

Отчёт по лабораторной работе №1

Игнатенкова Варвара Николаевна

Содержание

1	Цель работы	6
2	Теоретическое введение	7
3	Выполнение лабораторной работы	8
3.1	Подготовка	8
3.2	Создание проекта	8
3.3	Внесение изменений	9
3.4	Индексация изменений	9
3.5	Отмена локальных изменений (до индексации)	16
3.6	Отмена проиндексированных изменений (перед коммитом)	17
3.7	Отмена коммитов	19
3.8	Удаление коммитов из ветки	20
3.9	Удаление тега оорс	22
3.10	Внесение изменений в коммиты	22
3.11	Перемещение файлов	24
3.12	Второй способ перемещения файлов	24
3.13	Подробнее о структуре	24
3.14	Git внутри: Каталог.git	25
3.15	Работа непосредственно с объектами git	27
3.16	Создание ветки	28
3.17	Навигация по веткам	29
3.18	Изменения в ветке master	31
3.19	Сделаем коммит изменений README.md в ветку master.	31
3.20	Слияние	32
3.21	Создание конфликта	33
3.22	Разрешение конфликтов	34
3.23	Сброс ветки style	35
3.24	Сброс ветки master	37
3.25	Перебазирование	38
3.26	Слияние в ветку master	40
3.27	Клонирование репозитория	41
3.28	Просмотр клонированного репозитория	42
3.29	Что такое origin?	42
3.30	Удаленные ветки	43
3.31	Изменение оригинального репозитория	44
3.32	Слияние извлеченных изменений	44

3.33 Добавление ветки наблюдения	45
3.34 Чистые репозитории	45
3.35 Создадим чистый репозиторий	46
3.36 Добавление удаленного репозитория	46
3.37 Отправка изменений	46
3.38 Извлечение общих изменений	47
4 Выводы	48

List of Figures

3.1	26	14
3.2	64	27

List of Tables

1 Цель работы

Знакомство с git и изучение его работы.

2 Теоретическое введение

Git — распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года; координатор — Дзюн Хамано.

3 Выполнение лабораторной работы

3.1 Подготовка

Установка имени и электронной почты Установим имя и электронную почту для установки git

```
vnignatenkova@matmodd:~$ git config --global user.name 'vnignatenkovarudn'
vnignatenkova@matmodd:~$ git config --global user.email '1132226497@pfur.ru'
```

Параметры установки окончаний строк Настроим core.autocrlf с параметрами true и input для перевода строк текстовых файлов в главном репозитории в одинаковый формат.

```
vnignatenkova@matmodd:~$ git config --global core.autocrlf input
vnignatenkova@matmodd:~$ git config --global core.safecrlf true
```

Установка отображения unicode Установим отображение unicode.

```
vnignatenkova@matmodd:~$ git config --global core.quotePath off
```

3.2 Создание проекта

Создание страницы «Hello,World» Создадим пустой каталог с именем hello и создадим файл hello.html.

```
vnignatenkova@matmodd:~$ mkdir hello
vnignatenkova@matmodd:~$ cd hello/
vnignatenkova@matmodd:~/hello$ touch hello.html
vnignatenkova@matmodd:~/hello$ echo 'Hello, World!' > hello.html
```

Создание репозитория Создадим git репозиторий с помощью команды git init.

```
vnignatenkova@matmodd:~/hello$ git init
Initialized empty Git repository in /home/vnignatenkova/hello/.git/
```

Добавление файла в репозиторий Добавим файл в репозиторий.


```
vnignatenkova@matmodd:~/hello$ git add hello.html
vnignatenkova@matmodd:~/hello$ git commit -m 'Initial Commit'
[master (root-commit) 82393ec] Initial Commit
1 file changed, 1 insertion(+)
create mode 100644 hello.html
```

Проверка состояние репозитория Проверим состояние репозитория.

```
vnignatenkova@matmodd:~/hello$ git status
On branch master
nothing to commit, working tree clean
```

Команда проверки состояния сообщит, что коммитить нечего.

3.3 Внесение изменений

Изменим страницу «Hello,World» Добавим HTML-теги в файле hello.html.

```
GNU nano 4.8
<h1>Hello, World!</h1>
```

Проверим состояние рабочего каталога.

```
vnignatenkova@matmodd:~/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")
vnignatenkova@matmodd:~/hello$
```

Git знает, что файл hello.html был изменен,но при этом эти изменения еще не зафиксированы в репозитории.

3.4 Индексация изменений

Коммит изменений Проиндексируем изменения командой git и проверим состояние.

```

vlnignatenkova@natmodd:~/hello$ git add hello.html
vlnignatenkova@natmodd:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html

```

Изменения файла hello.html были проиндексированы. Это означает, что gitте перь знает об изменении, но изменение пока не записано в репозиторий. Следующий коммит будет включать в себя проиндексированные изменения.

Сделаем коммит без метки -m.

После открытия первой строки введем комментарий, сохраним файл и выйдем из редактора. Еще раз проверим состояние.

```

vlnignatenkova@natmodd:~/hello$ git commit
[master aaa292e] «Added h1 tag»
 1 file changed, 1 insertion(+), 1 deletion(-)
vlnignatenkova@natmodd:~/hello$ git status
On branch master
nothing to commit, working tree clean

```

Рабочий каталог чистый, продолжим работу.

Добавим стандартные теги страницы Измените страницу «Hello,World», чтобы она содержала стандартные теги html и body.

```

GNU nano 4.8
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```

Теперь добавим это изменение в индекс git.

```

vlnignatenkova@natmodd:~/hello$ git add hello.html
vlnignatenkova@natmodd:~/hello$

```

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

```
GNU nano 4.8
<html>
<head>
</head>
<body>
  <h1>Hello, World!</h1>
</body>
</html>
```

Проверим текущий статус.

```
vnignatenkova@natnodd:~/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

vnignatenkova@natnodd:~/hello$
```

Hello.html указан дважды в состоянии. Первое изменение (добавление стандартных тегов) проиндексировано и готово к коммиту. Второе изменение (добавление заголовков HTML) является непроиндексированным.

Произведем коммит проиндексированного изменения, а затем еще раз проверим состояние.

```
vnignatenkova@natnodd:~/hello$ git commit -m "Added standard HTML page tags"
[master 19a2de4] Added standard HTML page tags
1 file changed, 5 insertions(+), 1 deletion(-)
vnignatenkova@natnodd:~/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")
vnignatenkova@natnodd:~/hello$
```

Состояние команды говорит о том, что hello.html имеет незафиксированные изменения, но уже не в буферной зоне.

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

```

vnignatenkova@matmodd:~/hello$ git add .
vnignatenkova@matmodd:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

```

Сделаем коммит второго изменения.

```

vnignatenkova@matmodd:~/hello$ git commit -m "Added HTML header"
[master 24363a8] Added HTML header
 1 file changed, 2 insertions(+)
vnignatenkova@matmodd:~/hello$ git status
On branch master
nothing to commit, working tree clean

```

История

Получим список произведенных изменений.

```

vnignatenkova@matmodd:~/hello$ git log
commit 24363a89f6b3544c55bfd014bdce4b5e606168e1 (HEAD -> master)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:21:41 2025 +0300

    Added HTML header

commit 19a2de4ba0c1352570762e0e604f5d676308700d
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:20:54 2025 +0300

    Added standard HTML page tags

commit aaa292e91a2263a5aa4a9bd2f9befd896f52f22b
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:16:04 2025 +0300

    «Added h1 tag»

commit bbfb139cf8bdae5069fc729a6540f3007e909de2
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:11:23 2025 +0300

```

Однострочный формат истории:

```

vnignatenkova@matmodd:~/hello$ git log --pretty=oneline
24363a89f6b3544c55bfd014bdce4b5e606168e1 (HEAD -> master) Added HTML header
19a2de4ba0c1352570762e0e604f5d676308700d Added standard HTML page tags
aaa292e91a2263a5aa4a9bd2f9befd896f52f22b «Added h1 tag»
bbfb139cf8bdae5069fc729a6540f3007e909de2 Commit 1.3.2
573d29a2fb1b5f16f348a209d6e24b6edfe4d228 Commit 1.3
82393ec3e5ccbc89dcea65a3949ea82707d8bb11 Initial Commit

```

Есть много вариантов отображения лога.

```

vnignatenkova@matnodd:~/hello$ git log --pretty=oneline
24363a89f6b3544c55bfd014bdce4b5e606168e1 (HEAD -> master) Added HTML header
19a2de4ba0c1352570762e0e604f5d676308700d Added standard HTML page tags
aaa292e91a2263a5aa4a9bd2f9befd896f52f22b «Added h1 tag»
bbf8139cf8bdae5069fc729a6540f3007e909de2 Commit 1.3.2
573d29a2fb1b5f16f348a209d6e24b6edfe4d228 Commit 1.3
82393ec3e5ccbc89dcea65a3949ea82707d8bb11 Initial Commit
vnignatenkova@matnodd:~/hello$ git log --pretty=oneline --max-count=2
24363a89f6b3544c55bfd014bdce4b5e606168e1 (HEAD -> master) Added HTML header
19a2de4ba0c1352570762e0e604f5d676308700d Added standard HTML page tags
vnignatenkova@matnodd:~/hello$ git log --pretty=oneline --since='5 minutes ago'
vnignatenkova@matnodd:~/hello$ git log --pretty=oneline --since='10 minutes ago'
24363a89f6b3544c55bfd014bdce4b5e606168e1 (HEAD -> master) Added HTML header
19a2de4ba0c1352570762e0e604f5d676308700d Added standard HTML page tags
vnignatenkova@matnodd:~/hello$ git log --pretty=oneline --until='5 minutes ago'
24363a89f6b3544c55bfd014bdce4b5e606168e1 (HEAD -> master) Added HTML header
19a2de4ba0c1352570762e0e604f5d676308700d Added standard HTML page tags
aaa292e91a2263a5aa4a9bd2f9befd896f52f22b «Added h1 tag»
bbf8139cf8bdae5069fc729a6540f3007e909de2 Commit 1.3.2
573d29a2fb1b5f16f348a209d6e24b6edfe4d228 Commit 1.3
82393ec3e5ccbc89dcea65a3949ea82707d8bb11 Initial Commit
vnignatenkova@matnodd:~/hello$ git log --pretty=oneline --until='10 minutes ago'
aaa292e91a2263a5aa4a9bd2f9befd896f52f22b «Added h1 tag»
bbf8139cf8bdae5069fc729a6540f3007e909de2 Commit 1.3.2
573d29a2fb1b5f16f348a209d6e24b6edfe4d228 Commit 1.3
82393ec3e5ccbc89dcea65a3949ea82707d8bb11 Initial Commit

```

Справочная страница:

Получение старых версий

Получим хэши предыдущих версий.

```

commit 24363a89f6b3544c55bfd014bdce4b5e606168e1 (HEAD -> master)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:21:41 2025 +0300

    Added HTML header

commit 19a2de4ba0c1352570762e0e604f5d676308700d
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:20:54 2025 +0300

    Added standard HTML page tags

commit aaa292e91a2263a5aa4a9bd2f9befd896f52f22b
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:16:04 2025 +0300

    «Added h1 tag»

commit bbf8139cf8bdae5069fc729a6540f3007e909de2
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:11:23 2025 +0300

    Commit 1.3.2

commit 573d29a2fb1b5f16f348a209d6e24b6edfe4d228
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:09:44 2025 +0300

    Commit 1.3

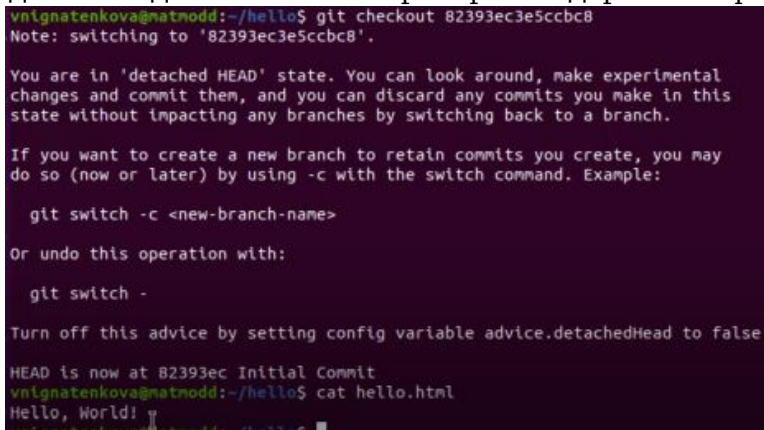
commit 82393ec3e5ccbc89dcea65a3949ea82707d8bb11
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:08:13 2025 +0300

    :

```

Изучим данные лога и найдем хэш для первого коммита. Используем этот хэш-

код в команде ниже. Затем проверим содержимое файла hello.html.

A terminal window with a dark purple background. The text is as follows:

```
vnignatenkova@natnodd:~/hello$ git checkout 82393ec3e5ccbc8
Note: switching to '82393ec3e5ccbc8'.

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 82393ec Initial Commit
vnignatenkova@natnodd:~/hello$ cat hello.html
Hello, World!
```

Вернемся к последней версии в ветке master.

Создание тегов версий Давайте назовем текущую версию страницы hello первой (v1). Создадим тег первой версии.

26

Figure 3.1: 26

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».


```

vnignatenkova@matmodd:~/hello$ git tag v1
vnignatenkova@matmodd:~/hello$ git checckout v1^
git: 'checckout' is not a git command. See 'git --h

The most similar command is
  checkout
vnignatenkova@matmodd:~/hello$ 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 19a2de4 Added standard HTML page tags
vnignatenkova@matmodd:~/hello$ cat hello.html
<html>
<body>
  <h1>Hello, World!</h1>
</body>
</html>
vnignatenkova@matmodd:~/hello$

```

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

```

vnignatenkova@matmodd:~/hello$ git tag v1-beta
vnignatenkova@matmodd:~/hello$ git checkout v1

```

Переключение по имени тега Теперь попробуем попереключаться между двумя отмеченными версиями.

```

vnignatenkovagmatmodd:~/hello$ git checkout v1
Previous HEAD position was 19a2de4 Added standard HTML page tags
HEAD is now at 24363a8 Added HTML header
vnignatenkovagmatmodd:~/hello$ git checkout v1-beta
Previous HEAD position was 24363a8 Added HTML header
HEAD is now at 19a2de4 Added standard HTML page tags

```

Просмотр тегов с помощью команды tag Мы можем еувидеть,какие теги доступны, используя команду git tag.

```

vnignatenkovagmatmodd:~/hello$ git tag
v1
v1-beta

```

Мы так же можем посмотреть теги в логе.

```

vnignatenkova@matmodd:~/hello$ git log master --all
commit 24363a89f6b3544c55bfd014bdce4b5e606168e1 (tag: v1.3.2)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:21:41 2025 +0300

    Added HTML header

commit 19a2de4ba0c1352570762e0e604f5d676308700d (HEAD, tag: v1-beta)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:20:54 2025 +0300

    Added standard HTML page tags

commit aaa292e91a2263a5aa4a9bd2f9befd896f52f22b
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:16:04 2025 +0300

    «Added h1 tag»

commit bbfb8139cf8bdae5069fc729a6540f3007e909de2
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:11:23 2025 +0300

    Commit 1.3.2

commit 573d29a2fb1b5f16f348a209d6e24b6edfe4d228
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:09:44 2025 +0300

    Commit 1.3

```

Мы можем видеть теги (v1 и v1-beta) в логе вместе с именем ветки (master). Кроме того HEAD показывает коммит, на который переключились (на данный момент это v1-beta).

3.5 Отмена локальных изменений (до индексации)

Переключимся на ветку master Убедимся, что мы находимся на последнем коммите ветки master, прежде чем продолжить работу.

```

vnignatenkova@matmodd:~/hello$ git checkout master
Previous HEAD position was 19a2de4 Added standard HTML page tags
Switched to branch 'master'

```

Изменим hello.html Внесем изменение в файл hello.html в виде нежелательного комментария.


```
GNU nano 4.8
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
<!-- This is a bad comment. We want to revert it. -->
</body>
</html>
```

Проверим состояние. Сначала проверим состояние рабочего каталога.

```
vnignatenkova@matmod:~/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")
```

Мы видим, что файл hello.html был изменен, но еще не проиндексирован. Отмена изменений в рабочем каталоге. Используем команду git checkout для переключения версии файла hello.html в репозитории.

```
vnignatenkova@matmod:~/hello$ git checkout hello.html
Updated 1 path from the index
vnignatenkova@matmod:~/hello$ git status
On branch master
nothing to commit, working tree clean
vnignatenkova@matmod:~/hello$ cat hello.html
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

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

3.6 Отмена проиндексированных изменений (перед коммитом)

Изменим файл и проиндексируем изменения. Внесем 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>

```

Проиндексируем это изменение. Проверим состояние. Проверим состояние нежелательного изменения.

```

vnignatenkova@natnodd:~/hello$ git add hello.html
vnignatenkova@natnodd:~/hello$ git statis
git: 'statis' is not a git command. See 'git --help'.

The most similar command is
    status
vnignatenkova@natnodd:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

```

Состояние показывает, что изменение было проиндексировано и готово к коммиту. Выполним сброс буферной зоны. Выполним отмену индексации изменения.

```

vnignatenkova@natnodd:~/hello$ git reset HEAD hello.html
Unstaged changes after reset:
M       hello.html
vnignatenkova@natnodd:~/hello$

```

Команда `git reset` сбрасывает буферную зону к HEAD. Это очищает буферную зону от изменений, которые мы только что проиндексировали. 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. Переключимся на версию коммита

```

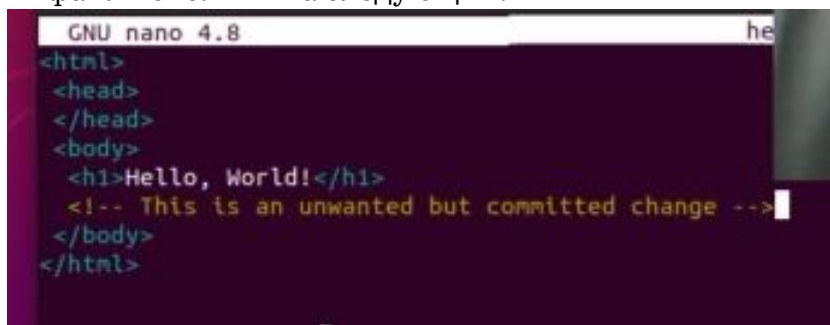
vnignatenkova@natnodd:~/hello$ git checkout hello.html
Updated 1 path from the index
vnignatenkova@natnodd:~/hello$ git status
On branch master
nothing to commit, working tree clean

```

Наш рабочий каталог опять чист.

3.7 Отмена коммитов

Отмена коммитов Мы отменим коммит путем создания нового коммита, отменяющего нежелательные изменения. Изменим файл и сделаем коммит Изменим файл hello.html на следующий.



```
GNU nano 4.8
<html>
<head>
</head>
<body>
<h1>Hello, World!</h1>
<!-- This is an unwanted but committed change -->
</body>
</html>
```

Выполним следующие команды:



```
vnignatenkova@natmod:~/hello$ git add hello.html
vnignatenkova@natmod:~/hello$ git commit -m 'Oops, we didn't want this commit'
>
> q
> ^C
vnignatenkova@natmod:~/hello$ git commit -m 'Oops, we didn't want this commit'
> ^C
vnignatenkova@natmod:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

vnignatenkova@natmod:~/hello$ git commit -m "Oops, we didn't want this commit"
[master 51586b7] Oops, we didn't want this commit
1 file changed, 1 insertion(+)
```

Сделаем коммит с новыми изменениями,отменяющими предыдущие Чтобы отменить коммит,нам необходимо сделать коммит, который удаляет изменения, сохраненные нежелательным коммитом.



```
vnignatenkova@natmod:~/hello$ git revert HEAD
[master 44b0335] Revert "Oops, we didn't want this commit"
1 file changed, 1 deletion(-)
```

Проверим лог Проверка лога показывает нежелательные и отмененные коммиты в наш репозиторий.

```

vnignatenkova@matmodd:~/hello$ git log
commit 44b033523859da1ad6ed292001653c274cd85a8a (HEAD -> master)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:41:40 2025 +0300

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

    This reverts commit 51586b76e73d3ffc3f6be6d51284bd5452e77916.

commit 51586b76e73d3ffc3f6be6d51284bd5452e77916
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:41:06 2025 +0300

    Oops, we didn't want this commit

commit 24363a89f6b3544c55bfd014bdce4b5e606168e1 (tag: v1)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:21:41 2025 +0300

    Added HTML header

commit 19a2de4ba0c1352570762e0e604f5d676308700d (tag: v1-beta)
Author: vnignatenkovarudn <1132226497@pfur.ru>

```

3.8 Удаление коммитов из ветки

Проверим нашу историю. Давайте сделаем быструю проверку нашей истории КОММИТОВ.

```

vnignatenkova@matmodd:~/hello$ git log
commit 44b033523859da1ad6ed292001653c274cd85a8a (HEAD -> master)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:41:40 2025 +0300

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

    This reverts commit 51586b76e73d3ffc3f6be6d51284bd5452e77916.

commit 51586b76e73d3ffc3f6be6d51284bd5452e77916
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:41:06 2025 +0300

    Oops, we didn't want this commit

commit 24363a89f6b3544c55bfd014bdce4b5e606168e1 (tag: v1)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:21:41 2025 +0300

```

Мы видим, что два последних коммита в этой ветке — «Oops» и «Revert Oops». Давайте удалим их с помощью сброса. Для начала отметим эту ветку. Но прежде чем удалить коммиты, давайте отметим последний коммит тегом, чтобы потом можно было его найти.

```

vnignatenkova@matmodd:~/hello$ git tag oops
vnignatenkova@matmodd:~/hello$ git reset --hard

```

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 с помощью тега.

```
vnignatenkova@natmodd:~/hello$ git reset --hard v1
HEAD is now at 24363a8 Added HTML header
vnignatenkova@natmodd:~/hello$ git log
commit 24363a89f6b3544c55bfd014bdce4b5e606168e1 (HEAD -> master, tag: v1)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:21:41 2025 +0300

    Added HTML header

commit 19a2de4ba0c1352570762e0e604f5d676308700d (tag: v1-beta)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:20:54 2025 +0300

    Added standard HTML page tags

commit aaa292e91a2263a5aa4a9bd2f9befd896f52f22b
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:16:04 2025 +0300

    «Added h1 tag»

commit bbf8139cf8bdae5069fc729a6540f3007e909de2
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:11:23 2025 +0300

    Commit 1.3.2

commit 573d29a2fb1b5f16f348a209d6e24b0edfe4d228
Author: vnignatenkovarudn <1132226497@pfur.ru>
```

Ничего никогда не теряется Давайте посмотрим на все коммиты.

```
vnignatenkova@natmodd:~/hello$ git log --all
commit 44b033523859da1ad6ed292001653c274cd85a8a (tag: oops)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:41:40 2025 +0300

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

    This reverts commit 51586b76e73d3ffc3f6be6d51284bd5452e77916.

commit 51586b76e73d3ffc3f6be6d51284bd5452e77916
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:41:06 2025 +0300

    Oops, we didn't want this commit

commit 24363a89f6b3544c55bfd014bdce4b5e606168e1 (HEAD -> master, tag: v1)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:21:41 2025 +0300

    Added HTML header

commit 19a2de4ba0c1352570762e0e604f5d676308700d (tag: v1-beta)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:20:54 2025 +0300

    Added standard HTML page tags
```

Мы видим, что ошибочные коммиты не исчезли. Они все еще находятся в репозитории. Просто они отсутствуют в ветке master. Опасность сброса Сброс в локальных ветках, как правило, безопасен. Последствия любой «аварии» как пра-

вило, можно восстановить простым сбросом с помощью нужного коммита.

3.9 Удаление тега оops

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

```
vnignatenkova@natmodd:~/hello$ git tag -d oops
Deleted tag 'oops' (was 44b0335)
vnignatenkova@natmodd:~/hello$ git log --all
commit 24363a89f6b3544c55bfd014bdce4b5e606168e1 (HEAD -> master, tag: v1)
Author: vignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:21:41 2025 +0300

    Added HTML header

commit 19a2de4ba0c1352570762e0e604f5d676308700d (tag: v1-beta)
Author: vignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:20:54 2025 +0300

    Added standard HTML page tags

commit aaa292e91a2263a5aa4a9bd2f9befd896f52f22b
Author: vignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:16:04 2025 +0300

    «Added h1 tag»

commit bbf8139cf8bdae5069fc729a6540f3007e909de2
Author: vignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:11:23 2025 +0300

    Commit 1.3.2

commit 573d29a2fb1b5f16f348a209d6e24b6edfe4d228
Author: vignatenkovarudn <1132226497@pfur.ru>
```

Тег «oops» больше не будет отображаться в репозитории.

3.10 Внесение изменений в коммиты

Изменим страницу, а затем сделаем коммит Добавим в страницу комментарий автора (вставим свою фамилию).

```
GNU nano 4.8
<!-- Author: Varvara N. Ignatenkova -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Выполним:

```
vnignatenkova@matmodd:~/hello$ git add hello.html
vnignatenkova@matmodd:~/hello$ git commit -m "Add an author commen"
[master 0aad2b7] Add an author commen
1 file changed, 1 insertion(+)
```

Необходим email Обновим страницу hello, включив в нее email.

```
GNU nano 4.8
<!-- Author: Varvara N. Ignatenkova (1132226497@pfur.ru) -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Изменим предыдущий коммит Выполним:

```
vnignatenkova@matmodd:~/hello$ git add hello.html
vnignatenkova@matmodd:~/hello$ git commit -m "Add an author commen"
[master 0aad2b7] Add an author commen
1 file changed, 1 insertion(+)
```

Просмотр истории Выполним:

```
vnignatenkova@matmodd:~/hello$ git log
commit 8c60bc09d773a5ed99ec8cbd375433858901e66c (HEAD -> master)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:47:50 2025 +0300

    Add an author/email commen

commit 24363a89f6b3544c55bfd014bdce4b5e606168e1 (tag: v1)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:21:41 2025 +0300

    Added HTML header

commit 19a2de4ba0c1352570762e0e604f5d676308700d (tag: v1-beta)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:20:54 2025 +0300

    Added standard HTML page tags
```

Мы можем увидеть, что оригинальный коммит «автор» заменен коммитом «ав

тор/email». Этого же эффекта можно достичь путем сброса последнего коммита в ветке, и повторного коммита новых изменений.

3.11 Перемещение файлов

Переместим файл `hello.html` в каталог `lib`. Сейчас мы собираемся создать структуру нашего репозитория. Давайте перенесем страницу в каталог `lib`.

```
vnignatenkova@natnodd:~/hello$ mkdir lib
vnignatenkova@natnodd:~/hello$ git mv hello.html lib
vnignatenkova@natnodd:~/hello$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:    hello.html -> lib/hello.html
```

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 этих факта сразу же проиндексированы и готовы к коммиту. Команда `git status` сообщает, что файл был перемещен.

3.12 Второй способ перемещения файлов

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` Коммит в новый каталог. Давайте сделаем коммит этого перемещения:

```
vnignatenkova@natnodd:~/hello$ git commit -m "Moved hello.html to lib"
[master 128974d] Moved hello.html to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)
vnignatenkova@natnodd:~/hello$
```

3.13 Подробнее о структуре

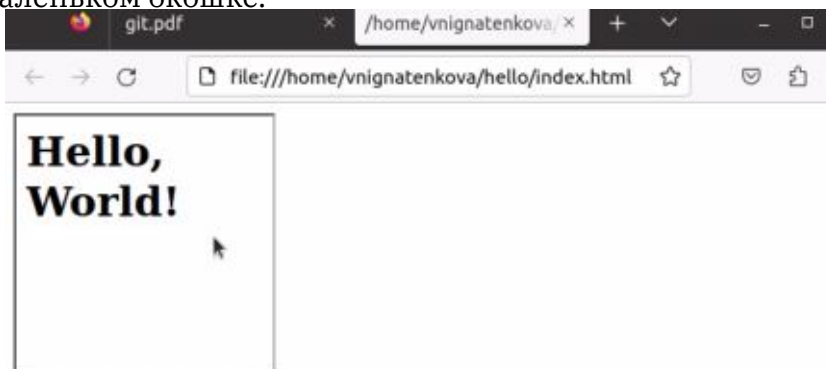
Добавление `index.html` Добавим файл `index.html` в наш репозиторий.


```
GNU nano 4.8
<html>
<body>
  <iframe src="lib/hello.html" width="200" height="
</body>
</html>
```

Добавим файл и сделаем коммит.

```
vnignatenkova@matnodd:~/hello$ git add index.html
vnignatenkova@matnodd:~/hello$ git commit -m 'Added index.html.'
[master 65a51fa] Added index.html.
1 file changed, 5 insertions(+)
create mode 100644 index.html
```

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



3.14 Git внутри: Каталог.git

Каталог .git Выполним:

```
vnignatenkova@matnodd:~/hello$ ls -C .git
branches  COMMIT_EDITMSG  config  description  HEAD  hooks  index  info  logs  objects  ORIG_HEAD  refs
vnignatenkova@matnodd:~/hello$
```

Это каталог, в котором хранится вся информация git. База данных объектов Выполним:

```
vnignatenkova@matnodd:~/hello$ ls -C .git/objects
0a  19  31  44  51  57  5e  65  82  8a  99  a9  bb  d3  f7  pack
12  24  43  47  52  59  62  72  84  8c  9c  aa  c3  f2  info
vnignatenkova@matnodd:~/hello$ ls -C .git/objects/info
```

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

Углубляемся в базу данных объектов Выполним:

```
vnignatenkova@natnodd:~/hello$ ls -C .git/objects/19
a2de4ba0c1352570762e0e604f5d676308700d
vnignatenkova@natnodd:~/hello$ git log
commit 65a51fa29143172c567639c65bf44df29cc67923 (HEAD -> master)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:58:07 2025 +0300

    Added index.html.

commit 128974d1caef304a5cc2ed7af56ebaf9819381dc
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:51:59 2025 +0300

    Moved hello.html to lib

commit 8c60bc09d773a5ed99ec8cbd375433858901e66c
Author: vnignatenkovarudn <1132226497@pfur.ru>
```

Смотрим в 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. Config File Выполним:

```
vnignatenkova@natnodd:~/hello$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
```

Это 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. Ветки и теги Выполним:

```
vnignatenkova@natnodd:~/hello$ ls .git/refs
heads  tags
vnignatenkova@natnodd:~/hello$ ls .git/refs/heads
master
vnignatenkova@natnodd:~/hello$ ls .git/refs/tags
v1  v1-beta
vnignatenkova@natnodd:~/hello$ ls .git/refs/tags/v1
.git/refs/tags/v1
```

Каждый файл соответствует тегу, ранее созданному с помощью команды git tag. Его содержание — это всего лишь хэш коммита, привязанный к тегу. Файл HEAD Выполним:

```

vnignatenkova@matnodd:~/hello$ cat .git/HEAD
ref: refs/heads/master
vnignatenkova@matnodd:~/hello$

```

Figure 3.2: 64

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

3.15 Работа непосредственно с объектами git

Поиск последнего коммита Выполним:

```

vnignatenkova@matnodd:~/hello$ git log --max-count=1
commit 65a51fa29143172c567639c65bf44df29cc67923 (HEAD -> master)
Author: vignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:58:07 2025 +0300

    Added index.html.
vnignatenkova@matnodd:~/hello$

```

Эта команда должна показать последний коммит в репозиторий. SHA1 хэш в нашей системе, вероятно, отличается от моего, но мы увидим что-то наподобие этого. Вывод последнего коммита с помощью SHA1 хэша Выполним:

```

vnignatenkova@matnodd:~/hello$ git cat-file -t 65a51fa29143172c567639c65bf44df29cc67923
commit
vnignatenkova@matnodd:~/hello$ git cat-file -p 65a51fa29143172c567639c65bf44df29cc67923
tree d3817c0b43a2622e0e6197cc17de3e8edb6137f2
parent 128974d1caef304a5cc2ed7af56ebaf9819381dc
author vignatenkovarudn <1132226497@pfur.ru> 1739469487 +0300
committer vignatenkovarudn <1132226497@pfur.ru> 1739469487 +0300

    Added index.html.
vnignatenkova@matnodd:~/hello$

```

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). Используем SHA1 хэш из строки «дерева».

```

vnignatenkova@matnodd:~/hello$ git cat-file -p d3817c0b43a2622e0e6197cc17de3e8edb6137f2
100644 blob 47b5d9ec954ec9b213762f4dcfd9c71a855dce4    index.html
040000 tree 593f618b88fbec0437554e75d84211a26c0ef9cf    lib
vnignatenkova@matnodd:~/hello$

```

Вывод каталога lib Выполним:

```

vnignatenkova@matnodd:~/hello$ git cat-file -p 593f618b88fbec0437554e75d84211a26c0ef9cf
100644 blob c392af025807ec1514c177a077a8b9e10465b071    hello.html
vnignatenkova@matnodd:~/hello$ git cat-f

```

Вывод файла hello.html Выполним:

```

vnignatenkova@matnodd:~/hello$ git cat-file -p 593f618b88fbec0437554e75d84211a26c0ef9cf
100644 blob c392af025807ec1514c177a077a8b9e10465b071    hello.html

```

Исследуем самостоятельно Нахождение оригинального файла hello.html с самого первого коммита вручную по ссылкам SHA1 хэша в последнем коммите.

```

vnignatenkova@matnodd:~/hello$ git cat-file -p aaa292e91a2263a5aa4a9bd2f9befd896f52f22b
tree 629377ed8eb96afede9fd6cb7b48b1e4a1d3dc56
parent bbfb139cf8bdae5069fc729a6540f3007e909de2
author vnignatenkovarudn <1132226497@pfur.ru> 1739466964 +0300
committer vnignatenkovarudn <1132226497@pfur.ru> 1739466964 +0300

«Added h1 tag»
vnignatenkova@matnodd:~/hello$ git cat-file -p 629377ed8eb96afede9fd6cb7b48b1e4a1d3dc56
100644 blob f76d8e928f10c56e115b18d77a21a0305d1b5e2f    hello.html
vnignatenkova@matnodd:~/hello$ git cat-file -p f76d8e928f10c56e115b18d77a21a0305d1b5e2f
<h1>Hello, World!</h1>
vnignatenkov@matnodd:~/hello$

```

3.16 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. Создадим ветку. Давайте назовем нашу новую ветку «style».

```

vnignatenkova@matnodd:~/hello$ git checkout -b style
Switched to a new branch 'style'

```

Добавим файл стилей style.css. Выполним:

```

vnignatenkova@matnodd:~/hello$ touch lib/style.css
vnignatenkova@matnodd:~/hello$ nano lib/style.css
vnignatenkova@matnodd:~/hello$ git add lib/style.css
vnignatenkova@matnodd:~/hello$ git commit -m 'Added css stylesheet'
[style 57f0a77] Added css stylesheet
1 file changed, 3 insertions(+)
create mode 100644 lib/style.css

```

Файл стилей:

```

GNU nano 4.8
1  {
2  color: red;
3  }

```

Изменим основную страницу. Обновим файл hello.html, чтобы использовать стили style.css.

```

GNU nano 4.8
<!-- Author: Varvara N. Ignatenkova (1132226497@pfu
<html>
<head>
  <link type="text/css" rel="stylesheet"
media="all" href="style.css" />
</head>
<body>
  <h1>Hello, World!</h1>
</body>
</html>

```

Выполним:

```

vnignatenkova@matmodd:~/hello$ git add lib/hello.html
vnignatenkova@matmodd:~/hello$ git commit -m "Hello uses style.css"
[style e025cca] Hello uses style.css
1 file changed, 2 insertions(+)
vnignatenkova@matmodd:~/hello$

```

Изменим index.html Обновим файл index.html, чтобы он тоже использовал style.css.

```

GNU nano 4.8
<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>

```

Выполним:

```

vnignatenkova@matmodd:~/hello$ git add index.html
vnignatenkova@matmodd:~/hello$ git commit -m "Updated index.html"
[style f048bee] Updated index.html
1 file changed, 4 insertions(+)
vnignatenkova@matmodd:~/hello$

```

3.17 Навигация по веткам

Теперь в вашем проекте есть две ветки:


```

commit f048beee368c07066e06455b7ebad273c4a99830 (HEAD)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 21:18:34 2025 +0300

    Updated index.html

commit e025cca58fc018b6b0b92d0d3202b85ba6b56c3f
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 21:16:34 2025 +0300

    Hello uses style.css

commit 57f0a774305cab57dbc714b68c272eb288ff3f3a
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 21:15:05 2025 +0300

    Added css stylesheet

commit 65a51fa29143172c567639c65bf44df29cc67923 (master)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:58:07 2025 +0300

    Added index.html.

commit 128974d1caef304a5cc2ed7af56ebaf9819381dc
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:51:59 2025 +0300

    Moved hello.html to lib

commit 8c60bc09d773a5ed99ec8cbd375433850901e66c
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:47:50 2025 +0300
vnignatenkova@matmodd:~/hello$ git checkout

```

Переключение на ветку master Используем команду git checkout для переключения междуветками:

```

vnignatenkova@matmodd:~/hello$ git checkout master
Switched to branch 'master'
vnignatenkova@matmodd:~/hello$ cat lib/hello.html
<!-- Author: Varvara N. Ignatenkova (1132226497@pfur.ru) -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```

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

```

vnignatenkova@matmodd:~/hello$ git checkout style
Switched to branch 'style'

```

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

3.18 Изменения в ветке master

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

```
vnignatenkova@matnodd:~/hello$ git checkout master
Switched to branch 'master'
vnignatenkova@matnodd:~/hello$ echo "This is the Hello World example from the git tutorial." > README.md
```

3.19 Сделаем коммит изменений README.md в ветку master.

Выполним:

```
vnignatenkova@matnodd:~/hello$ git add README.md
vnignatenkova@matnodd:~/hello$ git commit -m "Added README"
[master d423fab] Added README
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

Просмотр отличающихся веток Просмотр текущих веток Теперь у нас в репозитории есть две отличающиеся ветки. Используем следующую лог-команду для просмотра веток и их отличий.

```
vnignatenkova@matnodd:~/hello$ git log
* commit d423faba5bd229d0b831bb191152663e9e975802 (
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:22:10 2025 +0300

    Added README

* commit f048beee368c07066e06455b7ebad273c4a99830 (style)
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:18:34 2025 +0300

    Updated index.html

* commit e025cca58fc018b6b0b92d0d3202b85ba6b56c3f
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:16:34 2025 +0300

    Hello uses style.css

* commit 57f0a774305cab57dbc714b68c272eb288ff3f3a
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:15:05 2025 +0300

    Added css stylesheet

* commit 65a51fa29143172c567639c65bf44df29cc67923
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 20:58:07 2025 +0300

    Added index.html.

* commit 128974d1caef304a5cc2ed7af56ebaf9819381dc
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 20:51:59 2025 +0300
```

3.20 Слияние

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

```
vnignatenkova@matnodd:~/hello$ git checkout style
Switched to branch 'style'
vnignatenkova@matnodd:~/hello$ ls
index.html  lib
vnignatenkova@matnodd:~/hello$ git merge master
Merge made by the 'recursive' strategy.
 README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```


3.21 Создание конфликта

Вернемся в master и создадим конфликт Вернемся в ветку master и внесем следующие изменения:

```
vnignatenkova@matmodd:~/hello$ git checkout master  
switched to branch 'master'
```

Файл lib/hello.html

```
GNU nano 4.8 lib/  
<!-- Author: Varvara N. Ignatenkova (1132226497@pfu  
<html>  
<head>  
  <!-- no style -->  
</head>  
<body>  
  <h1>Hello, World!</h1>  
</body>  
</html>
```

Выполним:

```
vnignatenkova@matmodd:~/hello$ git add lib/hello.html  
vnignatenkova@matmodd:~/hello$ git commit -m 'Life is great'  
[master 731b3e4] Life is great  
1 file changed, 1 insertion(+)
```

Просмотр веток Выполним:

```
1 file changed, 1 insertion(+)
vnignatenkova@matnodd:~/hello$ git log --graph --all
* commit 731b3e45b6479c67ab472a66097b7400477ae463 (
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:30:27 2025 +0300

    Life is great

* commit 64279dd159d72a62b8d6d5b61e5ef7d7d1470bc1 (style)
  Merge: f048bee d423fab
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:28:23 2025 +0300

    Merge branch 'master' into style

* commit d423faba5bd229d0b831bb191152663e9e975802
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:22:10 2025 +0300

    Added README

* commit f048beee368c07066e06455b7ebad273c4a99830
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:18:34 2025 +0300

    Updated index.html

* commit e025cca58fc018b6b0b92d0d3202b05ba6b56c3f
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:16:34 2025 +0300

    Hello uses style.css
```

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. Последнее изменение в master конфликтует с некоторыми изменениями в style. На следующем шаге мы решим этот конфликт.

3.22 Разрешение конфликтов

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

```
vnignatenkova@matnodd:~/hello$ git checkout style
Switched to branch 'style'
vnignatenkova@matnodd:~/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.
```

Откроем lib/hello.html.

```

GNU nano 4.8 lib/
<!-- Author: Varvara N. Ignatenkova (1132226497@pfu
<html>
<head>
<<<<<<< HEAD
  <link type="text/css" rel="stylesheet"
media="all" href="style.css" />
=====
  <!-- no style -->
>>>>>> master
</head>
<body>
  <h1>Hello, World!</h1>
</body>
</html>

```

Первый раздел—версия текущей ветки (style). Второй раздел—версия ветки master. Решение конфликта Внесем изменения в lib/hello.html для достижения следующего результата.

```

GNU nano 4.8 lib/
<!-- Author: Varvara N. Ignatenkova (1132226497@pfu
<html>
<head>
  <link type="text/css" rel="stylesheet"
media="all" href="style.css" />
</head>
<body>
  <h1>Hello, World! Life is great!</h1>
</body>
</html>

```

Сделаем коммит решения конфликта Выполним:

```

vignatenkova@natnodd:~/hello$ git add lib/hello.html
vignatenkova@natnodd:~/hello$ git commit -m "Merged master fixed conflict."
[style 38c5d2d] Merged master fixed conflict.

```

Перебазирование как альтернатива слиянию Мы будем использовать команду reset для возврата веток к предыдущему состоянию.

3.23 Сброс ветки style

Сброс ветки style Вернемся на ветке style к точке перед тем, как мы слили ее с веткой master. Выполним:

```

vnignatenkova@matmodd:~/hello$ git checkout style
Already on 'style'
vnignatenkova@matmodd:~/hello$ git log --graph
*   commit 38c5d2dca1e2f8cec708c5255ae54213ed969717 (HEAD -> style)
|   Merge: 64279dd 731b3e4
|   Author: vignatenkovarudn <1132226497@pfur.ru>
|   Date: Thu Feb 13 21:34:00 2025 +0300
|
|       Merged master fixed conflict.
|
|   *   commit 731b3e45b64791c67ab472a66097b7400477ae463 (master)
|   |   Author: vignatenkovarudn <1132226497@pfur.ru>
|   |   Date: Thu Feb 13 21:30:27 2025 +0300
|   |
|   |       Life is great
|   |
|   *   commit 64279dd159d72a62b8d6d5b61e5ef7d7d1470bc1
|   |   Merge: f048bee d423fab
|   |   Author: vignatenkovarudn <1132226497@pfur.ru>
|   |   Date: Thu Feb 13 21:28:23 2025 +0300
|   |
|   |       Merge branch 'master' into style
|   |
|   *   commit d423faba5bd229d0b831bb191152663e9e975802
|   |   Author: vignatenkovarudn <1132226497@pfur.ru>
|   |   Date: Thu Feb 13 21:22:10 2025 +0300
|   |
|   |       Added README
|

```

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

```

vnignatenkova@matmodd:~/hello$ git reset --hard d423faba5bd229d0b831bb191152663e9e975802
HEAD is now at d423fab Added README
vnignatenkova@matmodd:~/hello$ git log --graph --all
*   commit d423faba5bd229d0b831bb191152663e9e975802 (HEAD -> master)
|   Author: vignatenkovarudn <1132226497@pfur.ru>
|   Date: Thu Feb 13 21:22:10 2025 +0300
|
|       Added README
|
|   *   commit f048beee368c07066e06455b7ebad273c4a99830 (style)
|   |   Author: vignatenkovarudn <1132226497@pfur.ru>
|   |   Date: Thu Feb 13 21:18:34 2025 +0300
|   |
|   |       Updated index.html
|   |
|   *   commit e025cca58fc018b6b0b92d0d3202b85ba6b56c3f
|   |   Author: vignatenkovarudn <1132226497@pfur.ru>
|   |   Date: Thu Feb 13 21:16:34 2025 +0300
|   |
|   |       Hello uses style.css
|

```

Проверьте ветку. Выполним:

```

vnignatenkova@natmodd:~/hello$ git checkout style
Switched to branch 'style'
vnignatenkova@natmodd:~/hello$ git rebase master
First, rewinding head to replay your work on top of
Applying: Added css stylesheet
Applying: Hello uses style.css
Applying: Updated index.html
vnignatenkova@natmodd:~/hello$ git log --graph
* commit 8b24839935f8c9b3fd0015faae81a747348be22d (HEAD -> sty
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:18:34 2025 +0300

    Updated index.html

* commit cdd52420e7fa9f9df4451c882190b923a03c1c28
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:16:34 2025 +0300

    Hello uses style.css

* commit 1c4fb4cc29abf37bffa64daeb9cdc6cb697aedb82
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:15:05 2025 +0300

    Added css stylesheet

* commit d423faba5bd229d0b831bb191152663e9e975802 (master)
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:22:10 2025 +0300

    Added README

* commit 65a51fa29143172c567639c65bf44df29cc67923
  Author: vnignatenkovarudn <1132226497@pfur.ru>

```

3.24 Сброс ветки master

Сброс ветки master Давайте вернемся в ветке master в точку перед внесением конфликтующих изменений.

```

vnignatenkova@natmodd:~/hello$ git checkout master
Switched to branch 'master'
vnignatenkova@natmodd:~/hello$ git log --graph
* commit 731b3e45b6479c67ab472a66097b7400477ae463 (HEAD -> master)
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:30:27 2025 +0300

    Life is great

* commit d423faba5bd229d0b831bb191152663e9e975802
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:22:10 2025 +0300

    Added README

* commit 65a51fa29143172c567639c65bf44df29cc67923
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 20:58:07 2025 +0300

    Added index.html.

```


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

```
vnignatenkova@natnodd:~/hello$ git reset --hard d423faba5bd229d0b831bb191152663e9e975802
HEAD is now at d423fab Added README
vnignatenkova@natnodd:~/hello$ git log --graph --all
* commit d423faba5bd229d0b831bb191152663e9e975802 (HEAD -> master)
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:22:10 2025 +0300

    Added README

* commit f048beee368c07066e06455b7ebad273c4a99830 (style)
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:18:34 2025 +0300

    Updated index.html

* commit e025cca58fc018b6b0b92d0d3202b85ba6b56c3f
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:16:34 2025 +0300

    Hello uses style.css
```

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

3.25 Перебазирование

Используем команду rebase вместо команды merge.

```
vnignatenkova@matnodd:~/hello$ git checkout style
Switched to branch 'style'
vnignatenkova@matnodd:~/hello$ git rebase master
First, rewinding head to replay your work on top of
Applying: Added css stylesheet
Applying: Hello uses style.css
Applying: Updated index.html
vnignatenkova@matnodd:~/hello$ git log --graph
* commit 8b24839935f8c9b3fd0015faae81a747348be22d (HEAD -> style)
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:18:34 2025 +0300

    Updated index.html

* commit cdd52420e7fa9f9df4451c882190b923a03c1c28
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:16:34 2025 +0300

    Hello uses style.css

* commit 1c4fb4cc29abf37b9f64daeb9cdc6cb697aedb82
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:15:05 2025 +0300

    Added css stylesheet

* commit d423faba5bd229d0b831bb191152663e9e975802 (master)
  Author: vnignatenkovarudn <1132226497@pfur.ru>
  Date: Thu Feb 13 21:22:10 2025 +0300

    Added README

* commit 65a51fa29143172c567639c65bf44df29cc67923
  Author: vnignatenkovarudn <1132226497@pfur.ru>
```

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 расширенной, 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 git rebase для кратковременных, локальных веток, а слияние для веток в публичном репозитории.

3.26 Слияние в ветку master

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

```
vnignatenkova@matnodd:~/hello$ git checkout master
Switched to branch 'master'
vnignatenkova@matnodd:~/hello$ git merge style
Updating d423fab..8b24839
Fast-forward
 index.html      | 4 ++++
 lib/hello.html  | 2 ++
 lib/style.css   | 3 +++
 3 files changed, 9 insertions(+)
 create mode 100644 lib/style.css
```

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. Просмотрим логи. Выполним:


```

commit 8b24839935f8c9b3fd0015faae81a747348be22d (HEAD)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 21:18:34 2025 +0300

    Updated index.html

commit cdd52420e7fa9f9df4451c882190b923a03c1c2B
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 21:16:34 2025 +0300

    Hello uses style.css

commit 1c4fb4cc29abf37bff64daeb9cdc6cb697aedb82
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 21:15:05 2025 +0300

    Added css stylesheet

commit d423faba5bd229d0b831bb191152663e9e975802
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 21:22:10 2025 +0300

    Added README

commit 65a51fa29143172c567639c65bf44df29cc67923
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:58:07 2025 +0300

    Added index.html.

commit 128974d1caef304a5cc2ed7af56ebaf9819381dc
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 20:51:59 2025 +0300

```

Теперь ветки style и master идентичны.

3.27 Клонирование репозитория

Перейдем в рабочий каталог. Выполним:

```

vnignatenkova@matnodd:~/hello$ cd ..
vnignatenkova@matnodd:~$ pwd
/home/vnignatenkova
vnignatenkova@matnodd:~$ ls

```

Создадим клон репозитория hello. Выполним:

```

vnignatenkova@matnodd:~$ git clone hello cloned_hello
Cloning into 'cloned_hello'...
done.
vnignatenkova@matnodd:~$ ls
cloned_hello Desktop Documents Downloads hello Music Pictures Public Templates Videos
vnignatenkova@matnodd:~$

```

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

3.28 Просмотр клонированного репозитория

Давайте взглянем на клонированный репозиторий. Выполним:

```
vnignatenkova@natmodd:~$ cd cloned_hello
vnignatenkova@natmodd:~/cloned_hello$ ls
index.html  lib  README.md
```

Мы увидим список всех файлов на верхнем уровне оригинального репозитория README.md, index.html и lib. Просмотрите историю репозитория

Выполним:

```
vnignatenkova@natmodd:~/cloned_hello$ git log --all
commit 8b24839935f8c9b3fd0015faae81a747348be22d (HEAD -> master, origin/style, origin/master, origin/HEAD)
Author: vnignatenkovarudn <i132226497@pfur.ru>
Date: Thu Feb 13 21:18:34 2025 +0300

    Updated index.html

commit cdd52420e7fa9f9df4451c882190b923a03c1c28
Author: vnignatenkovarudn <i132226497@pfur.ru>
Date: Thu Feb 13 21:16:34 2025 +0300

    Hello uses style.css

commit 1c4fb4cc29abf37bffe4daeb9cdc6cb697aedb82
Author: vnignatenkovarudn <i132226497@pfur.ru>
Date: Thu Feb 13 21:15:05 2025 +0300

    Added css stylesheet

commit d423faba5bd229d0b831bb191152663e9e975802
Author: vnignatenkovarudn <i132226497@pfur.ru>
Date: Thu Feb 13 21:22:10 2025 +0300

    Added README

commit 65a51fa29143172c567639c65bf44df29cc67923
```

Мы увидим 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. Удаленные ветки Мы увидим ветку master (HEAD) в списке истории. Мы также увидим ветки со странными именами (origin/master, origin/style и origin/HEAD).

3.29 Что такое origin?

Выполним:

```
vnignatenkova@natmodd:~/cloned_hello$ git remote
origin
```

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:

```
vnignatenkova@natnodd:~/cloned_hello$ git remote show origin
* remote origin
  Fetch URL: /home/vnignatenkova/hello
  Push URL: /home/vnignatenkova/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)
```

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).

3.30 Удаленные ветки

Выполним:

```
vnignatenkova@natnodd:~/cloned_hello$ git branch
* master
```

Как мы видим, в списке только ветка master. Где ветка style? Команда git branch выводит только список локальных веток по умолчанию. Список удаленных веток. Для того, чтобы просмотреть все ветки, выполним:

```
vnignatenkova@natnodd:~/cloned_hello$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/style
vnignatenkova@natnodd:~/cloned_hello$
```

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.

3.31 Изменение оригинального репозитория

Внесем некоторые изменения в оригинальный репозиторий, чтобы затем попытаться извлечь и слить изменения из удаленной ветки в текущую. Внесение изменения в оригинальный репозиторий hello Выполним:

```
vnignatenkova@matnodd:~/cloned_hello$ cd ../hello/  
vnignatenkova@matnodd:~/hello$
```

Внесем следующие изменения в файл README.md: Файл README.md

```
This is the Hello World example from the git tutorial  
This is the Hello World example from the git tutorial
```

Теперь добавим это изменение и сделаем коммит.

```
vnignatenkova@matnodd:~/hello$ git add README.md  
vnignatenkova@matnodd:~/hello$ git commit -m "Changed README in original repo"  
[master 71b4bf5] Changed README in original repo  
1 file changed, 1 insertion(+)
```

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

```
vnignatenkova@matnodd:~/hello$ cd ../cloned_hello/  
vnignatenkova@matnodd:~/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)  
Unpacking objects: 100% (3/3), 363 bytes | 181.00 KiB/s, done.  
From /home/vnignatenkova/hello  
8b24839..71b4bf5 master -> origin/master
```

Команда git fetch будет извлекать новые коммиты из удаленного репозитория, но не будет сливать их с вашими наработками в локальных ветках. Проверим README.md Выполним:

```
vnignatenkova@matnodd:~/cloned_hello$ cat README.md  
This is the Hello World example from the git tutorial.  
vnignatenkova@matnodd:~/cloned_hello$
```

3.32 Слияние извлеченных изменений

Сольем извлеченные изменения в локальную ветку Выполним:

```
vnignatenkova@matnodd:~/cloned_hello$ cat README.md  
This is the Hello World example from the git tutorial.
```

Еще раз проверим файл README.md Сейчас мы должны увидеть изменения.

Выполним:

```
vnignatenkova@natnodd:~/cloned_hello$ git pull
Updating 8b24839..71b4bf5
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
```

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

3.33 Добавление ветки наблюдения

Ветки, которые начинаются с `remotes/origin` являются ветками оригинального репозитория. Добавим локальную ветку, которая отслеживает удаленную ветку

Выполним:

```
vnignatenkova@natnodd:~/cloned_hello$ git branch --track style origin/style
Branch 'style' set up to track remote branch 'style' from 'origin'.
vnignatenkova@natnodd:~/cloned_hello$ git branch -a
* master
  style
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/style

vnignatenkova@natnodd:~/cloned_hello$ git log --max-count=2
commit 71b4bf585b1446a3b0a5970db1b2890985882578 (HEAD -> master, origin/master, origin/HEAD)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 21:55:50 2025 +0300

    Changed README in original repo

commit 8b24839935f8c9b3fd0015faae81a747348be22d (origin/style, style)
Author: vnignatenkovarudn <1132226497@pfur.ru>
Date: Thu Feb 13 21:18:34 2025 +0300

    Updated index.html
```

Теперь мы можем видеть ветку `style` в списке веток и лог.

3.34 Чистые репозитории

Чистые репозитории (без рабочих каталогов) обычно используются для расширения. 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-репозиторийские» 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».

3.35 Создадим чистый репозиторий

```
vnignatenkova@matnodd:~/cloned_hello$ cd ..
vnignatenkova@matnodd:~$ git clone --bare hello hello.git
Cloning into bare repository 'hello.git'...
done.
vnignatenkova@matnodd:~$ ls
cloned_hello  Documents  hello      Music      Public      Videos
Desktop      Downloads  hello.git  Pictures   Templates
vnignatenkova@matnodd:~$ cd hello.git/
```

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.

3.36 Добавление удаленного репозитория

Давайте добавим репозиторий hello.git к нашему оригинальному репозиторию.

```
vnignatenkova@matnodd:~$ cd hello
vnignatenkova@matnodd:~/hello$ git remote add shared ../hello.git
vnignatenkova@matnodd:~/hello$
```

3.37 Отправка изменений

Отредактируем файл README.md и сделаем коммит.

```
GNU nano 4.8 RE
This is the Hello World example from the git tutorial
(Changed in the original and pushed to shared)
```

Выполним:


```

vnignatenkova@natmodd:~/hello$ git checkout master
M       README.md
Already on 'master'
vnignatenkova@natmodd:~/hello$ git add README.md
vnignatenkova@natmodd:~/hello$ git commit -m "Added shared comment to readme"
[master 5b63ced] Added shared comment to readme
1 file changed, 1 insertion(+), 1 deletion(-)
vnignatenkova@natmodd:~/hello$ ls
index.html  lib  README.md
vnignatenkova@natmodd:~/hello$ git push shared master

```

Теперь отправим изменения в общий репозиторий.

```

vnignatenkova@natmodd:~/hello$ git push shared master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 409 bytes | 204.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To ../hello.git
71b4bf5..5b63ced master -> master

```

Общим называется репозиторий, получающий отправленные нами изменения.

3.38 Извлечение общих изменений

Быстро переключимся в клонированный репозиторий и извлечем изменения, только что отправленные в общий репозиторий.

```

vnignatenkova@natmodd:~/hello$ cd ../cloned_hello/

```

Выполним:

```

vnignatenkova@natmodd:~/cloned_hello$ git remote add
vnignatenkova@natmodd:~/cloned_hello$ git branch --
Branch 'shared' set up to track local branch 'master'
vnignatenkova@natmodd:~/cloned_hello$ git pull shared
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)
Unpacking objects: 100% (3/3), 389 bytes | 389.00 KiB/s, done.
From ../hello
* branch                master      -> FETCH_HEAD
* [new branch]          master      -> shared/master
Updating 71b4bf5..5b63ced
Fast-forward
 README.md | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
vnignatenkova@natmodd:~/cloned_hello$ cat README.md
This is the Hello World example from the git tutorial.
(Changed in the original and pushed to shared)

```

4 Выводы

Мы ознакомились с работой с git и приобрели соответствующие навыки.