

# **Отчёт по лабораторной работе №1**

**Дисциплина: Математическое моделирование**

Ганина Таисия Сергеевна, НФИбд-01-22

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>6</b>
<b>2</b>	<b>Задание</b>	<b>7</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>9</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>12</b>
<b>5</b>	<b>Выводы</b>	<b>51</b>
<b>6</b>	<b>Список литературы</b>	<b>52</b>

## Список иллюстраций

4.1	Подготовить пространство для работы . . . . .	12
4.2	Создание проекта . . . . .	13
4.3	Создать git репозиторий из этого каталога . . . . .	13
4.4	Добавим файл в репозиторий . . . . .	13
4.5	Команда git status, чтобы проверить текущее состояние репозитория	14
4.6	Изменила файл и проверила состояние каталога . . . . .	14
4.7	Провести индексацию изменений . . . . .	15
4.8	Редактор коммита . . . . .	15
4.9	Запись коммита, проверка состояния . . . . .	16
4.10	Код файла . . . . .	16
4.11	Продолжение кода . . . . .	16
4.12	Одно изменение проиндексировано, второе нет . . . . .	17
4.13	Закоммитила проиндексированное . . . . .	17
4.14	Проиндексировала второе изменение . . . . .	17
4.15	git log . . . . .	18
4.16	Первый коммит . . . . .	19
4.17	git tag v1 . . . . .	20
4.18	git tag v1-beta . . . . .	20
4.19	git log master –all . . . . .	21
4.20	Отменить локальные изменения (до индексации) . . . . .	21
4.21	Отменить проиндексированные изменения (перед коммитом) . .	22
4.22	Делаем коммит . . . . .	22
4.23	git revert HEAD . . . . .	22
4.24	git log . . . . .	23
4.25	Давайте отметим последний коммит тегом, и выполним сброс ком- митов к предшествующим коммиту oops . . . . .	24
4.26	Оказывается, что коммиты все еще находятся в репозитории . . .	25
4.27	Удаление тега oops . . . . .	26
4.28	Добавим в страницу комментариев автора . . . . .	26
4.29	Изменим предыдущий коммит . . . . .	27
4.30	Структура нашего репозитория . . . . .	28
4.31	Добавим файл index.html . . . . .	28
4.32	Кусок страницы hello в маленьком окошке . . . . .	28
4.33	Набор каталогов, имена которых состоят из 2 символов . . . . .	29
4.34	Посмотрим файл конфигурации . . . . .	30
4.35	Файл HEAD . . . . .	30
4.36	Работа непосредственно с объектами git . . . . .	31

4.37	Самостоятельное исследование 1 . . . . .	31
4.38	Самостоятельное исследование 2 . . . . .	32
4.39	Создайте ветку, файл стилей style.css . . . . .	33
4.40	Обновим файл hello.html. Обновим файл index.html . . . . .	33
4.41	Переключения между ветками . . . . .	34
4.42	git log –graph –all . . . . .	35
4.43	Слияние веток . . . . .	36
4.44	Создание конфликта . . . . .	37
4.45	Вернемся к ветке style и попытаемся объединить ее с новой веткой master . . . . .	38
4.46	Внесём изменения в lib/hello.html . . . . .	39
4.47	Сброс ветки style . . . . .	40
4.48	Сброс ветки master . . . . .	41
4.49	Сброс ветки master . . . . .	42
4.50	Слияние в ветку master . . . . .	43
4.51	Создадим клон репозитория hello . . . . .	44
4.52	Просмотр клонированного репозитория . . . . .	44
4.53	Более подробная информация об имени по умолчанию . . . . .	45
4.54	Научиться извлекать изменения из удаленного репозитория . . . . .	45
4.55	cat README.md . . . . .	46
4.56	Добавим локальную ветку, которая отслеживает удаленную ветку . . . . .	46
4.57	Отправка изменений . . . . .	46
4.58	Извлечение общих изменений . . . . .	47
4.59	Заголовок и нумерованный список . . . . .	47
4.60	Маркированный список . . . . .	48
4.61	Жирный текст . . . . .	48
4.62	Работа с изображениями . . . . .	49
4.63	Вставка кода . . . . .	49
4.64	Ссылки . . . . .	50

## Список таблиц

# 1 Цель работы

Целью данной работы является приобретение навыков работы с системой контроля версий git. Работа с репозиториями, внесение изменений, индексация, коммиты.

## 2 Задание

1. Подготовить пространство для работы.
2. Создать проект.
3. Внести изменения в проект.
4. Провести индексацию изменений
5. Работа с тегами версий.
6. Отменить локальные изменения (до индексации).
7. Отменить проиндексированные изменения (перед коммитом).
8. Отменить коммит.
9. Удалить коммиты из ветки.
10. Удаление тега.
11. Внести изменения в коммиты.
12. Переместить файлы.
13. Рассмотреть Git внутри: Каталог .git.
14. Работа непосредственно с объектами git.
15. Навигация по веткам. Изменения в ветке master.
16. Слияние веток. Создание конфликта.
17. Разрешение конфликтов.
18. Сброс ветки style. Сброс ветки master.
19. Перебазирование.
20. Слияние в ветку master.
21. Клонирование репозитория. Просмотр клонированного репозитория.
22. Что такое origin? Удаленные ветки. Изменение оригинального репозитория.

- 23. Слияние извлеченных изменений.
- 24. Извлечение общих изменений.
- 25. Написание отчета по лабораторной работе в формате Markdown.



## 3 Теоретическое введение

Система контроля версий (Version Control System) – это инструмент, который используется для отслеживания, внесения и управления изменениями в программном коде. Это также можно назвать просто контролем версий.

Системы контроля версий помогает разработчикам сохранять изменения, внесенные в файл, на разных этапах, чтобы и они сами, и их коллеги могли их увидеть позже.

Существует три типа систем контроля версий:

- Локальные системы контроля версий
- Централизованные системы контроля версий
- Распределенные системы контроля версий

### **Что такое локальная система контроля версий (LVCS)?**

Этот тип системы контроля версий очень распространен и прост в использовании. Однако этот метод может выдавать ошибки и подвержен атакам, потому что файлы хранятся в вашей локальной системе.

Это означает, что вы можете потерять системный файл или случайно забыть каталог, с которым вы работаете (и затем записать в другой каталог). Что такое централизованная система контроля версий (CVCS)?

В этом типе контроля версий сервер работает как общее хранилище, в котором находятся все версии кода. CVCS помогает разработчикам работать совместно.

Однако, несмотря на то, что такой метод позволяет разработчикам сотрудничать, если сервер отключится на несколько секунд или будет поврежден, то есть

шанс, что вы потеряете все файлы. Это является серьезной проблемой при работе с CVCS.

В CVCS только несколько разработчиков могут работать совместно над проектом. Что такое распределенная система контроля версий (DVCS)?

В настоящее время это новый и наиболее часто используемый тип системы контроля версий.

В DVCS у каждого разработчика есть полная резервная копия всех данных на сервере. Это означает, что всякий раз, когда сервер не будет работать или будет неисправен, то вы все равно сможете работать над своим проектом, а также копировать или создавать резервные копии своих хранилищ на сервере, чтобы восстановить их.

При работе с DVCS над одним проектом может работать много разработчиков. Одной из популярнейших DVCS является Git, о которой мы сейчас будем говорить подробнее.

### **Что такое Git?**

Git – это бесплатная распределенная система контроля версий с открытым исходным кодом, которую можно использовать для отслеживания изменений в ваших файлах. В Git можно работать над всеми типами и размерами проектов.

С помощью Git вы можете добавлять изменения в свой код, а затем фиксировать их (или сохранять), когда это необходимо. Это означает, что у вас есть возможность вернуться к ранее внесенным изменениям. Git работает рука об руку с GitHub. А что же такое GitHub? Что такое GitHub?

GitHub – это веб-интерфейс, в котором можно хранить свои репозитории Git, а также эффективно отслеживать и управлять своими изменениями. С его помощью разные разработчики имеют доступ к коду одного проекта. У вас есть возможность вносить свои собственные изменения в проект одновременно с другими разработчиками.

Например, если вы вдруг допустили какую-то ошибку во время внесения изменений, вы можете легко вернуться к предыдущему этапу, где ошибки еще

нет.

### **Для чего нужно использовать GitHub?**

Есть множество причин для использования GitHub. Давайте рассмотрим некоторые из них.

#### **Эффективное управление проектами**

GitHub – это своего рода хранилище ваших репозиториях. GitHub позволяет разработчикам работать над одним проектом, находясь в разных местах.

С помощью GitHub вы можете легко отслеживать внесенные вами изменения и управлять ими, а также проверять ход вашей работы над проектом. Простое сотрудничество

С GitHub разработчики со всего мира могут работать вместе на одном проектом без каких-либо проблем.

Команды разработчиков могут оставаться на одной странице во время совместной работы над проектом и могут легко организовывать и эффективно управлять проектом.

#### **Открытый исходный код**

GitHub – это бесплатная система с открытым исходным кодом. Это означает, что разработчики могут легко получить доступ к различным типам кода/проектов, которые они могут использовать для обучения и развития своих навыков.

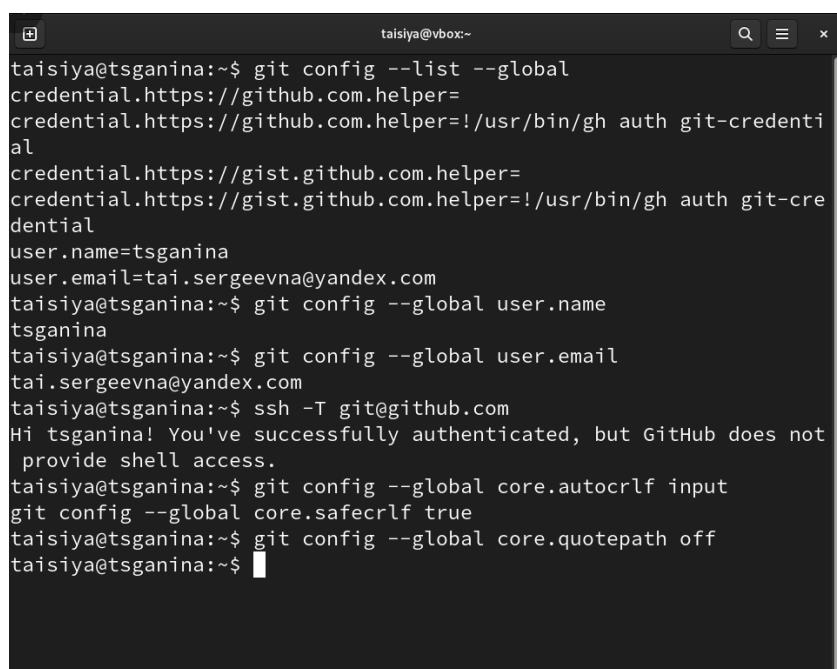
#### **Универсальность.**

Это свойство GitHub очень важно. GitHub – это веб-интерфейс не только для разработчиков. Его также могут использовать дизайнеры, писатели и все, кто хочет отслеживать историю своих проектов.

## 4 Выполнение лабораторной работы

### 1. Подготовить пространство для работы.

Я уже ранее использовала гит, поэтому сейчас выполнила несколько команд для того, чтобы вывести на экран имя и электронную почту, а после поработала с параметрами установки окончаний строк и с установкой отображения unicode. (рис. 4.1).



```
taisiya@vbox:~  
taisiya@tsganina:~$ git config --list --global  
credential.https://github.com.helper=  
credential.https://github.com.helper=!/usr/bin/gh auth git-credenti  
al  
credential.https://gist.github.com.helper=  
credential.https://gist.github.com.helper=!/usr/bin/gh auth git-cre  
dential  
user.name=tsganina  
user.email=tai.sergeevna@yandex.com  
taisiya@tsganina:~$ git config --global user.name  
tsganina  
taisiya@tsganina:~$ git config --global user.email  
tai.sergeevna@yandex.com  
taisiya@tsganina:~$ ssh -T git@github.com  
Hi tsganina! You've successfully authenticated, but GitHub does not  
provide shell access.  
taisiya@tsganina:~$ git config --global core.autocrlf input  
git config --global core.safecrlf true  
taisiya@tsganina:~$ git config --global core.quotePath off  
taisiya@tsganina:~$
```

Рис. 4.1: Подготовить пространство для работы

### 2. Создание проекта.

Я начала работу в пустом рабочем каталоге с создания пустого каталога с именем hello, затем вошла в него и создала там файл с именем hello.html.

Потом я создала git репозиторий из этого каталога. Добавила файл в репозиторий, проверила его состояние. (рис. 4.2, 4.3, 4.4, 4.5).

```
taisiya@tsganina:~$ mkdir hello
cd hello
touch hello.html
echo "Hello, World!" > hello.html
taisiya@tsganina:~/hello$ cat hello.html
Hello, World!
taisiya@tsganina:~/hello$
```

Рис. 4.2: Создание проекта

```
taisiya@tsganina:~/hello$ cat hello.html
Hello, World!
taisiya@tsganina:~/hello$ git init
подсказка: Using 'master' as the name for the initial branch. This
default branch name
подсказка: is subject to change. To configure the initial branch na
me to use in all
подсказка: of your new repositories, which will suppress this warni
ng, call:
подсказка:
подсказка:     git config --global init.defaultBranch <name>
подсказка:
подсказка: Names commonly chosen instead of 'master' are 'main', 't
runk' and
подсказка: 'development'. The just-created branch can be renamed vi
a this command:
подсказка:
подсказка:     git branch -m <name>
Инициализирован пустой репозиторий Git в /home/taisiya/hello/.git/
taisiya@tsganina:~/hello$
```

Рис. 4.3: Создать git репозиторий из этого каталога

```
Инициализирован пустой репозиторий Git в /home/taisiya/hello/.git/
taisiya@tsganina:~/hello$ git add hello.html
git commit -m "Initial Commit"
[master (корневой коммит) 64a83a2] Initial Commit
1 file changed, 1 insertion(+)
create mode 100644 hello.html
taisiya@tsganina:~/hello$
```

Рис. 4.4: Добавим файл в репозиторий

```
taisiya@tsganina:~/hello$ git status
Текущая ветка: master
нечего коммитить, нет изменений в рабочем каталоге
taisiya@tsganina:~/hello$
```

Рис. 4.5: Команда git status, чтобы проверить текущее состояние репозитория

### 3. Внесение изменений в проект.

Я добавила HTML-теги к нашему приветствию. Проверила состояние рабочего каталога, видно, что git знает, что файл hello.html был изменен, но при этом эти изменения еще не зафиксированы в репозитории. (рис. 4.6).

```
taisiya@tsganina:~/hello$ gedit hello.html
^C
taisiya@tsganina:~/hello$ git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в
рабочем каталоге)
        изменено:      hello.html

индекс пуст (используйте «git add» и/или «git commit -a»)
taisiya@tsganina:~/hello$
```

Рис. 4.6: Изменила файл и проверила состояние каталога

### 4. Провести индексацию изменений. Выполнила следующие команды, чтобы проиндексировать изменения и опять проверить состояние:

```
git add hello.html
git status
```

(рис. 4.7).

```

taisiya@tsganina:~/hello$ git add hello.html
taisiya@tsganina:~/hello$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)

    изменено:      hello.html

taisiya@tsganina:~/hello$

```

Рис. 4.7: Провести индексацию изменений

После этого я сделала коммит, но не писала метку `-m` в строке, поэтому в редакторе смогла интерактивно отредактировать комментарии для коммита. (рис. 4.8, 4.9).



```

taisiya@vbox:~/hello — git commit
GNU nano 6.4 /home/taisiya/hello/.git/COMMIT_EDITMSG
# Пожалуйста, введите сообщение коммита для ваших изменений. Строки
# начинающиеся с «#» будут проигнорированы, а пустое сообщение
# отменяет процесс коммита.
#
# Текущая ветка: master
# Изменения, которые будут включены в коммит:
#   изменено:      hello.html
#
[ Прочитано 9 строк ]
^G Справка  ^O Записать  ^W Поиск    ^K Вырезать  ^T Выполнить
^X Выход    ^R ЧитФайл  ^\ Замена   ^U Вставить  ^J Выводить

```

Рис. 4.8: Редактор коммита

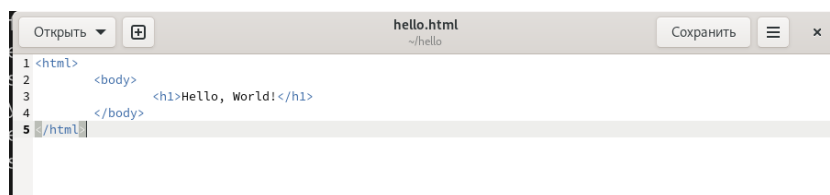
```

taisiya@tsganina:~/hello$ git commit
[master d4e86f6] Added h1 tag
1 file changed, 1 insertion(+), 1 deletion(-)
taisiya@tsganina:~/hello$ git status
Текущая ветка: master
нечего коммитить, нет изменений в рабочем каталоге
taisiya@tsganina:~/hello$

```

Рис. 4.9: Запись коммита, проверка состояния

Потом я опять изменила файл html и добавила туда стандартные теги страницы. При этом часть изменений я проиндексировала, а часть (заголовок) – нет. Произвела коммит проиндексированного изменения (значение по умолчанию), а затем еще раз проверила состояние. Потом последовательно проиндексировала второе изменение и закомитила. (рис. 4.10, 4.11, 4.12, 4.13, 4.14).

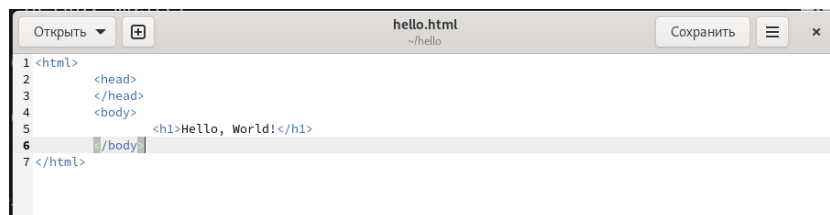


```

1 <html>
2   <body>
3     <h1>Hello, World!</h1>
4   </body>
5 </html>

```

Рис. 4.10: Код файла



```

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

```

Рис. 4.11: Продолжение кода



```

taisiya@tsganina:~/hello$ gedit hello.html
taisiya@tsganina:~/hello$ git add hello.html
taisiya@tsganina:~/hello$ gedit hello.html
taisiya@tsganina:~/hello$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)

    изменено:      hello.html

Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)

    изменено:      hello.html

taisiya@tsganina:~/hello$

```

Рис. 4.12: Одно изменение проиндексировано, второе нет

```

taisiya@tsganina:~/hello$ git commit -m "Added standard HTML page tags"
[master e75a8df] Added standard HTML page tags
 1 file changed, 5 insertions(+), 1 deletion(-)
taisiya@tsganina:~/hello$ git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)

    изменено:      hello.html

индекс пуст (используйте «git add» и/или «git commit -a»)
taisiya@tsganina:~/hello$

```

Рис. 4.13: Закоммитила проиндексированное

```

taisiya@tsganina:~/hello$ git add .
taisiya@tsganina:~/hello$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)

    изменено:      hello.html

taisiya@tsganina:~/hello$

```

Рис. 4.14: Проиндексировала второе изменение

Далее была работа с историей и просмотр версий. (рис. 4.15, 4.16).

```
taisiya@tsganina:~/hello$ git commit -m "Added HTML header"
[master 0b677b8] Added HTML header
 1 file changed, 2 insertions(+)
taisiya@tsganina:~/hello$ git log
commit 0b677b80f537736cbb73fef8c47e844e8bd7c565 (HEAD -> master)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:10:13 2025 +0300

    Added HTML header

commit e75a8df68e315c7312ae7d12304b8a045654f58b
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:07:44 2025 +0300

    Added standard HTML page tags

commit d4e86f696c9dd1b0f4db31b4242b0296ad278480
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:02:19 2025 +0300

    Added h1 tag

commit 64a83a23934628601807fa3be276b9ad3ac9a0cf
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 13:54:58 2025 +0300

    Initial Commit
taisiya@tsganina:~/hello$
```

Рис. 4.15: git log

```

taisiya@tsganina:~/hello$ git checkout 64a83a23934628601807fa3be276
b9ad3ac9a0cf
Примечание: переключение на «64a83a23934628601807fa3be276b9ad3ac9a0
cf».

Вы сейчас в состоянии «отсоединённого указателя HEAD». Можете осмот
реться,
внести экспериментальные изменения и зафиксировать их, также можете
отменить любые коммиты, созданные в этом состоянии, не затрагивая д
ругие
ветки, переключившись обратно на любую ветку.

Если хотите создать новую ветку для сохранения созданных коммитов,
можете
сделать это (сейчас или позже), используя команду switch с параметр
ом -с.
Например:

    git switch -с <новая-ветка>

Или отмените эту операцию с помощью:

    git switch -

Отключите этот совет, установив переменную конфигурации
advice.detachedHead в значение false

HEAD сейчас на 64a83a2 Initial Commit
taisiya@tsganina:~/hello$ cat hello.html
Hello, World!
taisiya@tsganina:~/hello$ █

```

Рис. 4.16: Первый коммит

5. Работа с тегами версий. Вместо поиска до хэш, использовала ^, обозначаю-  
щее «родитель v1». Вместо обозначения v1^ можно использовать v1~1. Это  
обозначение можно определить как «первую версию предшествующую v1».  
Создала тег первой версии и также тег для предыдущей версии при помощи  
команд:

```

git tag v1
git checkout v1^
cat hello.html
git tag v1-beta

```

Потом переключалась между двумя отмеченными версиями и смотрела теги в  
логе. (рис. 4.17, 4.18, 4.19).

```

taisiya@tsganina:~/hello$ git tag v1
taisiya@tsganina:~/hello$ git checkout v1^
cat hello.html
Примечание: переключение на «v1^».

Вы сейчас в состоянии «отсоединённого указателя HEAD». Можете осмотреться,
внести экспериментальные изменения и зафиксировать их, также можете
отменить любые коммиты, созданные в этом состоянии, не затрагивая другие
ветки, переключившись обратно на любую ветку.

Если хотите создать новую ветку для сохранения созданных коммитов, можете
сделать это (сейчас или позже), используя команду switch с параметром -с.
Например:

    git switch -с <новая-ветка>

Или отмените эту операцию с помощью:

    git switch -

Отключите этот совет, установив переменную конфигурации
advice.detachedHead в значение false

HEAD сейчас на e75a8df Added standard HTML page tags
<html>
    <body>
        <h1>Hello, World!</h1>
    </body>
</html>
taisiya@tsganina:~/hello$

```

Рис. 4.17: git tag v1

```

taisiya@tsganina:~/hello$ git tag v1-beta
taisiya@tsganina:~/hello$ git checkout v1
Предыдущая позиция HEAD была e75a8df Added standard HTML page tags
HEAD сейчас на 0b677b8 Added HTML header
taisiya@tsganina:~/hello$ git checkout v1-beta
Предыдущая позиция HEAD была 0b677b8 Added HTML header
HEAD сейчас на e75a8df Added standard HTML page tags
taisiya@tsganina:~/hello$

```

Рис. 4.18: git tag v1-beta

```

taisiya@tsganina:~/hello$ git tag
v1
v1-beta
taisiya@tsganina:~/hello$ git log master --all
commit 0b677b80f537736cbb73fef8c47e844e8bd7c565 (tag: v1, master)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:10:13 2025 +0300

    Added HTML header

commit e75a8df68e315c7312ae7d12304b8a045654f58b (HEAD, tag: v1-beta)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:07:44 2025 +0300

    Added standard HTML page tags

commit d4e86f696c9dd1b0f4db31b4242b0296ad278480
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:02:19 2025 +0300

    Added h1 tag

commit 64a83a23934628601807fa3be276b9ad3ac9a0cf
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 13:54:58 2025 +0300

    Initial Commit
taisiya@tsganina:~/hello$

```

Рис. 4.19: git log master --all

6. Отменить локальные изменения (до индексации). Убедилась, что нахожусь на последнем коммите ветки master, прежде чем продолжить работу. Потом внесла нежелательный комментарий, но не индексировала изменения и отменила их при помощи git checkout (рис. 4.20)

```

taisiya@tsganina:~/hello$ gedit hello.html
taisiya@tsganina:~/hello$ git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)

    изменено:      hello.html

индекс пуст (используйте «git add» и/или «git commit -a»)
taisiya@tsganina:~/hello$ git checkout hello.html
Updated 1 path from the index
taisiya@tsganina:~/hello$ git status
Текущая ветка: master
нечего коммитить, нет изменений в рабочем каталоге
taisiya@tsganina:~/hello$ cat hello.html
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
taisiya@tsganina:~/hello$

```

Рис. 4.20: Отменить локальные изменения (до индексации)

7. Отменить проиндексированные изменения (перед коммитом). Я опять внесла нежелательные изменения в файл, но теперь уже проиндексировала их, а после – отменила. (рис. 4.21).

```
taisiya@tsganina:~/hello$ gedit hello.html
taisiya@tsganina:~/hello$ git add hello.html
taisiya@tsganina:~/hello$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:      hello.html

taisiya@tsganina:~/hello$ git reset HEAD hello.html
Непроиндексированные изменения после сброса:
M       hello.html
taisiya@tsganina:~/hello$ git checkout hello.html
Updated 1 path from the index
taisiya@tsganina:~/hello$ git status
Текущая ветка: master
ничего коммитить, нет изменений в рабочем каталоге
taisiya@tsganina:~/hello$
```

Рис. 4.21: Отменить проиндексированные изменения (перед коммитом)

8. Отменить коммит. (рис. 4.22, 4.23, 4.24).

```
taisiya@tsganina:~/hello$ gedit hello.html
taisiya@tsganina:~/hello$ git add hello.html
git commit -m "Oops, we didn't want this commit"
[master ccae291] Oops, we didn't want this commit
1 file changed, 1 insertion(+)
taisiya@tsganina:~/hello$ git status
Текущая ветка: master
ничего коммитить, нет изменений в рабочем каталоге
taisiya@tsganina:~/hello$
```

Рис. 4.22: Делаем коммит

```
taisiya@tsganina:~/hello$ git revert HEAD
[master 3d304be] Revert "Oops, we didn't want this commit"
1 file changed, 1 deletion(-)
taisiya@tsganina:~/hello$
```

Рис. 4.23: git revert HEAD

```
taisiya@vbox:~/hello$ git log
commit 3d304be49cf7cb59b172f296c9f0367828ca06c4 (HEAD -> master)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:29:05 2025 +0300

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

    This reverts commit ccae291c2f2efa564e22e6b4852ee49178ad3e13.

commit ccae291c2f2efa564e22e6b4852ee49178ad3e13
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:25:28 2025 +0300

    Oops, we didn't want this commit

commit 0b677b80f537736cbb73fef8c47e844e8bd7c565 (tag: v1)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:10:13 2025 +0300

    Added HTML header

commit e75a8df68e315c7312ae7d12304b8a045654f58b (tag: v1-beta)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:07:44 2025 +0300

    Added standard HTML page tags

commit d4e86f696c9dd1b0f4db31b4242b0296ad278480
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:02:19 2025 +0300

    Added h1 tag

commit 64a83a23934628601807fa3be276b9ad3ac9a0cf
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 13:54:58 2025 +0300

    Initial Commit
taisiya@tsganina:~/hello$
```

Рис. 4.24: git log

9. Удалить коммиты из ветки. Часто мы делаем коммит, и сразу понимаем, что это была ошибка. Было бы неплохо иметь команду «возврата», которая позволила бы нам сделать вид, что неправильного коммита никогда и не было. Команда «возврата» даже предотвратила бы появление нежелательного коммита в истории git log. (рис. 4.25, 4.26)

```

taisiya@tsganina:~/hello$ git tag oops
taisiya@tsganina:~/hello$ git reset --hard v1
git log
Указатель HEAD сейчас на коммите 0b677b8 Added HTML header
commit 0b677b80f537736cbb73fef8c47e844e8bd7c565 (HEAD -> master, tag: v1)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:10:13 2025 +0300

    Added HTML header

commit e75a8df68e315c7312ae7d12304b8a045654f58b (tag: v1-beta)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:07:44 2025 +0300

    Added standard HTML page tags

commit d4e86f696c9dd1b0f4db31b4242b0296ad278480
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:02:19 2025 +0300

    Added h1 tag

commit 64a83a23934628601807fa3be276b9ad3ac9a0cf
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 13:54:58 2025 +0300

    Initial Commit
taisiya@tsganina:~/hello$

```

Рис. 4.25: Давайте отметим последний коммит тегом, и выполним сброс коммитов к предшествующим коммиту oops



```
taisiya@vbox:~/hello
taisiya@tsganina:~/hello$ git log --all
commit 3d304be49cf7cb59b172f296c9f0367828ca06c4 (tag: oops)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:29:05 2025 +0300

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

    This reverts commit ccae291c2f2efa564e22e6b4852ee49178ad3e13.

commit ccae291c2f2efa564e22e6b4852ee49178ad3e13
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:25:28 2025 +0300

    Oops, we didn't want this commit

commit 0b677b80f537736cbb73fef8c47e844e8bd7c565 (HEAD -> master, tag: v1)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:10:13 2025 +0300

    Added HTML header

commit e75a8df68e315c7312ae7d12304b8a045654f58b (tag: v1-beta)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:07:44 2025 +0300

    Added standard HTML page tags

commit d4e86f696c9dd1b0f4db31b4242b0296ad278480
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:02:19 2025 +0300

    Added h1 tag

commit 64a83a23934628601807fa3be276b9ad3ac9a0cf
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 13:54:58 2025 +0300

    Initial Commit
taisiya@tsganina:~/hello$
```

Рис. 4.26: Оказывается, что коммиты все еще находятся в репозитории

10. Удаление тега. Сброс в локальных ветках, как правило, безопасен. Последствия любой «аварии» как правило, можно восстановить простым сбросом с помощью нужного коммита. Однако, если ветка «расшарена» на удаленных репозиториях, сброс может сбить с толку других пользователей ветки. (рис. 4.27).

```

taisiya@tsganina:~/hello$ git tag -d oops
git log --all
Метка «oops» удалена (была 3d304be)
commit 0b677b80f537736cbb73fef8c47e844e8bd7c565 (HEAD -> master, tag: v1)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:10:13 2025 +0300

    Added HTML header

commit e75a8df68e315c7312ae7d12304b8a045654f58b (tag: v1-beta)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:07:44 2025 +0300

    Added standard HTML page tags

commit d4e86f696c9dd1b0f4db31b4242b0296ad278480
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:02:19 2025 +0300

    Added h1 tag

commit 64a83a23934628601807fa3be276b9ad3ac9a0cf
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 13:54:58 2025 +0300

    Initial Commit
taisiya@tsganina:~/hello$

```

Рис. 4.27: Удаление тега oops

11. Внести изменения в коммиты. Добавим в страницу комментариев автора. После совершения коммита мы понимаем, что любой хороший комментарий должен включать электронную почту автора. Обновим страницу hello, включив в нее email. Давайте изменим предыдущий коммит, включив в него адрес электронной почты. (рис. 4.28, 4.29)

```

taisiya@tsganina:~/hello$ gedit hello.html
taisiya@tsganina:~/hello$ git add hello.html
git commit -m "Add an author comment"
[master 60c1ccd] Add an author comment
1 file changed, 1 insertion(+)
taisiya@tsganina:~/hello$ cat hello.html
<!-- Author: Taisia S. Ganina -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
taisiya@tsganina:~/hello$

```

Рис. 4.28: Добавим в страницу комментарий автора

```

taisiya@tsganina:~/hello$ gedit hello.html
taisiya@tsganina:~/hello$ git add hello.html
git commit --amend -m "Add an author/email comment"
[master fbc03d5] Add an author/email comment
Date: Sat Feb 15 14:34:47 2025 +0300
1 file changed, 1 insertion(+)
taisiya@tsganina:~/hello$ git log
commit fbc03d56d4f59a0c988142541f8a36667f5837b0 (HEAD -> master)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:34:47 2025 +0300

    Add an author/email comment

commit 0b677b80f537736cbb73fef8c47e844e8bd7c565 (tag: v1)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:10:13 2025 +0300

    Added HTML header

commit e75a8df68e315c7312ae7d12304b8a045654f58b (tag: v1-beta)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:07:44 2025 +0300

    Added standard HTML page tags

commit d4e86f696c9dd1b0f4db31b4242b0296ad278480
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:02:19 2025 +0300

    Added h1 tag

commit 64a83a23934628601807fa3be276b9ad3ac9a0cf
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 13:54:58 2025 +0300

    Initial Commit
taisiya@tsganina:~/hello$ █

```

Рис. 4.29: Изменим предыдущий коммит

12. Переместить файлы. Сейчас мы собираемся создать структуру нашего репозитория. Давайте перенесем страницу в каталог `lib`, сделаем коммит этого перемещения. Добавим файл `index.html` в наш репозиторий. Теперь при открытии `index.html`, мы должны увидеть кусок страницы `hello` в маленьком окошке. (рис. 4.30, 4.31, 4.32)

```

taisiya@tsganina:~/hello$ mkdir lib
git mv hello.html lib
git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    переименовано: hello.html -> lib/hello.html

taisiya@tsganina:~/hello$ git commit -m "Moved hello.html to lib"
[master 7df9605] Moved hello.html to lib
 1 file changed, 0 insertions(+), 0 deletions(-)
  rename hello.html => lib/hello.html (100%)
taisiya@tsganina:~/hello$ git status
Текущая ветка: master
ничего коммитить, нет изменений в рабочем каталоге
taisiya@tsganina:~/hello$

```

Рис. 4.30: Структура нашего репозитория

```

taisiya@tsganina:~/hello$ touch index.html
taisiya@tsganina:~/hello$ nano index.html
taisiya@tsganina:~/hello$ cat index.html
<html>
  <body>
    <iframe src="lib/hello.html" width="200" height="200" />
  </body>
</html>
taisiya@tsganina:~/hello$ git add index.html
git commit -m "Added index.html."
[master fe9904d] Added index.html.
 1 file changed, 5 insertions(+)
 create mode 100644 index.html
taisiya@tsganina:~/hello$

```

Рис. 4.31: Добавим файл index.html

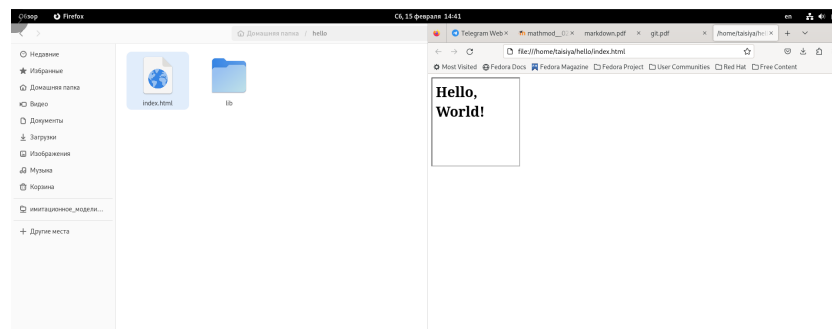


Рис. 4.32: Кусок страницы hello в маленьком окошке

13. Рассмотреть Git внутри: Каталог .git. Посмотрим каталог, в котором хранится вся информация git.

Выполним:

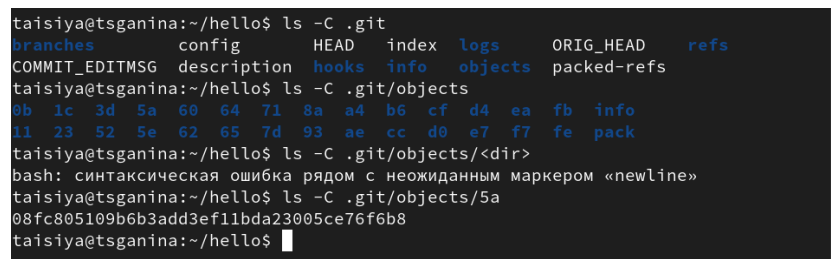
```
ls -C .git/objects
```

Мы должны увидеть набор каталогов, имена которых состоят из 2 символов. Имена каталогов являются первыми двумя буквами хэша sha1 объекта, хранящегося в git. После углубимся в базу данных объектов, посмотрим файл конфигурации, создающийся для каждого конкретного проекта.

Выполним:

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

Мы узнаём 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. Посмотрим файл HEAD содержит ссылку на текущую ветку, в данный момент это должна быть ветка `master`. (рис. 4.33, 4.34, 4.35).



```
taisiya@tsganina:~/hello$ ls -C .git
branches      config        HEAD         index        logs         ORIG_HEAD    refs
COMMIT_EDITMSG description  hooks        info         objects      packed-refs
taisiya@tsganina:~/hello$ ls -C .git/objects
0b 1c 3d 5a 60 64 71 8a a4 b6 cf d4 ea fb info
11 23 52 5e 62 65 7d 93 ae cc d0 e7 f7 fe pack
taisiya@tsganina:~/hello$ ls -C .git/objects/<dir>
bash: синтаксическая ошибка рядом с неожиданным маркером «newline»
taisiya@tsganina:~/hello$ ls -C .git/objects/5a
08fc805109b6b3add3ef11bda23005ce76f6b8
taisiya@tsganina:~/hello$
```

Рис. 4.33: Набор каталогов, имена которых состоят из 2 символов

```

taisiya@tsganina:~/hello$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
taisiya@tsganina:~/hello$ ls .git/refs
heads tags
taisiya@tsganina:~/hello$ ls .git/refs/heads
master
taisiya@tsganina:~/hello$ ls .git/refs/tags
v1 v1-beta
taisiya@tsganina:~/hello$ cat .git/refs/tags/v1
0b677b80f537736cbb73fef8c47e844e8bd7c565
taisiya@tsganina:~/hello$

```

Рис. 4.34: Посмотрим файл конфигурации

```

taisiya@tsganina:~/hello$ cat .git/HEAD
ref: refs/heads/master
taisiya@tsganina:~/hello$ git log --max-count=1
commit fe9904d006e78306b1c23eb37f9433e5de3af3f9 (HEAD -> master)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:41:10 2025 +0300

    Added index.html.
taisiya@tsganina:~/hello$ git cat-file -t fe9904d
commit
taisiya@tsganina:~/hello$ git cat-file -p fe9904d
tree 711ac7a9245185717a3e100fe149fc6bf821b4eb
parent 7df96059d482c6cb77042f300441f95af22b6a85
author tsganina <tai.sergeevna@yandex.com> 1739619670 +0300
committer tsganina <tai.sergeevna@yandex.com> 1739619670 +0300

    Added index.html.
taisiya@tsganina:~/hello$

```

Рис. 4.35: Файл HEAD

14. Работа непосредственно с объектами git. Посмотрим последний коммит в репозиторий. SHA1 хэш в нашей системе. Мы можем вывести дерево каталогов, ссылка на который идет в коммите. Это должно быть описание файлов (верхнего уровня) в нашем проекте (для конкретного коммита). Вывод каталога lib, вывод файла hello.html.(рис.@fig:036)

```

taisiya@tsganina:~/hello$ git log --max-count=1
commit fe9904d006e78306b1c23eb37f9433e5de3af3f9 (HEAD -> master)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:41:10 2025 +0300

    Added index.html.
taisiya@tsganina:~/hello$ git cat-file -t fe9904d
commit
taisiya@tsganina:~/hello$ git cat-file -p fe9904d
tree 711ac7a9245185717a3e100fe149fc6bf821b4eb
parent 7df96059d482c6cb77042f300441f95af22b6a85
author tsganina <tai.sergeevna@yandex.com> 1739619670 +0300
committer tsganina <tai.sergeevna@yandex.com> 1739619670 +0300

Added index.html.
taisiya@tsganina:~/hello$ git cat-file -p 711ac7a
100644 blob cf5bc0ea78289c0be141b578ca407f1159bd25bc    index.html
040000 tree aeeae7cd371d664fb2afee8f60e3a2db98f053ce    lib
taisiya@tsganina:~/hello$ git cat-file -p aeeae7c
100644 blob b69ab178cf5175103dfb1b4b98b9bd80031119a0    hello.html
taisiya@tsganina:~/hello$ git cat-file -p b69ab17
<!-- Author: Taisia S. Ganina (1132226429@pfur.ru) -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
taisiya@tsganina:~/hello$

```

Рис. 4.36: Работа непосредственно с объектами git

**Самостоятельное исследование.** Исследуем git репозиторий вручную самостоятельно. Смотрим, удастся ли нам найти оригинальный файл hello.html с самого первого коммита вручную по ссылкам SHA1 хэша в последнем коммите. (рис. 4.37, 4.38)

```

commit 64a83a23934628601807fa3be276b9ad3ac9a0cf
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 13:54:58 2025 +0300

    Initial Commit
taisiya@tsganina:~/hello$ git cat-file -t 64a83a23934628601807fa3be276b9ad3ac9a0cf
commit
taisiya@tsganina:~/hello$ git cat-file -p 64a83a23934628601807fa3be276b9ad3ac9a0cf
tree 523cd318fc3b74ad3852cd34a74720ad6f7dd422
author tsganina <tai.sergeevna@yandex.com> 1739616898 +0300
committer tsganina <tai.sergeevna@yandex.com> 1739616898 +0300

Initial Commit
taisiya@tsganina:~/hello$ git cat-file -p 523cd31
100644 blob 8ab686eafeb1f44702738c8b0f24f2567c36da6d    hello.html
taisiya@tsganina:~/hello$ git cat-file -p 8ab686e
Hello, World!
taisiya@tsganina:~/hello$

```

Рис. 4.37: Самостоятельное исследование 1

```

taisiya@tsganina:~/hello$ git cat-file -p 7df9605
tree 5e5438c90e1c97ee536e953fa46ee712046cfdbe
parent fbc03d56d4f59a0c988142541f8a36667f5837b0
author tsganina <ta1.sergeevna@yandex.com> 1739619537 +0300
committer tsganina <ta1.sergeevna@yandex.com> 1739619537 +0300

Moved hello.html to lib
taisiya@tsganina:~/hello$ git cat-file -p fbc03d5
tree aeae7cd371d664fb2afee8f60e3a2db98f953ce
parent 0b677b80f537736cbb73fef8c47e844e8bd7c565
author tsganina <ta1.sergeevna@yandex.com> 1739619287 +0300
committer tsganina <ta1.sergeevna@yandex.com> 1739619389 +0300

Add an author/email comment
taisiya@tsganina:~/hello$ git cat-file -p 0b677b8
tree a4d5a7e4cc555b8041bcd34ad3692e96e3f266e
parent e75a8df68e315c7312ae7d12304b8a045654f58b
author tsganina <ta1.sergeevna@yandex.com> 1739617813 +0300
committer tsganina <ta1.sergeevna@yandex.com> 1739617813 +0300

Added HTML header
taisiya@tsganina:~/hello$ git cat-file -p e75a8df
tree 5a08fc805109b6b3add3ef11bda23005ce76f6b8
parent d4e86f696c9dd1b0f4db31b4242b0296ad278480
author tsganina <ta1.sergeevna@yandex.com> 1739617664 +0300
committer tsganina <ta1.sergeevna@yandex.com> 1739617664 +0300

Added standard HTML page tags
taisiya@tsganina:~/hello$ git cat-file -p d4e86f6
tree 629377ed8eb96afed9fd6cb7b48b1e4a1d3dc56
parent 64a83a23934628601807fa3be276b9ad3ac9a0cf
author tsganina <ta1.sergeevna@yandex.com> 1739617339 +0300
committer tsganina <ta1.sergeevna@yandex.com> 1739617339 +0300

Added h1 tag
taisiya@tsganina:~/hello$ git cat-file -p 64a83a2
tree 523cd318fc3b74ad3852cd34a74720ad6f7dd422
author tsganina <ta1.sergeevna@yandex.com> 1739616898 +0300
committer tsganina <ta1.sergeevna@yandex.com> 1739616898 +0300

Initial Commit
taisiya@tsganina:~/hello$ git cat-file -p 523cd31
100644 blob 8ab686eafeb1f44702738c8b0f24f2567c36da6d    hello.html
taisiya@tsganina:~/hello$ git cat-file -p 8ab686e
Hello, World!
taisiya@tsganina:~/hello$

```

Рис. 4.38: Самостоятельное исследование 2

## 15. Навигация по веткам. Изменения в ветке master.

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. Давайте назовем нашу новую ветку «style». Добавим файл стилей style.css. Обновим файл hello.html, чтобы использовать стили style.css. Обновим файл index.html, чтобы он тоже использовал style.css. (рис. 4.39, 4.40).



```

taisiya@tsganina:~/hello$ git checkout -b style
git status
Переключились на новую ветку «style»
Текущая ветка: style
нечего коммитить, нет изменений в рабочем каталоге
taisiya@tsganina:~/hello$
taisiya@tsganina:~/hello$
taisiya@tsganina:~/hello$ touch lib/style.css
taisiya@tsganina:~/hello$ vim lib/style.css
taisiya@tsganina:~/hello$ cat lib/style.css
hl {
color: red;
}
taisiya@tsganina:~/hello$ git add lib/style.css
git commit -m "Added css stylesheet"
[style 17188de] Added css stylesheet
1 file changed, 3 insertions(+)
create mode 100644 lib/style.css
taisiya@tsganina:~/hello$

```

Рис. 4.39: Создайте ветку, файл стилей style.css

```

taisiya@tsganina:~/hello$ touch lib/style.css
taisiya@tsganina:~/hello$ vim lib/style.css
taisiya@tsganina:~/hello$ cat lib/style.css
hl {
color: red;
}
taisiya@tsganina:~/hello$ git add lib/style.css
git commit -m "Added css stylesheet"
[style 17188de] Added css stylesheet
1 file changed, 3 insertions(+)
create mode 100644 lib/style.css
taisiya@tsganina:~/hello$ vim lib/hello.html
taisiya@tsganina:~/hello$ cat lib/hello.html
<!-- Author: Taisia S. Ganina (1132226429@pfur.ru) -->
<html>
<head>
<link type="text/css" rel="stylesheet"
media="all" href="style.css" />
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>
taisiya@tsganina:~/hello$ git add lib/hello.html
git commit -m "Hello uses style.css"
[style 2f480ed] Hello uses style.css
1 file changed, 2 insertions(+)
taisiya@tsganina:~/hello$ vim index.html
taisiya@tsganina:~/hello$ cat index.html
cat: index.html: Нет такого файла или каталога
taisiya@tsganina:~/hello$ cat index.html
<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>
taisiya@tsganina:~/hello$ git add index.html
git commit -m "Updated index.html"
[style 5c82e97] Updated index.html
1 file changed, 4 insertions(+)
taisiya@tsganina:~/hello$

```

Рис. 4.40: Обновим файл hello.html. Обновим файл index.html

Потом используем команду `git checkout` для переключения между ветками. (рис. 4.41).

```
taisiya@tsganina:~/hello$ git checkout master
cat lib/hello.html
Переключились на ветку «master»
<!-- Author: Taisia S. Ganina (1132226429@pfur.ru) -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
taisiya@tsganina:~/hello$ git checkout style
cat lib/hello.html
Переключились на ветку «style»
<!-- Author: Taisia S. Ganina (1132226429@pfur.ru) -->
<html>
  <head>
    <link type="text/css" rel="stylesheet"
media="all" href="style.css" />
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
taisiya@tsganina:~/hello$
```

Рис. 4.41: Переключения между ветками

Создадим файл README в ветке master. Сделаем коммит изменений README.md в ветку master. Просмотр отличающихся веток. (рис. 4.42).

```
taisiya@vbox:~/hello — git log --graph --all
* commit 3e577d603c22c1b2d0ea9b5eb54ba4898aa44e46 (HEAD -> master)
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:15:13 2025 +0300
|
| Added README
|
* commit 5c82e97b2eaba243a7317eaf809270d6461615a5 (style)
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:11:56 2025 +0300
|
| Updated index.html
|
* commit 2f480ed46785944a825fbf87d7a11d693b6ed37d
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:10:01 2025 +0300
|
| Hello uses style.css
|
* commit 17188de946847f19929432e1207dd610d322a081
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:07:51 2025 +0300
|
| Added css stylesheet
|
* commit fe9904d006e78306b1c23eb37f9433e5de3af3f9
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 14:41:10 2025 +0300
|
| Added index.html.
|
* commit 7df96059d482c6cb77042f300441f95af22b6a85
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 14:38:57 2025 +0300
|
| Moved hello.html to lib
|
* commit fbc03d56d4f59a0c988142541f8a36667f5837b0
| Author: tsganina <tai.sergeevna@yandex.com>
```

Рис. 4.42: git log –graph –all

## 16. Слияние веток. Создание конфликта.

Слияние переносит изменения из двух веток в одну. Давайте вернемся к ветке style и сольем master с style. Путем периодического слияния ветки master с веткой style мы можем переносить из master любые изменения и поддерживать совместимость изменений style с изменениями в основной ветке. Но что если изменения в ветке master конфликтуют с изменениями в style? (рис. 4.43, 4.44, 4.45).

```
lasiya@vbox:~/hello — git log --graph --all
* commit 0bcf53f4ba31822402fa0ee975dc017bd9ce077a (HEAD -> style)
| Merge: 5c82e97 3e577d6
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:17:09 2025 +0300
|
| Merge branch 'master' into style
|
* commit 3e577d603c22c1b2d0ea9b5eb54ba4898aa44e46 (master)
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:15:13 2025 +0300
|
| Added README
|
* commit 5c82e97b2eaba243a7317eaf809270d6461615a5
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:11:56 2025 +0300
|
| Updated index.html
|
* commit 2f480ed46785944a825fbf87d7a11d693b6ed37d
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:10:01 2025 +0300
|
| Hello uses style.css
|
* commit 17188de946847f19929432e1207dd610d322a081
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:07:51 2025 +0300
|
| Added css stylesheet
|
* commit fe9904d006e78306b1c23eb37f9433e5de3af3f9
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 14:41:10 2025 +0300
|
| Added index.html.
|
* commit 7df96059d482c6cb77042f300441f95af22b6a85
|
```

Рис. 4.43: Слияние веток

```

* commit 5ba398e706ac7754f90349ff7a81aef4c5a7be2c (HEAD -> master)
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:20:37 2025 +0300
|
| Life is great
|
* commit 0bcf53f4ba31822402fa0ee975dc017bd9ce077a (style)
| \ Merge: 5c82e97 3e577d6
| / Author: tsganina <tai.sergeevna@yandex.com>
| / Date: Sat Feb 15 15:17:09 2025 +0300
|
| Merge branch 'master' into style
|
* commit 3e577d603c22c1b2d0ea9b5eb54ba4898aa44e46
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:15:13 2025 +0300
|
| Added README
|
* commit 5c82e97b2eaba243a7317eaf809270d6461615a5
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:11:56 2025 +0300
|
| Updated index.html
|
* commit 2f480ed46785944a825fbf87d7a11d693b6ed37d
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:10:01 2025 +0300
|
| Hello uses style.css
|
* commit 17188de946847f19929432e1207dd610d322a081
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:07:51 2025 +0300
|
| Added css stylesheet
|
* commit fe9904d006e78306b1c23eb37f9433e5de3af3f9
|

```

Рис. 4.44: Создание конфликта

```

taisiya@tsganina:~/hello$ git checkout style
git merge master
Переключились на ветку «style»
Автослияние lib/hello.html
КОНФЛИКТ (содержимое): Конфликт слияния в lib/hello.html
Сбой автоматического слияния; исправьте конфликты, затем зафиксируйте результа
т.
taisiya@tsganina:~/hello$ cat ib/hello.html
cat: ib/hello.html: Нет такого файла или каталога
taisiya@tsganina:~/hello$ cat lib/hello.html
<!-- Author: Taisia S. Ganina (1132226429@pfur.ru) -->
<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>
taisiya@tsganina:~/hello$

```

Рис. 4.45: Вернемся к ветке style и попытаемся объединить ее с новой веткой master

## 17. Разрешение конфликтов. (рис. 4.46).

```

*   commit c1db7ef65458c760e1e5e067ad191014cc4c072a (HEAD -> style)
|   Merge: 0bcf53f 5ba398e
|   Author: tsganina <tai.sergeevna@yandex.com>
|   Date:   Sat Feb 15 15:24:55 2025 +0300
|
|       Merged master fixed conflict.
|
*   commit 5ba398e706ac7754f90349ff7a81aef4c5a7be2c (master)
|   Author: tsganina <tai.sergeevna@yandex.com>
|   Date:   Sat Feb 15 15:20:37 2025 +0300
|
|       Life is great
|
*   commit 0bcf53f4ba31822402fa0ee975dc017bd9ce077a
|   Merge: 5c82e97 3e577d6
|   Author: tsganina <tai.sergeevna@yandex.com>
|   Date:   Sat Feb 15 15:17:09 2025 +0300
|
|       Merge branch 'master' into style
|
*   commit 3e577d603c22c1b2d0ea9b5eb54ba4898aa44e46
|   Author: tsganina <tai.sergeevna@yandex.com>
|   Date:   Sat Feb 15 15:15:13 2025 +0300
|
|       Added README
|
*   commit 5c82e97b2eaba243a7317eaf809270d6461615a5
|   Author: tsganina <tai.sergeevna@yandex.com>
|   Date:   Sat Feb 15 15:11:56 2025 +0300
|
|       Updated index.html
|
*   commit 2f480ed46785944a825fbf87d7a11d693b6ed37d
|   Author: tsganina <tai.sergeevna@yandex.com>
|   Date:   Sat Feb 15 15:10:01 2025 +0300
|
|       Hello uses style.css
|
:

```

Рис. 4.46: Внесём изменения в lib/hello.html

## 18. Сброс ветки style. Сброс ветки master.

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.(рис. 4.47, 4.48)

```

taisiya@tsganina:~/hello$ git reset --hard 5c82e97
Указатель HEAD сейчас на коммите 5c82e97 Updated index.html
taisiya@tsganina:~/hello$ git log --graph --all
* commit 5ba398e706ac7754f90349ff7a81aef4c5a7be2c (master)
  | Author: tsganina <tai.sergeevna@yandex.com>
  | Date: Sat Feb 15 15:20:37 2025 +0300
  |
  | Life is great
  |
* commit 3e577d603c22c1b2d0ea9b5eb54ba4898aa44e46
  | Author: tsganina <tai.sergeevna@yandex.com>
  | Date: Sat Feb 15 15:15:13 2025 +0300
  |
  | Added README
  |
* commit 5c82e97b2eaba243a7317eaf809270d6461615a5 (HEAD -> style)
  | Author: tsganina <tai.sergeevna@yandex.com>
  | Date: Sat Feb 15 15:11:56 2025 +0300
  |
  | Updated index.html
  |
* commit 2f480ed46785944a825fbf87d7a11d693b6ed37d
  | Author: tsganina <tai.sergeevna@yandex.com>
  | Date: Sat Feb 15 15:10:01 2025 +0300
  |
  | Hello uses style.css
  |
* commit 17188de946847f19929432e1207dd610d322a081
  | Author: tsganina <tai.sergeevna@yandex.com>
  | Date: Sat Feb 15 15:07:51 2025 +0300
  |
  | Added css stylesheet
  |
* commit fe9904d006e78306b1c23eb37f9433e5de3af3f9
  | Author: tsganina <tai.sergeevna@yandex.com>
  | Date: Sat Feb 15 14:41:10 2025 +0300
  |
  | Added index.html.

```

Рис. 4.47: Сброс ветки style



```
taisiya@vbox:~/hello — git log --graph --all
| Author: tsganina <tai.sergeevna@yandex.com>
taisiya@tsganina:~/hello$ git reset --hard 3e577d6
Указатель HEAD сейчас на коммите 3e577d6 Added README
taisiya@tsganina:~/hello$ git log --graph --all
* commit 3e577d603c22c1b2d0ea9b5eb54ba4898aa44e46 (HEAD -> master)
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:15:13 2025 +0300
|
| Added README
|
* commit 5c82e97b2eaba243a7317eaf809270d6461615a5 (style)
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:11:56 2025 +0300
|
| Updated index.html
|
* commit 2f480ed46785944a825fbf87d7a11d693b6ed37d
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:10:01 2025 +0300
|
| Hello uses style.css
|
* commit 17188de946847f19929432e1207dd610d322a081
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:07:51 2025 +0300
|
| Added css stylesheet
|
* commit fe9904d006e78306b1c23eb37f9433e5de3af3f9
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 14:41:10 2025 +0300
|
| Added index.html.
|
* commit 7df96059d482c6cb77042f300441f95af22b6a85
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 14:38:57 2025 +0300
|
| Moved hello.html to lib
```

Рис. 4.48: Сброс ветки master

## 19. 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. (рис. 4.49).

```

taisiya@tsganina:~/hello$ git checkout style
git rebase master
git log --graph
Переключились на ветку «style»
Успешно перемещён и обновлён refs/heads/style.
* commit 3a8f38d41447f3178f8572fe33f8e41453f408d9 (HEAD -> style)
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:11:56 2025 +0300
|
| Updated index.html
|
* commit d1cdb5c5b66cda1bb8b2ed8c6b02cb7fbf33c9cf
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:10:01 2025 +0300
|
| Hello uses style.css
|
* commit 393a90e1024ddeb6b4a4f790caf34ad234f15094
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:07:51 2025 +0300
|
| Added css stylesheet
|
* commit 3e577d603c22c1b2d0ea9b5eb54ba4898aa44e46 (master)
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 15:15:13 2025 +0300
|
| Added README
|
* commit fe9904d006e78306b1c23eb37f9433e5de3af3f9
| Author: tsganina <tai.sergeevna@yandex.com>
| Date: Sat Feb 15 14:41:10 2025 +0300
|
| Added index.html.
|
* commit 7df96059d482c6cb77042f300441f95af22b6a85

```

Рис. 4.49: Сброс ветки master

Конечный результат перебазирувания 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.

## 20. Слияние в ветку master.

Мы поддерживали соответствие ветки style с веткой master (с помощью rebase), теперь давайте сольем изменения style в ветку master. (рис. 4.50).

```

taisiya@tsganina:~/hello$
taisiya@tsganina:~/hello$
taisiya@tsganina:~/hello$
taisiya@tsganina:~/hello$ git checkout master
git merge style
Переключились на ветку «master»
Обновление 3e577d6..3a8f38d
Fast-forward
 index.html      | 4 ++++
 lib/hello.html | 2 ++
 lib/style.css   | 3 +++
 3 files changed, 9 insertions(+)
 create mode 100644 lib/style.css
taisiya@tsganina:~/hello$ git log
commit 3a8f38d41447f3178f8572fe33f8e41453f408d9 (HEAD -> master, style)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 15:11:56 2025 +0300

    Updated index.html

commit d1cdb5c5b66cda1bb8b2ed8c6b02cb7fbf33c9cf
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 15:10:01 2025 +0300

    Hello uses style.css

commit 393a90e1024ddeb6b4a4f790caf34ad234f15094
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 15:07:51 2025 +0300

    Added css stylesheet

commit 3e577d603c22c1b2d0ea9b5eb54ba4898aa44e46
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 15:15:13 2025 +0300

    Added README

```

Рис. 4.50: Слияние в ветку master

21. Клонирование репозитория. Просмотр клонированного репозитория. (рис. 4.51, 4.52).

```

taisiya@tsganina:~/hello$ cd ..
pwd
ls
/home/taisiya
backup          Видео
bin             'Внешний курс инфобез'
hello          Документы
lab07.sh~      Загрузки
pandoc-2.19.2  Изображения
pandoc-2.19.2-linux-amd64.tar.gz Музыка
pandoc-crossref  Общедоступные
pandoc-crossref.1 'Рабочий стол'
pandoc-crossref-Linux.tar.xz Шаблоны
work
taisiya@tsganina:~$ git clone hello cloned_hello
ls
Клонирование в «cloned_hello»...
готово.
backup          work
bin             Видео
cloned_hello    'Внешний курс инфобез'
hello          Документы
lab07.sh~      Загрузки
pandoc-2.19.2  Изображения
pandoc-2.19.2-linux-amd64.tar.gz Музыка
pandoc-crossref  Общедоступные
pandoc-crossref.1 'Рабочий стол'
pandoc-crossref-Linux.tar.xz Шаблоны
taisiya@tsganina:~$

```

Рис. 4.51: Создадим клон репозитория hello

```

taisiya@tsganina:~/cloned_hello$ ls
index.html  lib  README.md
taisiya@tsganina:~/cloned_hello$

```

Рис. 4.52: Просмотр клонированного репозитория

## 22. Что такое origin? Удаленные ветки. Изменение оригинального репозитория.

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. (рис. 4.53).

```

taisiya@tsganina:~/cloned_hello$ git remote
origin
taisiya@tsganina:~/cloned_hello$ git remote show origin
* внешний репозиторий origin
  URL для извлечения: /home/taisiya/hello
  URL для отправки: /home/taisiya/hello
  HEAD ветка: master
  Внешние ветки:
    master отслеживается
    style отслеживается
  Локальная ветка, настроенная для «git pull»:
    master будет слита с внешней веткой master
  Локальная ссылка, настроенная для «git push»:
    master будет отправлена в master (уже актуальна)
taisiya@tsganina:~/cloned_hello$

```

Рис. 4.53: Более подробная информация об имени по умолчанию

## 23. Слияние извлеченных изменений. Научимся извлекать изменения из удаленного репозитория. (рис. 4.54, 4.55, 4.56, 4.57).

```

taisiya@tsganina:~/hello$ cd ../cloned_hello
git fetch
git log --all
commit 3a8f38d41447f3178f8572fe33f8e41453f408d9 (HEAD -> master, origin/style,
origin/master, origin/HEAD)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 15:11:56 2025 +0300

    Updated index.html

commit d1cdb5c5b66cda1bb8b2ed8c6b02cb7fbf33c9cf
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 15:10:01 2025 +0300

    Hello uses style.css

commit 393a90e1024ddeb6b4a4f790caf34ad234f15094
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 15:07:51 2025 +0300

    Added css stylesheet

commit 3e577d603c22c1b2d0ea9b5eb54ba4898aa44e46
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 15:15:13 2025 +0300

    Added README

commit fe9904d006e78306b1c23eb37f9433e5de3af3f9
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 14:41:10 2025 +0300

    Added index.html.

commit 7d50c0f0d402e6eb77042f200441f0f5e522b6e0f

```

Рис. 4.54: Научиться извлекать изменения из удаленного репозитория

```

taisiya@tsganina:~/cloned_hello$ cat README.md
This is the Hello World example from the git tutorial.
taisiya@tsganina:~/cloned_hello$ cd ../hello
taisiya@tsganina:~/hello$ cat README.md
This is the Hello World example from the git tutorial.
taisiya@tsganina:~/hello$ cd ../cloned_hello
taisiya@tsganina:~/cloned_hello$ git merge origin/master
Уже актуально.
taisiya@tsganina:~/cloned_hello$

```

Рис. 4.55: cat README.md

```

taisiya@tsganina:~/cloned_hello$ git branch --track style origin/style
git branch -a
git log --max-count=2
branch 'style' set up to track 'origin/style'.
* master
  style
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style
commit 3a8f38d41447f3178f8572fe33f8e41453f408d9 (HEAD -> master, origin/style,
origin/master, origin/HEAD, style)
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 15:11:56 2025 +0300

    Updated index.html

commit d1cdb5c5b66cda1bb8b2ed8c6b02cb7fbf33c9cf
Author: tsganina <tai.sergeevna@yandex.com>
Date: Sat Feb 15 15:10:01 2025 +0300

    Hello uses style.css
taisiya@tsganina:~/cloned_hello$

```

Рис. 4.56: Добавим локальную ветку, которая отслеживает удаленную ветку

```

taisiya@tsganina:~/hello$ git checkout master
git add README.md
git commit -m "Added shared comment to readme"
M README.md
Уже на «master»
[master 5c399d3] Added shared comment to readme
1 file changed, 1 insertion(+)
taisiya@tsganina:~/hello$ git push shared master
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 405 байтов | 405.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To ../hello.git
3a8f38d..5c399d3 master -> master
taisiya@tsganina:~/hello$

```

Рис. 4.57: Отправка изменений

## 24. Извлечение общих изменений. (рис. 4.58)

```

taisiya@tsganina:~/hello$ cd ../cloned_hello
taisiya@tsganina:~/cloned_hello$ git remote add shared ../hello.git
git branch --track shared master
git pull shared master
cat README.md
branch 'shared' set up to track 'master'.
remote: Перечисление объектов: 5, готово.
remote: Подсчет объектов: 100% (5/5), готово.
remote: Сжатие объектов: 100% (3/3), готово.
remote: Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно
использовано пакетов 0
Распаковка объектов: 100% (3/3), 385 байтов | 385.00 КиБ/с, готово.
Из ../hello
* branch          master      -> FETCH_HEAD
* [новая ветка]    master      -> shared/master
Обновление 3a8f38d..5c399d3
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
This is the Hello World example from the git tutorial.
(Changed in the original and pushed to shared)
taisiya@tsganina:~/cloned_hello$ █

```

Рис. 4.58: Извлечение общих изменений

## 25. Написание отчета по лабораторной работе в формате Markdown.

```

73 # Задание
74
75 1. Подготовить пространство для работы.
76 2. Создать проект.
77 3. Внести изменения в проект.
78 4. Провести индексацию изменений
79 5. Работа с тегами версий.
80 6. Отменить локальные изменения (до индексации).
81 7. Отменить проиндексированные изменения (перед коммитом).
82 8. Отменить коммит.
83 9. Удалить коммиты из ветки.
84 10. Удаление тега.
85 11. Внести изменения в коммиты.
86 12. Переместить файлы.
87 13. Рассмотреть git внутри: Каталог .git.
88 14. Работа непосредственно с объектами git.
89 15. Навигация по веткам. Изменения в ветке master.
90 16. Слияние веток. Создание конфликта.
91 17. Разрешение конфликтов.
92 18. Сброс ветки style. Сброс ветки master.
93 19. Перебазирование.
94 20. Слияние в ветку master.
95 21. Клонирование репозитория. Просмотр клонированного репозитория.
96 22. Что такое origin? Удаленные ветки. Изменение оригинального репозитория.
97 23. Слияние извлеченных изменений.
98 24. Извлечение общих изменений.
99 25. Написание отчета по лабораторной работе в формате Markdown.

```

Рис. 4.59: Заголовок и нумерованный список

```
106
107 Существует три типа систем контроля версий:
108 |
109 - Локальные системы контроля версий
110 - Централизованные системы контроля версий
111 - Распределенные системы контроля версий
112
```

Рис. 4.60: Маркированный список

```
разных этапах, чтобы и они сами, и их коллеги могли их увидеть позже.
106
107 Существует три типа систем контроля версий: - Локальные системы контроля версий -
Централизованные системы контроля версий - Распределенные системы контроля версий
108
109 **Что такое локальная система контроля версий (LVCS)?**
110
111 Этот тип системы контроля версий очень распространен и прост в использовании. Однако этот
метод может выдавать ошибки и подвержен атакам, потому что файлы хранятся в вашей
локальной системе.
112
113 Это означает, что вы можете потерять системный файл или случайно забыть каталог, с
которым вы работаете (и затем записать в другой каталог). Что такое централизованная
система контроля версий (CVCS)?
114
115 В этом типе контроля версий сервер работает как общее хранилище, в котором находятся все
версии кода. CVCS помогает разработчикам работать совместно.
116
117 Однако, несмотря на то, что такой метод позволяет разработчикам сотрудничать, если сервер
отключится на несколько секунд или будет поврежден, то есть шанс, что вы потеряете все
файлы. Это является серьезной проблемой при работе с CVCS.
118
119 В CVCS только несколько разработчиков могут работать совместно над проектом. Что такое
распределенная система контроля версий (DVCS)?
120
121 В настоящее время это новый и наиболее часто используемый тип системы контроля версий.
122
123 В DVCS у каждого разработчика есть полная резервная копия всех данных на сервере. Это
```

Рис. 4.61: Жирный текст



```

179
180 Это свойство GitHub очень важно. GitHub -- это веб-интерфейс не только
181 для разработчиков. Его также могут использовать дизайнеры, писатели и
182 все, кто хочет отслеживать историю своих проектов.
183
184 # Выполнение лабораторной работы
185
186 1. Подготовить пространство для работы.
187
188 Я уже ранее использовала гит, поэтому сейчас выполнила несколько команд
189 для того, чтобы вывести на экран имя и электронную почту, а после
190 поработала с параметрами установки окончаний строк и с установкой
191 отображения unicode. (рис. @fig:001).
192
193 ![Подготовить пространство для работы](image/1.png){#fig:001 width=70%}
194
195 2. Создание проекта.
196
197 Я начала работу в пустом рабочем каталоге с создания пустого каталога с
198 именем hello, затем вошла в него и создала там файл с именем hello.html.
199
200 Потом я создала git репозиторий из этого каталога. Добавила файл в
201 репозиторий, проверила его состояние. (рис. @fig:002,
202 @fig:003, @fig:004, @fig:005).
203
204 ![Создание проекта](image/2.png){#fig:002 width=70%}
205
206 ![Создать git репозиторий из этого каталога](image/3.png){#fig:003 width=70%}
207

```

Рис. 4.62: Работа с изображениями

```

343
344 13. Рассмотреть Git внутри: Каталог .git. Посмотрим каталог, в котором
345 хранится вся информация git.
346
347 Выполним:
348
349 ```
350 ls -C .git/objects
351 ```
352
353 Мы должны увидеть набор каталогов, имена которых состоят из 2 символов.
354 Имена каталогов являются первыми двумя буквами хэша sha1 объекта,
355 хранящегося в git. После углубимся в базу данных объектов, посмотрим
356 файл конфигурации, создающийся для каждого конкретного проекта.
357
358 Выполним:
359
360 ```
361 ls .git/refs
362 ls .git/refs/heads
363 ls .git/refs/tags
364 cat .git/refs/tags/v1
365 ```
366
367 Мы узнаём файлы в подкаталоге тегов. Каждый файл соответствует тегу,
368 ранее созданному с помощью команды git tag. Его содержание --- это всего
369 лишь хэш коммита, привязанный к тегу. Посмотрим файл HEAD содержит
370 ссылку на текущую ветку, в данный момент это должна быть ветка master.
371 (рис. @fig:033, @fig:034, @fig:035).
372

```

Рис. 4.63: Вставка кода

```
506
507 # 5 Выводы
508
509 В ходе данной работы я приобрела практические навыки работы с системой
510 контроля версий git. Работа с репозиториями, внесение изменений,
511 индексация, коммиты.
512
513 # Список литературы
514
515 1. [Руководство к лабораторной работе по git] (https://esystem.rudn.ru/pluginfile.php/2584275/mod\_folder/content/0/git.pdf)
516 2. [Руководство к лабораторной работе по Markdown] (https://esystem.rudn.ru/pluginfile.php/2584275/mod\_folder/content/0/markdown.pdf)
```

Рис. 4.64: Ссылки

## 5 Выводы

В ходе данной работы я приобрела практические навыки работы с системой контроля версий git. Работа с репозиториями, внесение изменений, индексация, коммиты.

## 6 Список литературы

1. Руководство к лабораторной работе по git
2. Руководство к лабораторной работе по Markdown