2024 +0100

    Added HTML header

commit 145e701e1d01c0e40e11f79e3f9c13284ca51c59 (tag: v1-beta)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 23:18:13 2024 +0100

    Added standard HTML page tags

commit cf44210518c23990dce379d135ae8480030bdbbd
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 23:05:26 2024 +0100

    Added h1 tag

commit 19ed51bf0c44ff5b32155412cd0ea3db22ad27c8
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 22:50:43 2024 +0100

    Initial Commit
```

рис.32 проверку нашей истории коммитов

Мы видим, что два последних коммита в этой ветке — «Oops» и «Revert Oops». Давайте удалим их с помощью сброса.

1.8.3 Для начала отметьте эту ветку

Но прежде чем удалить коммиты, давайте отметим последний коммит тегом, чтобы потом можно

было его найти.

```
git tag oops
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git tag oops
```

рис.33 отметим последний коммит тегом

1.8.4 Сброс коммитов к предшествующим коммиту Oops

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).

```
git reset --hard v1
```

```
git log
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git reset --hard v1
HEAD is now at c22d10a Added HTML header

C:\Users\Acer\Downloads\ADM\Mm\hello>git log
commit c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (HEAD -> master,
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 23:19:05 2024 +0100

    Added HTML header

commit 145e701e1d01c0e40e11f79e3f9c13284ca51c59 (tag: v1-beta)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 23:18:13 2024 +0100

    Added standard HTML page tags

commit cf44210518c23990dce379d135ae8480030bdbbd
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 23:05:26 2024 +0100

    Added h1 tag

commit 19ed51bf0c44ff5b32155412cd0ea3db22ad27c8
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 22:50:43 2024 +0100

    Initial Commit
```

рис.34 git reset --hard v1, git log

Наша ветка master теперь указывает на коммит v1, а коммитов Oops и RevertOops в ветке уже нет. Параметр --hard указывает, что рабочий каталог должен быть обновлен в соответствии с новым head ветки.

1.8.5 Ничего никогда не теряется

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.

```
git log --all
```



```

C:\Users\Acer\Downloads\ADM\Mm\hello>git log --all
commit 5e4821be630ebd5d9657630fd0a80a6786ba24b9 (tag: oops)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 00:43:01 2024 +0100

    Revert "Oops, we didn't want this commit"

    This reverts commit 3a91dcbb9d8ecfa0d1f13757d001b974b5cd5f18.

commit 3a91dcbb9d8ecfa0d1f13757d001b974b5cd5f18
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 00:42:34 2024 +0100

    Oops, we didn't want this commit

commit c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (HEAD -> master, tag: v1)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 23:19:05 2024 +0100

    Added HTML header

commit 145e701e1d01c0e40e11f79e3f9c13284ca51c59 (tag: v1-beta)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 23:18:13 2024 +0100

    Added standard HTML page tags

commit cf44210518c23990dce379d135ae8480030bdbbd
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 23:05:26 2024 +0100

    Added h1 tag

commit 19ed51bf0c44ff5b32155412cd0ea3db22ad27c8
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 22:50:43 2024 +0100

    Initial Commit

```

рис.35 все коммиты

Мы видим, что ошибочные коммиты не исчезли. Они все еще находятся в репозитории. Просто они отсутствуют в ветке master.

9. 1.9 Удаление тега oops

1.9.1 Удаление тега oops

Тег oops свою функцию выполнил. Давайте удалим его и коммиты, на которые он ссылался, сборщиком мусора.

```
git tag -d oops
```

```
git log --all
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>git tag -d oops
Deleted tag 'oops' (was 5e4821b)

C:\Users\Acer\Downloads\ADM\Mm\hello>git log --all
commit c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (HEAD -> master, tag: v1)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 23:19:05 2024 +0100

    Added HTML header

commit 145e701e1d01c0e40e11f79e3f9c13284ca51c59 (tag: v1-beta)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 23:18:13 2024 +0100

    Added standard HTML page tags

commit cf44210518c23990dce379d135ae8480030bdbbd
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 23:05:26 2024 +0100

    Added h1 tag

commit 19ed51bf0c44ff5b32155412cd0ea3db22ad27c8
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date: Thu Feb 8 22:50:43 2024 +0100

    Initial Commit

```

рис.36 Удаление тега oops

Тег «oops» больше не будет отображаться в репозитории.

10. 1.10 Внесение изменений в коммиты

1.10.1 Измените страницу, а затем сделайте коммит

Добавьте в страницу комментарий автора (вставьте свою фамилию).

```

<!-- Author:Satlihana Petriti -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```

рис.37 Изменим страницу

Выполните:

```
git add hello.html
```

```
git commit -m "Add an author comment"
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>git add hello.html

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Add an author comment"
[master 854cfff] Add an author comment
1 file changed, 1 insertion(+), 1 deletion(-)

```

рис.38 выполнения команды

1.10.2 Необходим email

```

<!-- Author:Satlihana Petriti (satlihanapetriti@gmail.com) -->
<html>
<head>
</head>
<body>
  <h1>Hello, World!</h1>
</body>
</html>

```

рис.39 Изменим страницу

После совершения коммита вы понимаете, что любой хороший комментарий должен включать электронную почту автора. Обновите страницу hello, включив в нее email.

1.10.3 Измените предыдущий коммит

Мы действительно не хотим создавать отдельный коммит только ради электронной почты.

Давайте изменим предыдущий коммит, включив в него адрес электронной почты. Выполните:

```

git add hello.html
git commit --amend -m "Add an author/email comment"

```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>git add hello.html

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit --amend -m "Add an author/email comment"
[master 7dbd15b] Add an author/email comment
Date: Fri Feb 9 01:02:11 2024 +0100
1 file changed, 1 insertion(+), 1 deletion(-)

```

рис.40 выполнения команды

1.10.4 Просмотр истории

Выполните:

```

git log

```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>git log
commit 7dbd15b6f9395f4fc47278dd60fca66f5685fe4d (HEAD -> master)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 01:02:11 2024 +0100

    Add an author/email comment

commit c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (tag: v1)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 23:19:05 2024 +0100

    Added HTML header

commit 145e701e1d01c0e40e11f79e3f9c13284ca51c59 (tag: v1-beta)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 23:18:13 2024 +0100

    Added standard HTML page tags

commit cf44210518c23990dce379d135ae8480030bdbbd
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 23:05:26 2024 +0100

    Added h1 tag

commit 19ed51bf0c44ff5b32155412cd0ea3db22ad27c8
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Thu Feb 8 22:50:43 2024 +0100

    Initial Commit

```

рис.41 Просмотр истории

Мы можем увидеть, что оригинальный коммит «автор» заменен коммитом «автор/email». Этого же эффекта можно достичь путем сброса последнего коммита в ветке, и повторного коммита новых изменений.

11. 1.11 Перемещение файлов

1.11.1 Переместите файл hello.html в каталог lib

Сейчас мы собираемся создать структуру нашего репозитория. Давайте перенесем страницу в каталог lib.

```
mkdir lib
```

```
git mv hello.html lib
```

```
git status
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>mkdir lib

C:\Users\Acer\Downloads\ADM\Mm\hello>git mv hello.html lib

C:\Users\Acer\Downloads\ADM\Mm\hello>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    hello.html -> lib/hello.html

```

рис.42 Переместите файл hello.html в каталог lib

Перемещая файлы с помощью git mv, мы информируем git о 2 вещах:

- Что файл hello.html был удален.
- Что файл lib/hello.html был создан.
- Оба эти факта сразу же проиндексированы и готовы к коммиту. Команда git status сообщает, что файл был перемещен

12. 1.12 Второй способ перемещения файлов

Положительной чертой git является то, что вы можете забыть о версионном контроле до того момента, когда вы готовы приступить к коммиту кода.

Следующий набор команд идентичен нашим последним действиям. Работы здесь побольше, но результат тот же.

Мы могли бы выполнить:

```
mkdir lib
```

```
mv hello.html lib
```

```
git add lib/hello.html
```

```
git rm hello.html
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>mkdir lib

C:\Users\Acer\Downloads\ADM\Mm\hello>git mv hello.html lib

C:\Users\Acer\Downloads\ADM\Mm\hello>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    hello.html -> lib/hello.html

C:\Users\Acer\Downloads\ADM\Mm\hello>git add lib/hello.html

C:\Users\Acer\Downloads\ADM\Mm\hello>git rm hello.html
fatal: pathspec 'hello.html' did not match any files

```

рис.43 Второй способ перемещения файлов

1.12.1 Коммит в новый каталог

Давайте сделаем коммит этого перемещения:

```
git commit -m "Moved hello.html to lib"
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Moved hello.html to lib"
[master 38c5346] Moved hello.html to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)
```

рис.44 Коммит в новый каталог

13. 1.13 Подробнее о структуре

1.13.1 Добавление index.html

Добавим файл index.html в наш репозиторий

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

рис.45 Добавление index.html

Hello,
World!

рис.46 результат

Добавьте файл и сделайте коммит.

```
git add index.html
```

```
git commit -m "Added index.html."
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git add index.html

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Added index.html."
[master 3e5fd72] Added index.html.
1 file changed, 1 insertion(+)
create mode 100644 index.html
```

рис.46 Добавьте файл и сделайте коммит

Теперь при открытии index.html, вы должны увидеть кусок страницы hello в маленьком окошке.

14. 1.14 Git внутри: Каталог .git

1.14.1 Каталог .git

Выполните:

```
ls -C .git
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>ls -C .git
COMMIT_EDITMSG  ORIG_HEAD  description  hooks  info  objects  refs
HEAD           config     gitk.cache   index  logs  packed-refs
```

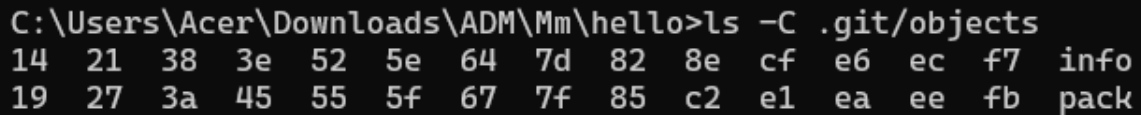
рис.47 Каталог .git

Это каталог, в котором хранится вся информация git.

1.14.2 База данных объектов

Выполните:

```
ls -C .git/objects
```



```
C:\Users\Acer\Downloads\ADM\Mm\hello>ls -C .git/objects
14 21 38 3e 52 5e 64 7d 82 8e cf e6 ec f7 info
19 27 3a 45 55 5f 67 7f 85 c2 e1 ea ee fb pack
```

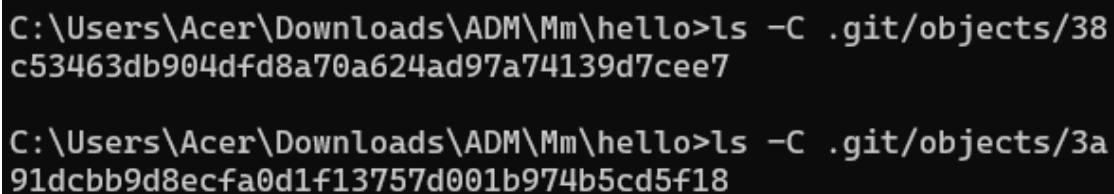
рис.48 База данных объектов

Вы должны увидеть набор каталогов, имена которых состоят из 2 символов. Имена каталогов являются первыми двумя буквами хэша sha1 объекта, хранящегося в git.

1.14.3 Углубляемся в базу данных объектов

Выполните:

```
ls -C .git/objects/<dir>
```



```
C:\Users\Acer\Downloads\ADM\Mm\hello>ls -C .git/objects/38
c53463db904dfd8a70a624ad97a74139d7cee7

C:\Users\Acer\Downloads\ADM\Mm\hello>ls -C .git/objects/3a
91dcbb9d8ecfa0d1f13757d001b974b5cd5f18
```

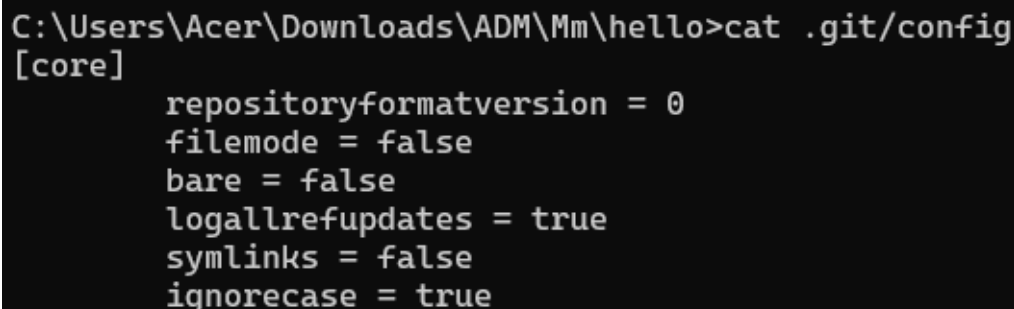
рис.49 Углубляемся в базу данных объектов

Смотрим в один из каталогов с именем из 2 букв. Вы увидите 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.

1.14.4 Config File

Выполните:

```
cat .git/config
```



```
C:\Users\Acer\Downloads\ADM\Mm\hello>cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
```

рис.50 Config File

Это файл конфигурации, создающийся для каждого конкретного проекта. Записи в этом файле будут перезаписывать записи в файле .gitconfig вашего главного каталога, по крайней мере в рамках этого проекта.

1.14.5 Ветки и теги

Выполните:

```
ls .git/refs
```

```
ls .git/refs/heads
```

```
ls .git/refs/tags
```

```
cat .git/refs/tags/v1
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>ls .git/refs
heads  tags

C:\Users\Acer\Downloads\ADM\Mm\hello>ls .git/refs/heads
master

C:\Users\Acer\Downloads\ADM\Mm\hello>ls .git/refs/tags
v1  v1-beta

C:\Users\Acer\Downloads\ADM\Mm\hello>cat .git/refs/tags/v1
c22d10a2c98f6aac7bc0de674166cd3348f0eb85

```

рис.51 Ветки и теги

Вы должны узнавать файлы в подкаталоге тегов. Каждый файл соответствует тегу, ранее созданному с помощью команды `git tag`. Его содержание — это всего лишь хэш коммита, привязанный к тегу.

1.14.6 Файл HEAD

Выполните:

```
cat .git/HEAD
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>cat .git/HEAD
ref: refs/heads/master

```

рис.52 Файл HEAD

Файл HEAD содержит ссылку на текущую ветку, в данный момент это должна быть ветка `master`.

15. 1.15 Работа непосредственно с объектами git

1.15.1 Поиск последнего коммита

Выполните:

```
git log --max-count=1
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>git log --max-count=1
commit 3e5fd727a5e2f58f4398e628471207cc16b52a5d (HEAD -> master)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 01:26:28 2024 +0100

    Added index.html.

```

рис.53 Поиск последнего коммита

Эта команда должна показать последний коммит в репозитории. SHA1 хэш в вашей системе, вероятно, отличается от моего, но вы увидите что-то наподобие этого.

1.15.2 Вывод последнего коммита с помощью SHA1 хэша

Выполните:

```
git cat-file -t <hash>
```

```
git cat-file -p <hash>
```



```
C:\Users\Acer\Downloads\ADM\Mm\hello>git cat-file -t 3e5fd727a5e2f58f4398e628471207cc16b52a5d
commit

C:\Users\Acer\Downloads\ADM\Mm\hello>git cat-file -p 3e5fd727a5e2f58f4398e628471207cc16b52a5d
tree 7f3d993c168186e4a4f57ac53a6eee33a20d82f3
parent 38c53463db904dfd8a70a624ad97a74139d7cee7
author SatlihanPetriti <satlihanapetriti@gmail.com> 1707438388 +0100
committer SatlihanPetriti <satlihanapetriti@gmail.com> 1707438388 +0100

Added index.html.
```

рис.54 Вывод последнего коммита с помощью SHA1 хэша

1.15.3 Поиск дерева

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

Выполните:

```
git cat-file -p <treehash>
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git cat-file -p 7f3d993c168186e4a4f57ac53a6eee33a20d82f3
100644 blob e14d4a02c87d15f67a1fb2338468e079f79309d1 index.html
040000 tree 45764eef5af33561da3c80505f5fa3a6f0ead5fa lib
```

рис.55 Поиск дерева

1.15.4 Вывод каталога lib

Выполните:

```
git cat-file -p <libhash>
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git cat-file -p 45764eef5af33561da3c80505f5fa3a6f0ead5fa
100644 blob e6ff301e875c624d0e9e1989179bb6ef0e8c47bc hello.html
```

рис.56 Вывод каталога lib

1.15.5 Вывод файла hello.html

Выполните:

```
git cat-file -p <hellohash>
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git cat-file -p e6ff301e875c624d0e9e1989179bb6ef0e8c47bc
</html>>Hello, World!</html>iti (satlihanapetriti@gmail.com) -->
```

рис.57 Вывод файла hello.html

1.15.6 Исследуйте самостоятельно

Исследуйте git репозиторий вручную самостоятельно. Смотрите, удастся ли вам найти оригинальный файл hello.html с самого первого коммита вручную по ссылкам SHA1 хэша в последнем коммите.

```

C:\Users\Acer\Downloads\ADM\Mm\hello\lib>git rev-list --max-parents=0 HEAD
19ed51bf0c44ff5b32155412cd0ea3db22ad27c8

C:\Users\Acer\Downloads\ADM\Mm\hello\lib>git cat-file -p 19ed51bf0c44ff5b32155412cd0ea3db22ad27c8
tree ea95e17ea6fe834a7d915ba1c22916628aef1a7e
author SatlihanaPetriti <satlihanapetriti@gmail.com> 1707429043 +0100
committer SatlihanaPetriti <satlihanapetriti@gmail.com> 1707429043 +0100

Initial Commit

C:\Users\Acer\Downloads\ADM\Mm\hello\lib>git cat-file -p ea95e17ea6fe834a7d915ba1c22916628aef1a7e
100644 blob 21d47457bd178004e7c495b214908ee44524b6e5    hello.html

C:\Users\Acer\Downloads\ADM\Mm\hello\lib>git cat-file -p 21d47457bd178004e7c495b214908ee44524b6e5
"Hello, World!"

```

рис.58 оригинальный файл hello.html с самого первого коммита

16. 1.16 Создание ветки

Пора сделать наш hello world более выразительным. Так как это может занять некоторое время, лучше переместить эти изменения в отдельную ветку, чтобы изолировать их от изменений в ветке master.

1.16.1 Создайте ветку

Давайте назовем нашу новую ветку «style». Выполните:

```
git checkout -b style
```

```
git status
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello\lib>git checkout -b style
Switched to a new branch 'style'

C:\Users\Acer\Downloads\ADM\Mm\hello\lib>git status
On branch style
nothing to commit, working tree clean

```

рис.59 Создайте ветку

`git checkout -b <имя ветки>` является шорткатом для `git branch <имя ветки>` за которым идет `git checkout <имя ветки>`.

Обратите внимание, что команда `git status` сообщает о том, что вы находитесь в ветке «style».

1.16.2 Добавьте файл стилей style.css

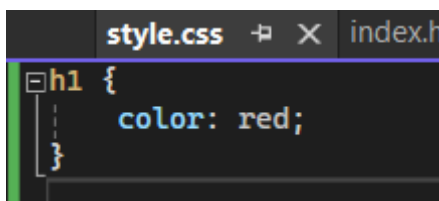


рис.60 Добавьте файл стилей style.css

Выполните:

```
touch lib/style.css
```

Файл lib/style.css:

```

h1 {
  color: red;
}

```

Выполните:

```
git add lib/style.css
```

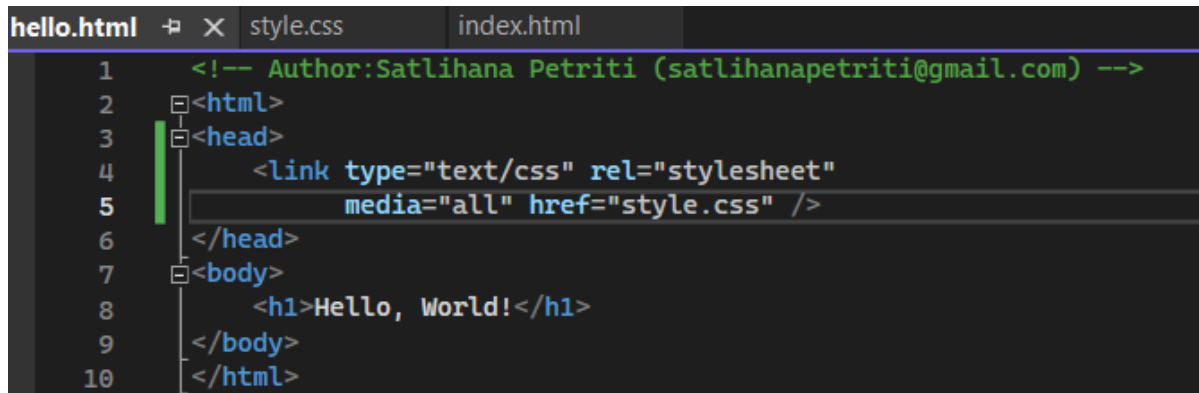
```
git commit -m "Added css stylesheet"
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello\lib>cd C:\Users\Acer\Downloads\ADM\Mm\hello
C:\Users\Acer\Downloads\ADM\Mm\hello>touch lib/style.css
C:\Users\Acer\Downloads\ADM\Mm\hello>git add lib/style.css
C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Added css stylesheet"
[style 5b54157] Added css stylesheet
1 file changed, 3 insertions(+)
create mode 100644 lib/style.css
```

рис.61 выполнение команды

1.16.3 Измените основную страницу

Обновите файл hello.html, чтобы использовать стили style.css



```
hello.html  X style.css  index.html
1  <!-- Author:Satlihana Petriti (satlihanapetriti@gmail.com) -->
2  <html>
3  <head>
4      <link type="text/css" rel="stylesheet"
5          media="all" href="style.css" />
6  </head>
7  <body>
8      <h1>Hello, World!</h1>
9  </body>
10 </html>
```

рис.62 Измените основную страницу

Выполните:

```
git add lib/hello.html
```

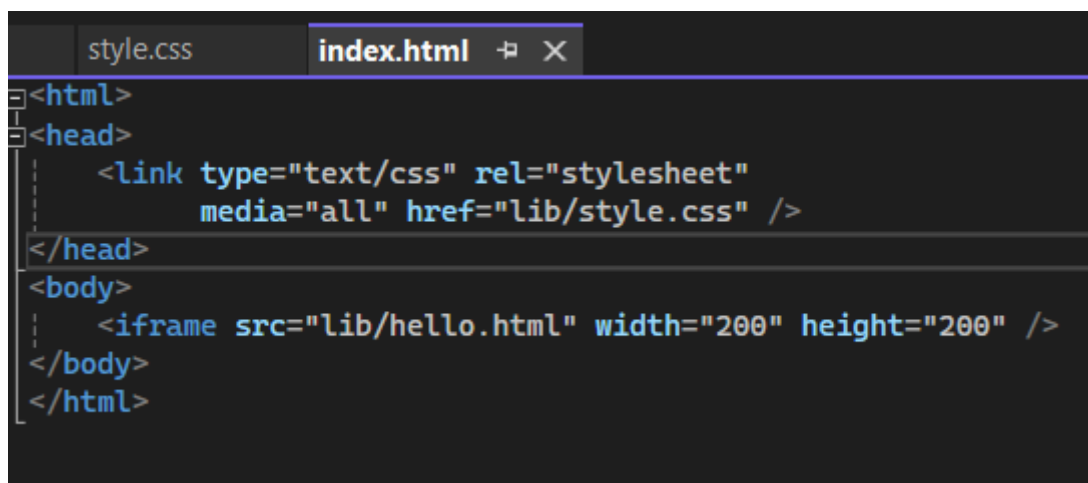
```
git commit -m "Hello uses style.css"
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git add lib/hello.html
C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Hello uses style.css"
[style 640cee4] Hello uses style.css
1 file changed, 1 insertion(+), 1 deletion(-)
```

рис.63 выполнение команды

1.16.4 Измените index.html

Обновите файл index.html, чтобы он тоже использовал style.css



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

рис.64 Измените index.html

Выполните:

```
git add index.html
```

```
git commit -m "Updated index.html"
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git add index.html

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Updated index.html"
[style 029d63b] Updated index.html
1 file changed, 1 insertion(+), 1 deletion(-)
```

рис.65 выполнение команды

17. 1.17 Навигация по веткам

Теперь в вашем проекте есть две ветки: Выполните:

```
git log --all
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git log --all
commit 029d63b47d672f2dc14090061c26f3c03be21lead (HEAD -> style)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:32:57 2024 +0100

    Updated index.html

commit 640cee4ab85cbff386fef3124952cdeb1d78d9c9
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:31:48 2024 +0100

    Hello uses style.css

commit 5b541570abc4f3eb958659dd9c1c4f5595988487
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:30:46 2024 +0100

    Added css stylesheet

commit 3e5fd727a5e2f58f4398e628471207cc16b52a5d (master)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 01:26:28 2024 +0100

    Added index.html.
```

рис.66 Навигация по веткам- есть две ветки

1.17.1 Переключение на ветку master

Используйте команду git checkout для переключения между ветками:

```
git checkout master
```

```
cat lib/hello.html
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout master
Switched to branch 'master'

C:\Users\Acer\Downloads\ADM\Mm\hello>cat lib/hello.html
</html>>Hello, World!</h1>iti (satlihanapetriti@gmail.com) -->
```

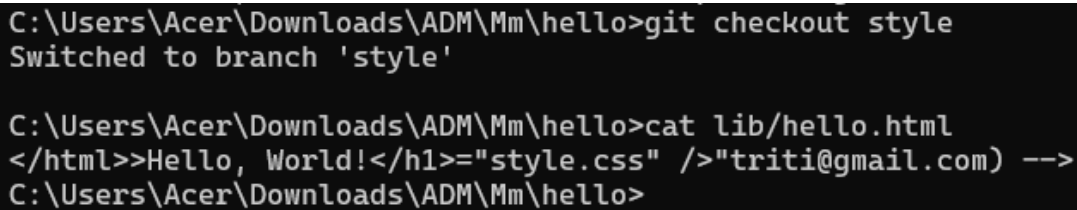
рис.67 переключения между ветками

Сейчас мы находимся на ветке master. Это заметно по тому, что файл hello.html не использует стили style.css.

1.17.2 Вернемся к ветке style

Выполните:

```
git checkout style
cat lib/hello.html
```



```
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout style
Switched to branch 'style'

C:\Users\Acer\Downloads\ADM\Mm\hello>cat lib/hello.html
</html>>Hello, World!</h1>="style.css" />"triti@gmail.com) -->
C:\Users\Acer\Downloads\ADM\Mm\hello>
```

рис.68 Вернемся к ветке style

Содержимое lib/hello.html подтверждает, что мы вернулись на ветку style.

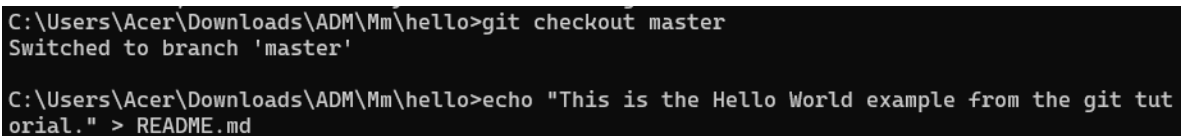
18. 1.18 Изменения в ветке master

Пока вы меняли ветку style, кто-то решил обновить ветку master. Они добавили файл README.md.

1.18.1 Создайте файл README в ветке master

Выполните:

```
git checkout master
Создайте файл README.md
echo "This is the hello world example from the git tutorial." > README.md
```



```
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout master
Switched to branch 'master'

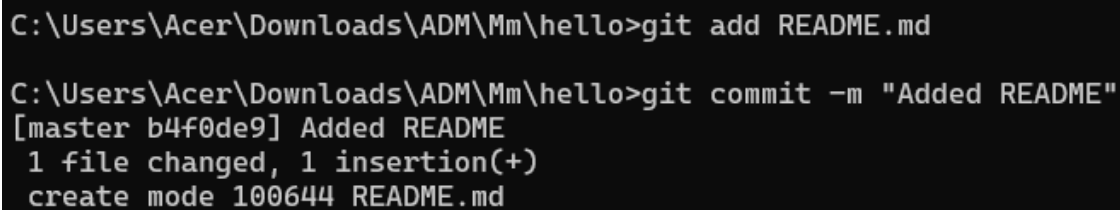
C:\Users\Acer\Downloads\ADM\Mm\hello>echo "This is the Hello World example from the git tut
orial." > README.md
```

рис.69 Изменения в ветке master

19. 1.19 Сделайте коммит изменений README.md в ветку master.

Выполните:

```
git add README.md
git commit -m "Added README"
```



```
C:\Users\Acer\Downloads\ADM\Mm\hello>git add README.md

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Added README"
[master b4f0de9] Added README
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

рис.70 коммит изменений README.md в ветку master

1.19.1 Просмотр отличающихся веток

1.19.2 Просмотрите текущие ветки

Теперь у нас в репозитории есть две отличающиеся ветки. Используйте следующую лог-команду для просмотра веток и их отличий. Выполните:

```
git log --graph --all
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>git log --graph --all
* commit b4f0de96ceb34c8679f93472fa342882ac8ad520 (HEAD -> master)
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:45:37 2024 +0100
|
|     Added README
|
| * commit 029d63b47d672f2dc14090061c26f3c03be21ead (style)
| | Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| | Date:   Fri Feb 9 02:32:57 2024 +0100
| |
| |     Updated index.html
| |
| * commit 640cee4ab85cbff386fef3124952cdeb1d78d9c9
| | Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| | Date:   Fri Feb 9 02:31:48 2024 +0100
| |
| |     Hello uses style.css
| |
| * commit 5b541570abc4f3eb958659dd9c1c4f5595988487
| / Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:30:46 2024 +0100
|
|     Added css stylesheet
|
| * commit 3e5fd727a5e2f58f4398e628471207cc16b52a5d
| | Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| | Date:   Fri Feb 9 01:26:28 2024 +0100
| |
| |     Added index.html.
| |
| * commit 38c53463db904dfd8a70a624ad97a74139d7cee7
| | Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| | Date:   Fri Feb 9 01:13:54 2024 +0100
| |
| |     Moved hello.html to lib
| |
| * commit 7dbd15b6f9395f4fc47278dd60fca66f5685fe4d
| | Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| | Date:   Fri Feb 9 01:02:11 2024 +0100
| |
| |     Add an author/email comment
| |
| * commit c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (tag: v1)
| | Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| | Date:   Thu Feb 8 23:19:05 2024 +0100
|

```

рис.71 просмотра веток и их отличий

Добавление опции --graph в git log вызывает построение дерева коммитов с помощью простых ASCII символов. Мы видим обе ветки (style и master), и то, что ветка master является текущей HEAD. Общим предшественником обеих веток является коммит «Added index.html». Опция --all гарантированно означает, что мы видим все ветки. По умолчанию показывается только текущая ветка.

20. 1.20 Слияние

1.20.1 Слияние веток

Слияние переносит изменения из двух веток в одну. Давайте вернемся к ветке style и сольем master с style.

Выполните:

```
git checkout style
```

```
git merge master
```

```
git log --graph --all
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout style
Switched to branch 'style'

C:\Users\Acer\Downloads\ADM\Mm\hello>git merge master
Merge made by the 'ort' strategy.
 README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

C:\Users\Acer\Downloads\ADM\Mm\hello>git log --graph --all
*   commit 47b63a89c7b3a7ef051ade6c8a6cc8bb646b862d (HEAD -> style)
|  \ Merge: 029d63b b4f0de9
|   Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 02:46:28 2024 +0100
|
|       Merge branch 'master' into style
|
| *   commit b4f0de96ceb34c8679f93472fa342882ac8ad520 (master)
|  \ Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 02:45:37 2024 +0100
|
|       Added README
|
| *   commit 029d63b47d672f2dc14090061c26f3c03be2lead
|  \ Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 02:32:57 2024 +0100
|
|       Updated index.html
|
| *   commit 640cee4ab85cbff386fef3124952cdeb1d78d9c9
|  \ Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 02:31:48 2024 +0100
|
|       Hello uses style.css
|
| *   commit 5b541570abc4f3eb958659dd9c1c4f5595988487
|  \ Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 02:30:46 2024 +0100
|
|       Added css stylesheet
|
| *   commit 3e5fd727a5e2f58f4398e628471207cc16b52a5d
|  \ Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 01:26:28 2024 +0100
|
|       Added index.html.
|
| *   commit 38c53463db904dfd8a70a624ad97a74139d7cee7
|  \ Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 01:13:54 2024 +0100
|
|       Moved hello.html to lib
|
| *   commit 7dbd15b6f9395f4fc47278dd60fca66f5685fe4d
|  \ Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 01:02:11 2024 +0100
```

рис.72 Слияние веток

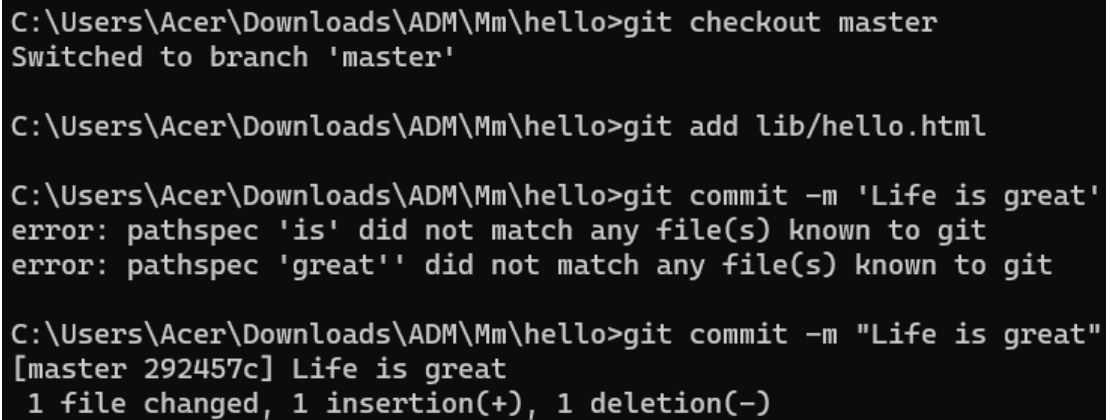
Путем периодического слияния ветки master с веткой style вы можете переносить из master любые изменения и поддерживать совместимость изменений style с изменениями в основной ветке.

21. 1.21 Создание конфликта

1.21.1 Вернитесь в master и создайте конфликт

Вернитесь в ветку master и внесите следующие изменения:

```
git checkout master
```



```
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout master
Switched to branch 'master'

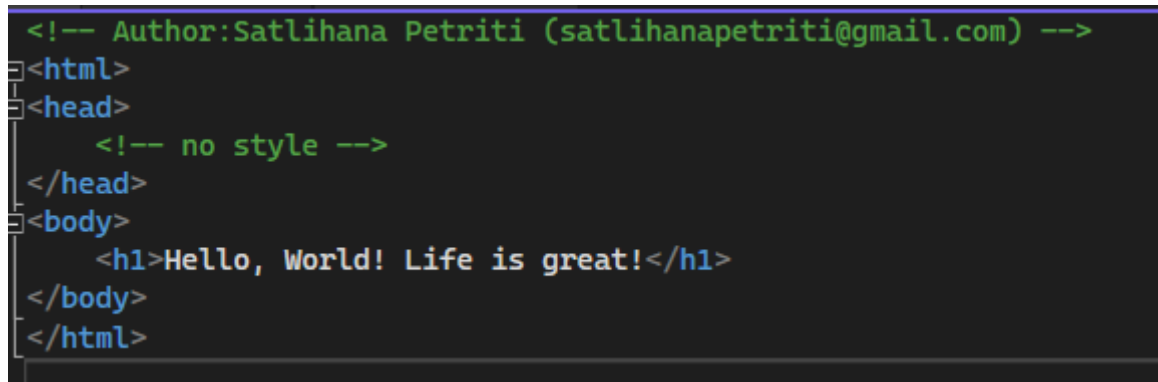
C:\Users\Acer\Downloads\ADM\Mm\hello>git add lib/hello.html

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m 'Life is great'
error: pathspec 'is' did not match any file(s) known to git
error: pathspec 'great' did not match any file(s) known to git

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Life is great"
[master 292457c] Life is great
1 file changed, 1 insertion(+), 1 deletion(-)
```

рис.73 Вернитесь в master+выполнение команды

Файл lib/hello.html



```
<!-- Author:Satlihana Petriti (satlihanapetriti@gmail.com) -->
<html>
<head>
  <!-- no style -->
</head>
<body>
  <h1>Hello, World! Life is great!</h1>
</body>
</html>
```

рис.74 Файл lib/hello.html

Выполните:

```
git add lib/hello.html
```

```
git commit -m 'Life is great'
```

1.21.2 Просмотр веток

Выполните:

```
git log --graph --all
```



```

C:\Users\Acer\Downloads\ADM\Mm\hello>git log --graph --all
* commit 292457c402777795ee156ca7c3832746bcb83391 (HEAD -> master)
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:54:18 2024 +0100
|
|     Life is great
|
| * commit 47b63a89c7b3a7ef051ade6c8a6cc8bb646b862d (style)
| \ Merge: 029d63b b4f0de9
| / Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| / Date:   Fri Feb 9 02:46:28 2024 +0100
|
|     Merge branch 'master' into style
|
| * commit b4f0de96ceb34c8679f93472fa342882ac8ad520
| | Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| | Date:   Fri Feb 9 02:45:37 2024 +0100
| |
| |     Added README
| |
| | * commit 029d63b47d672f2dc14090061c26f3c03be2lead
| | | Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| | | Date:   Fri Feb 9 02:32:57 2024 +0100
| | |
| | |     Updated index.html
| | |
| | | * commit 640cee4ab85cbff386fef3124952cdeb1d78d9c9
| | | | Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| | | | Date:   Fri Feb 9 02:31:48 2024 +0100
| | | |
| | | |     Hello uses style.css
| | | |
| | | | * commit 5b541570abc4f3eb958659dd9c1c4f5595988487
| | | | / Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| | | | Date:   Fri Feb 9 02:30:46 2024 +0100
| | | |
| | | |     Added css stylesheet
| | | |
| | | | * commit 3e5fd727a5e2f58f4398e628471207cc16b52a5d
| | | | | Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| | | | | Date:   Fri Feb 9 01:26:28 2024 +0100
| | | | |
| | | | |
|
C:\Users\Acer\Downloads\ADM\Mm\hello>

```

рис.75 Просмотры веток

После коммита «Added README» ветка master была объединена с веткой style, но в настоящее время в master есть дополнительный коммит, который не был слит с style.

22. 1.22 Разрешение конфликтов

1.22.1 Слияние master с веткой style

Теперь вернемся к ветке style и попытаемся объединить ее с новой веткой master.

Выполните:

```
git checkout style
```

```
git merge master
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout style
Switched to branch 'style'

C:\Users\Acer\Downloads\ADM\Mm\hello>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.
```

рис.76 Слияние master с веткой style

Если вы откроете lib/hello.html, вы увидите:

```
<<<<<<< HEAD
<!-- Author:Satlihana Petriti (satlihanapetriti@gmail.com) -->
<html>
<head>
  <link type="text/css" rel="stylesheet"
    media="all" href="style.css" />
</head>
<body>
  <h1>Hello, World!</h1>
</body>
</html>
=====
<!-- Author:Satlihana Petriti (satlihanapetriti@gmail.com) -->
<html>
<head>
  <!-- no style -->
</head>
<body>
  <h1>Hello, World! Life is great!</h1>
</body>
</html>
>>>>>> master
```

рис.77 результаты выполнения команды

Первый раздел — версия текущей ветки (style). Второй раздел — версия ветки master.

1.22.2 Решение конфликта

Вам необходимо вручную разрешить конфликт. Внесите изменения в lib/hello.html для достижения следующего результата.

```
<!-- Author:Satlihana Petriti (satlihanapetriti@gmail.com) -->
<html>
<head>
  <link type="text/css" rel="stylesheet"
    media="all" href="style.css" />
</head>
<body>
  <h1>Hello, World! Life is great!</h1>
</body>
</html>
```

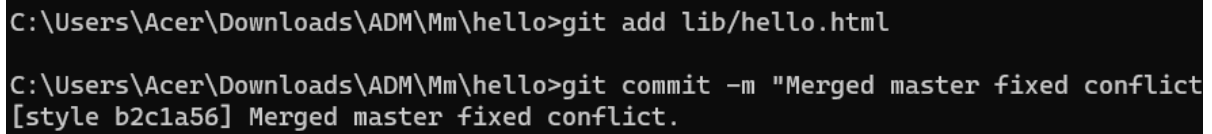
рис.78 Решение конфликта

1.22.3 Сделайте коммит решения конфликта

Выполните:

```
git add lib/hello.html
```

```
git commit -m "Merged master fixed conflict."
```



```
C:\Users\Acer\Downloads\ADM\Mm\hello>git add lib/hello.html  
  
C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Merged master fixed conflict  
[style b2c1a56] Merged master fixed conflict.
```

рис.79 коммит решения конфликта

23. 1.23 Сброс ветки style

1.23.1 Сброс ветки style

Вернемся на ветке style к точке перед тем, как мы слили ее с веткой master. Мы можем сбросить 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. Выполните:

```
git checkout style
```

```
git log --graph
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout style
Already on 'style'

C:\Users\Acer\Downloads\ADM\Mm\hello>git log --graph
*   commit b2c1a56b58af9b8c0ddc911e081557c65ae31e3c (HEAD -> style)
|  \ Merge: 47b63a8 292457c
|   Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 03:04:21 2024 +0100
|
|   Merged master fixed conflict.
|
| *   commit 292457c402777795ee156ca7c3832746bcb83391 (master)
|  \ Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 02:54:18 2024 +0100
|
|   Life is great
|
| *   commit 47b63a89c7b3a7ef051ade6c8a6cc8bb646b862d
|  \ Merge: 029d63b b4f0de9
|   Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 02:46:28 2024 +0100
|
|   Merge branch 'master' into style
|
| *   commit b4f0de96ceb34c8679f93472fa342882ac8ad520
|  \ Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 02:45:37 2024 +0100
|
|   Added README
|
| *   commit 029d63b47d672f2dc14090061c26f3c03be21ead
|  \ Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 02:32:57 2024 +0100
|
|   Updated index.html
|
| *   commit 640cee4ab85cbff386fef3124952cdeb1d78d9c9
|  \ Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
|   Date:   Fri Feb 9 02:31:48 2024 +0100
|
|   Hello uses style.css
|
| *   commit 5b541570abc4f3eb958659dd9c1c4f5595988487
|  \ Author: SatlihanaPetriti <satlihanapetriti@gmail.com>

```

рис.80 Сброс ветки style

Мы видим, что коммит «Updated index.html» был последним на ветке style перед слиянием. Давайте сбросим ветку style к этому коммиту.

Выполните:

```
git reset --hard <hash>
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>git reset --hard 029d63b47d672f2dc14090061c26f3c03be21ead
HEAD is now at 029d63b Updated index.html

```

рис.81 сбросим ветку style к этому коммиту

1.23.2 Проверьте ветку.

Поищите лог ветки style. У нас в истории больше нет коммитов слияний. Выполните:

```
git log --graph --all
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>git log --graph --all
* commit 292457c402777795ee156ca7c3832746bcb83391 (master)
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:54:18 2024 +0100
|
|     Life is great
|
* commit b4f0de96ceb34c8679f93472fa342882ac8ad520
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:45:37 2024 +0100
|
|     Added README
|
* commit 029d63b47d672f2dc14090061c26f3c03be21ead (HEAD -> style)
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:32:57 2024 +0100
|
|     Updated index.html
|
* commit 640cee4ab85cbff386fef3124952cdeb1d78d9c9
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:31:48 2024 +0100
|
|     Hello uses style.css
|
* commit 5b541570abc4f3eb958659dd9c1c4f5595988487
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:30:46 2024 +0100
|
|     Added css stylesheet
|
* commit 3e5fd727a5e2f58f4398e628471207cc16b52a5d
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 01:26:28 2024 +0100
|
|     Added index.html.
|
* commit 38c53463db904dfd8a70a624ad97a74139d7cee7
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 01:13:54 2024 +0100
|
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout master
Switched to branch 'master'

```

рис.82 Проверка ветки

24. 1.24 Сброс ветки master

1.24.1 Сброс ветки master

Добавив интерактивный режим в ветку master, мы внесли изменения, конфликтующие с изменениями в ветке style. Давайте вернемся в ветке master в точку перед внесением конфликтующих изменений. Это позволяет нам продемонстрировать работу команды git rebase, не беспокоясь о конфликтах.

Выполните:

```
git checkout master
```

```
git log --graph
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout master
Switched to branch 'master'

C:\Users\Acer\Downloads\ADM\Mm\hello>git log --graph
* commit 292457c402777795ee156ca7c3832746bcb83391 (HEAD -> master)
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:54:18 2024 +0100
|
|     Life is great
|
* commit b4f0de96ceb34c8679f93472fa342882ac8ad520
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:45:37 2024 +0100
|
|     Added README
|
* commit 3e5fd727a5e2f58f4398e628471207cc16b52a5d
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 01:26:28 2024 +0100
|
|     Added index.html.
|
* commit 38c53463db904dfd8a70a624ad97a74139d7cee7
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 01:13:54 2024 +0100
|
|     Moved hello.html to lib
|
* commit 7dbd15b6f9395f4fc47278dd60fca66f5685fe4d
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 01:02:11 2024 +0100
|
|     Add an author/email comment
|
* commit c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (tag: v1)
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Thu Feb 8 23:19:05 2024 +0100
|
|     Added HTML header
|
* commit 145e701e1d01c0e40e11f79e3f9c13284ca51c59 (tag: v1-beta)
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Thu Feb 8 23:18:13 2024 +0100
|
```

рис.83 Сброс ветки master

Коммит «Added README» идет непосредственно перед коммитом конфликтующего интерактивного режима. Мы сбросим ветку master к коммиту «Added README».

Выполните:

```
git reset --hard <hash>
```

```
git log --graph --all
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>git reset --hard b4f0de96ceb34c8679f93472fa342882ac8ad520
HEAD is now at b4f0de9 Added README

C:\Users\Acer\Downloads\ADM\Mm\hello>git log --graph --all
* commit b4f0de96ceb34c8679f93472fa342882ac8ad520 (HEAD -> master)
| Author: SatlihanaPetrity <satlihanapetrity@gmail.com>
| Date:   Fri Feb 9 02:45:37 2024 +0100
|
|     Added README
|
| * commit 029d63b47d672f2dc14090061c26f3c03be21ead (style)
| | Author: SatlihanaPetrity <satlihanapetrity@gmail.com>
| | Date:   Fri Feb 9 02:32:57 2024 +0100
| |
| |     Updated index.html
| |
| | * commit 640cee4ab85cbff386fef3124952cdeb1d78d9c9
| | | Author: SatlihanaPetrity <satlihanapetrity@gmail.com>
| | | Date:   Fri Feb 9 02:31:48 2024 +0100
| | |
| | |     Hello uses style.css
| | |
| | | * commit 5b541570abc4f3eb958659dd9c1c4f5595988487
| | | / Author: SatlihanaPetrity <satlihanapetrity@gmail.com>
| | | | Date:   Fri Feb 9 02:30:46 2024 +0100
| | | |
| | | |     Added css stylesheet
| | | |
| | | | * commit 3e5fd727a5e2f58f4398e628471207cc16b52a5d
| | | | | Author: SatlihanaPetrity <satlihanapetrity@gmail.com>
| | | | | Date:   Fri Feb 9 01:26:28 2024 +0100
| | | | |
| | | | |     Added index.html.
| | | | |
| | | | | * commit 38c53463db904dfd8a70a624ad97a74139d7cee7
| | | | | Author: SatlihanaPetrity <satlihanapetrity@gmail.com>
| | | | | Date:   Fri Feb 9 01:13:54 2024 +0100
| | | | |
| | | | |     Moved hello.html to lib
| | | | |
| | | | | * commit 7dbd15b6f9395f4fc47278dd60fca66f5685fe4d
| | | | | Author: SatlihanaPetrity <satlihanapetrity@gmail.com>
| | | | | Date:   Fri Feb 9 01:02:11 2024 +0100
| | | | |
| | | | |

```

рис.84 сбросим ветку master к коммиту «AddedREADME»

Посмотрите лог. Он должен выглядеть, как будто репозиторий был перемотан назад во времени к точке до какого-либо слияния.

25. 1.25 Перебазирование

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 использовать команду git rebase вместо слияния.

Выполните:

```
git checkout style
```

```
git rebase master
```

```
git log --graph
```



```

C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout style
Switched to branch 'style'

C:\Users\Acer\Downloads\ADM\Mm\hello>git rebase master
Successfully rebased and updated refs/heads/style.

C:\Users\Acer\Downloads\ADM\Mm\hello>git log --graph
* commit d8c8e9c41b0d11651fb9609dc28fe4e3ec1f83e7 (HEAD -> style)
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:32:57 2024 +0100
|
|     Updated index.html
|
* commit d0e5d7ebe2cce710554d6717f0ebe18a6c88b1a0
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:31:48 2024 +0100
|
|     Hello uses style.css
|
* commit 8c9e0cdd0a1d14c08e16a48ba55cc801fcf54980
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:30:46 2024 +0100
|
|     Added css stylesheet
|
* commit b4f0de96ceb34c8679f93472fa342882ac8ad520 (master)
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 02:45:37 2024 +0100
|
|     Added README
|
* commit 3e5fd727a5e2f58f4398e628471207cc16b52a5d
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 01:26:28 2024 +0100
|
|     Added index.html.
|
* commit 38c53463db904dfd8a70a624ad97a74139d7cee7
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 01:13:54 2024 +0100
|
|     Moved hello.html to lib
|
* commit 7dbd15b6f9395f4fc47278dd60fca66f5685fe4d
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Fri Feb 9 01:02:11 2024 +0100
|
|     Add an author/email comment
|
* commit c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (tag: v1)
| Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
| Date:   Thu Feb 8 23:19:05 2024 +0100
|
|     Added HTML header
|
* commit 145e701e1d01c0e40e11f79e3f9c13284ca51c59 (tag: v1-beta)

```

рис.85 Перебазирование

1.25.1 Слияние VS перебазирование

Конечный результат перебазирования очень похож на результат слияния. Ветка `style` в настоящее время содержит все свои изменения, а также все изменения ветки `master`. Однако, дерево коммитов значительно отличается. Дерево коммитов ветки `style` было переписано таким образом, что ветка `master` является частью истории коммитов. Это делает цепь коммитов линейной и гораздо более читабельной.

Не используйте перебазирование:

- если ветка является публичной и расшаренной, поскольку переписывание

общих веток будет мешать работе других членов команды;

- когда важна точная история коммитов ветки, так как команда rebase переписывает историю коммитов;

Учитывая приведенные выше рекомендации, рекомендуется использовать gitrebase для кратковременных, локальных веток, а слияние для веток в публичном репозитории.

26. 1.26 Слияние в ветку master

Мы поддерживали соответствие ветки style с веткой master (с помощью rebase), теперь давайте сольем изменения style в ветку master.

1.26.1 Слияние style в master

Выполните:

```
git checkout master
```

```
git merge style
```

Поскольку последний коммит ветки master прямо предшествует последнему коммиту ветки style, git может выполнить ускоренное слияние-перемотку. При быстрой перемотке вперед git просто передвигает указатель вперед, таким образом указывая на тот же коммит, что и ветка style.

При быстрой перемотке конфликтов быть не может.

1.26.2 Просмотрите логи

Выполните:

```
git log
```

Теперь ветки style и master идентичны.

```

C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout master
Switched to branch 'master'

C:\Users\Acer\Downloads\ADM\Mm\hello>git merge style
Updating b4f0de9..d8c8e9c
Fast-forward
 index.html      | 2 +-
 lib/hello.html  | 2 +-
 lib/style.css   | 3 +++
 3 files changed, 5 insertions(+), 2 deletions(-)
 create mode 100644 lib/style.css

C:\Users\Acer\Downloads\ADM\Mm\hello>git log
commit d8c8e9c41b0d11651fb9609dc28fe4e3ec1f83e7 (HEAD -> master, style)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:32:57 2024 +0100

    Updated index.html

commit d0e5d7ebe2cce710554d6717f0ebe18a6c88b1a0
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:31:48 2024 +0100

    Hello uses style.css

commit 8c9e0cdd0a1d14c08e16a48ba55cc801fcf54980
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:30:46 2024 +0100

    Added css stylesheet

commit b4f0de96ceb34c8679f93472fa342882ac8ad520
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:45:37 2024 +0100

    Added README

commit 3e5fd727a5e2f58f4398e628471207cc16b52a5d
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 01:26:28 2024 +0100

    Added index.html.

commit 38c53463db904dfd8a70a624ad97a74139d7cee7
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 01:13:54 2024 +0100

    Moved hello.html to lib

commit 7dbd15b6f9395f4fc47278dd60fca66f5685fe4d
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 01:02:11 2024 +0100

    Add an author/email comment

commit c22d10a2c98f6aac7bc0de674166cd3348f0eb85 (tag: v1)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>

```

рис.86 Слияние в ветку master

27. 1.27 Клонирование репозитория

1.27.1 Перейдите в рабочий каталог

Перейдите в рабочий каталог и сделайте клон вашего репозитория hello.Выполните:

```
cd ..
```

```
pwd
```

```
ls
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>cd C:\Users\Acer\Downloads\ADM\Mm
```

рис.87 Перейдите в рабочий каталог

Сейчас мы находимся в рабочем каталоге.

В этот момент вы должны находиться в «рабочем» каталоге. Здесь должен быть единственный репозиторий под названием «hello».

1.27.2 Создайте клон репозитория hello

Создадим клон репозитория. Выполните:

```
git clone hello cloned_hello
```

```
ls
```

```
C:\Users\Acer\Downloads\ADM\Mm>git clone hello cloned_hello
Cloning into 'cloned_hello'...
done.
```

```
C:\Users\Acer\Downloads\ADM\Mm>ls
cloned_hello  hello
```

рис.88 Создайте клон репозитория hello

В вашем рабочем каталоге теперь должно быть два репозитория: оригинальный репозиторий «hello» и клонированный репозиторий «cloned_hello»

28. 1.28 Просмотр клонированного репозитория

1.28.1 Давайте взглянем на клонированный репозиторий.

Выполните:

```
cd cloned_hello
```

```
ls
```

```
C:\Users\Acer\Downloads\ADM\Mm>cd cloned_hello
```

```
C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>ls
README.md  index.html  lib
```

рис.89 взглянем на клонированный репозиторий

Вы увидите список всех файлов на верхнем уровне оригинального репозитория README.md, index.html и lib.

1.28.2 Просмотрите историю репозитория

Выполните:

```
git log --all
```

```

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git log --all
commit d8c8e9c41b0d11651fb9609dc28fe4e3ec1f83e7 (HEAD -> master, origin/style, origin/master, origin/HEAD)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:32:57 2024 +0100

    Updated index.html

commit d0e5d7ebe2cce710554d6717f0ebe18a6c88b1a0
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:31:48 2024 +0100

    Hello uses style.css

commit 8c9e0cdd0a1d14c08e16a48ba55cc801fcf54980
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:30:46 2024 +0100

    Added css stylesheet

commit b4f0de96ceb34c8679f93472fa342882ac8ad520
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:45:37 2024 +0100

    Added README

commit 3e5fd727a5e2f58f4398e628471207cc16b52a5d
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 01:26:28 2024 +0100

    Added index.html.

commit 38c53463db904dfd8a70a624ad97a74139d7cee7
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 01:13:54 2024 +0100

    Moved hello.html to lib

commit 7dbd15b6f9395f4fc47278dd60fca66f5685fe4d
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 01:02:11 2024 +0100

```

рис.90 история репозитории

Вы увидите список всех коммитов в новый репозиторий, и он должен (более или менее) совпадать с историей коммитов в оригинальном репозитории. Единственная разница должна быть в названиях веток.

Вы увидите ветку master (HEAD) в списке истории. Вы также увидите ветки со странными именами (origin/master, origin/style и origin/HEAD).

29. 1.29 Что такое origin?

Выполните:

```
git remote
```

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

Выполните:

```
git remote show origin
```

```

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git remote
origin

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git remote show origin
* remote origin
  Fetch URL: C:/Users/Acer/Downloads/ADM/Mm/hello
  Push URL: C:/Users/Acer/Downloads/ADM/Mm/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)

```

рис.91 получить более подробную информацию об имени по умолчанию

Удаленные репозитории обычно размещаются на отдельной машине, возможно, централизованном сервере. Однако, как мы видим здесь, они могут с тем же успехом указывать на репозиторий на той же машине. Нет ничего особенного в имени «origin», однако существует традиция использовать «origin» в качестве имени первичного централизованного репозитория (если таковой имеется).

30. 1.30 Удаленные ветки

Давайте посмотрим на ветки, доступные в нашем клонированном репозитории. Выполните:

```
git branch
```

Как мы видим, в списке только ветка master. Команда gitbranch выводит только список локальных веток по умолчанию.

```

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git branch
* master

```

рис.92 Удаленные ветки

1.30.1 Список удаленных веток

Для того, чтобы увидеть все ветки, попробуйте следующую команду:

```
git branch -a
```

```

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/style

```

рис.93 Список удаленных веток

Git выводит все коммиты в оригинальный репозиторий, но ветки в удаленном репозитории не рассматриваются как локальные. Если мы хотим собственную ветку style, мы должны сами ее создать. Через минуту вы увидите, как это делается.

31. 1.31 Изменение оригинального репозитория

Внесите некоторые изменения в оригинальный репозиторий, чтобы затем попытаться извлечь и слить изменения из удаленной ветки в текущую

1.31.1 Внесите изменения в оригинальный репозиторий hello

Выполните:

```
cd ../hello
```

```
C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>cd ../hello

C:\Users\Acer\Downloads\ADM\Mm\hello>git add README
fatal: pathspec 'README' did not match any files

C:\Users\Acer\Downloads\ADM\Mm\hello>ls
README.md index.html lib

C:\Users\Acer\Downloads\ADM\Mm\hello>git add README.md

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Changed README in original repo"
[master 003adeb] Changed README in original repo
1 file changed, 1 insertion(+), 1 deletion(-)
```

рис.94 Внесите изменения в оригинальный репозиторий hello

Примечание: Сейчас мы находимся в репозитории hello. Внесите следующие изменения в файл README.md: Файл README.md

```
This is the hello world example from the git tutorial.
```

Теперь добавьте это изменение и сделайте коммит. Выполните:

```
git add README
```

```
git commit -m "Changed README in original repo"
```

Теперь в оригинальном репозитории есть более поздние изменения, которых нет в клонированной версии. Далее мы извлечем и сольем эти изменения в клонированный репозиторий.

1.31.2 Извлечение изменений

Научиться извлекать изменения из удаленного репозитория. Выполните:

```
cd ../cloned_hello
```

```
git fetch
```

```
git log --all
```

```

C:\Users\Acer\Downloads\ADM\Mm\hello>cd ../cloned_hello

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>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), 361 bytes | 16.00 KiB/s, done.
From C:/Users/Acer/Downloads/ADM/Mm/hello
    d8c8e9c..003adeb  master    -> origin/master

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git log --all
commit 003adeb8f335c6fdff17f129d3561d12bf922926 (origin/master, origin/HEAD)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 03:33:20 2024 +0100

    Changed README in original repo

commit d8c8e9c41b0d11651fb9609dc28fe4e3ec1f83e7 (HEAD -> master, origin/style)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:32:57 2024 +0100

    Updated index.html

commit d0e5d7ebe2cce710554d6717f0ebe18a6c88b1a0
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:31:48 2024 +0100

    Hello uses style.css

commit 8c9e0cdd0a1d14c08e16a48ba55cc801fcf54980
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:30:46 2024 +0100

    Added css stylesheet

commit b4f0de96ceb34c8679f93472fa342882ac8ad520
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:45:37 2024 +0100

    Added README

commit 3e5fd727a5e2f58f4398e628471207cc16b52a5d
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 01:26:28 2024 +0100

    Added index.html.

```

рис.95 Извлечение изменений

1.31.3 Проверьте README.md

Мы можем продемонстрировать, что клонированный файл README.md не изменился.

Выполните:

```
cat README
```

```

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>cat README.md
"This is the Hello World example from the git tutorial."

```

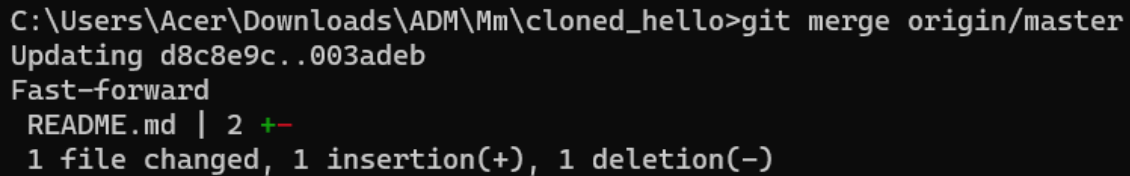
рис.96 Проверка README.md

32. 1.32 Слияние извлеченных изменений

1.32.1 Слейте извлеченные изменения в локальную ветку master

Выполните:

```
git merge origin/master
```



```
C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git merge origin/master
Updating d8c8e9c..003adeb
Fast-forward
 README.md | 2 +--
 1 file changed, 1 insertion(+), 1 deletion(-)
```

рис.97 Слейте извлеченные изменения в локальную ветку master

1.32.2 Еще раз проверьте файл README.md

Сейчас мы должны увидеть изменения.Выполните:

```
cat README.md
```

Хотя команда `git fetch` не сливает изменения, мы можем вручную слить изменения из удаленного репозитория.

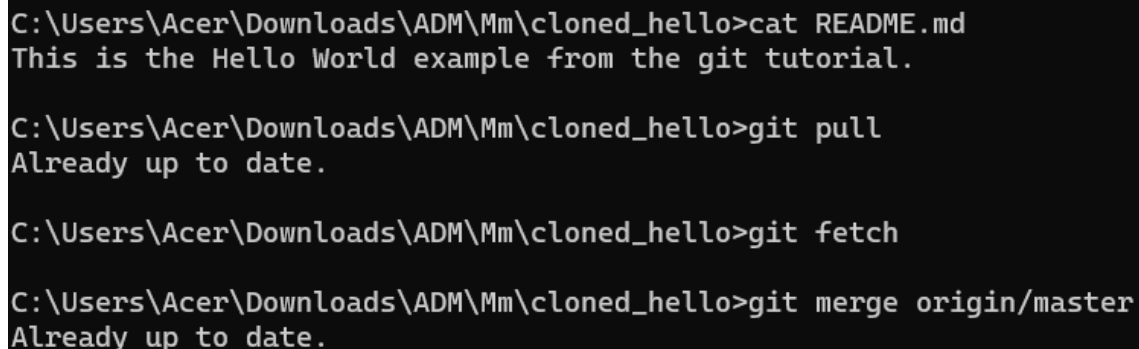
Теперь давайте рассмотрим объединение `fetch` и `merge` в одну команду. Выполнение:

```
git pull
```

эквивалентно двум следующим шагам:

```
git fetch
```

```
git merge origin/master
```



```
C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>cat README.md
This is the Hello World example from the git tutorial.

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git pull
Already up to date.

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git fetch

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git merge origin/master
Already up to date.
```

рис.98 Проверка README.md

33. 1.33 Добавление ветки наблюдения

Ветки, которые начинаются с `remotes/origin` являются ветками оригинального репозитория.

Обратите внимание, что у вас больше нет ветки под названием `style`, но система контроля версий знает, что в оригинальном репозитории ветка `style` была.

1.33.1 Добавьте локальную ветку, которая отслеживает удаленную ветку

Выполните:

```
git branch --track style origin/style
```

```
git branch -a
```

```
git log --max-count=2
```



```

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git branch --track style origin/style
branch 'style' set up to track 'origin/style'.

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git branch -a
* master
  style
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/style

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git log --max-count=2
commit 003adeb8f335c6fdff17f129d3561d12bf922926 (HEAD -> master, origin/master, ori
gin/HEAD)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 03:33:20 2024 +0100

    Changed README in original repo

commit d8c8e9c41b0d11651fb9609dc28fe4e3ec1f83e7 (origin/style, style)
Author: SatlihanaPetriti <satlihanapetriti@gmail.com>
Date:   Fri Feb 9 02:32:57 2024 +0100

    Updated index.html

```

рис.99 Добавьте локальную ветку, которая отслеживает удаленную ветку

Теперь мы можем видеть ветку style в списке веток и логе

34. 1.34 Чистые репозитории

Чистые репозитории (без рабочих каталогов) обычно используются для расшаривания. Обычный git-репозиторий подразумевает, что вы будете использовать его как рабочую директорию, поэтому вместе с файлами проекта в актуальной версии, git хранит все служебные, «чисто-репозиториевские» файлы в поддиректории .git. В удаленных репозиториях нет смысла хранить рабочие файлы на диске (как это делается в рабочих копиях), а все что им действительно нужно — это дельты изменений и другие бинарные данные репозитория. Вот это и есть «чистый репозиторий».

35. 1.35 Создайте чистый репозиторий

```

cd ..
git clone --bare hello hello.git
ls hello.git

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>cd C:\Users\Acer\Downloads\ADM\Mm
C:\Users\Acer\Downloads\ADM\Mm>ls
cloned_hello  hello

C:\Users\Acer\Downloads\ADM\Mm>git clone --bare hello hello.git
Cloning into bare repository 'hello.git'...
done.

C:\Users\Acer\Downloads\ADM\Mm>ls hello.git
HEAD config description hooks info objects packed-refs refs

```

рис.100 Создам чистый репозиторий

Сейчас мы находимся в рабочем каталоге

Как правило, репозитории, оканчивающиеся на .git являются чистыми репозиториями. Мы видим, что в репозитории hello.git нет рабочего каталога. По сути, это есть не что иное, как каталог .git нечистого репозитория.

36. 1.36 Добавление удаленного репозитория

Давайте добавим репозиторий hello.git к нашему оригинальному репозиторию.

```
cd hello
git remote add shared ../hello.git
```

```
C:\Users\Acer\Downloads\ADM\Mm>cd hello
C:\Users\Acer\Downloads\ADM\Mm\hello>git remote add shared ../hello.git
```

рис.101 Добавление удаленного репозитория

37. 1.37 Отправка изменений

Так как чистые репозитории, как правило, расшариваются на каком-нибудь сетевом сервере, нам необходимо отправить наши изменения в другие репозитории. Начнем с создания изменения для отправки. Отредактируйте файл README.md и сделайте коммит

```
файл README.md:
This is the Hello world example from the git tutorial.
(Changed in the original and pushed to shared)
```

Выполните:

```
git checkout master
git add README
git commit -m "Added shared comment to readme"
```

Теперь отправьте изменения в общий репозиторий. Выполните:

```
git push shared master
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>git checkout master
Already on 'master'
M      README.md

C:\Users\Acer\Downloads\ADM\Mm\hello>git add README
fatal: pathspec 'README' did not match any files

C:\Users\Acer\Downloads\ADM\Mm\hello>git add README.md

C:\Users\Acer\Downloads\ADM\Mm\hello>git commit -m "Added shared comment to readme"

[master a9ce16e] Added shared comment to readme
1 file changed, 1 insertion(+)

C:\Users\Acer\Downloads\ADM\Mm\hello>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), 411 bytes | 58.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To ../hello.git
003adeb..a9ce16e  master -> master
```

рис.102 Отправка изменений

Общим называется репозиторий, получающий отправленные нами изменения.

38. 1.38 Извлечение общих изменений

Научиться извлекать изменения из общего репозитория. Быстро переключитесь в клонированный репозиторий и извлеките изменения, только что отправленные в общий репозиторий. Выполните:

```
cd ../cloned_hello
```

Сейчас мы находимся в репозитории cloned_hello. Выполните:

```
git remote add shared ../hello.git
git branch --track shared master
git pull shared master
cat README.md
```

```
C:\Users\Acer\Downloads\ADM\Mm\hello>cd ../cloned_hello

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git remote add shared ../hello.git

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>git branch --track shared master
branch 'shared' set up to track 'master'.

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>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), 391 bytes | 10.00 KiB/s, done.
From ../hello
 * branch                master      -> FETCH_HEAD
 * [new branch]          master      -> shared/master
Updating 003adeb..a9ce16e
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)

C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>cat README.md
This is the Hello World example from the git tutorial.
(Changed in the original and pushed to shared)
C:\Users\Acer\Downloads\ADM\Mm\cloned_hello>
```

рис.102 Извлечение общих изменений

Вывод

Я научилась работать с Git и познакомилась с его инструментами и утилитами. Это включает в себя развитие практического понимания рабочего процесса Git для эффективного контроля версий при разработке программного обеспечения.