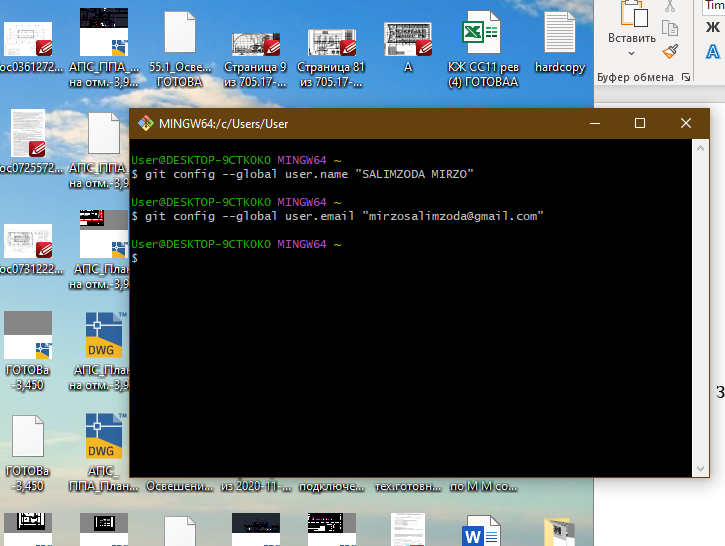
**Салимзода Мирзо ДКИП 204 прог**

**Практика номер 1**

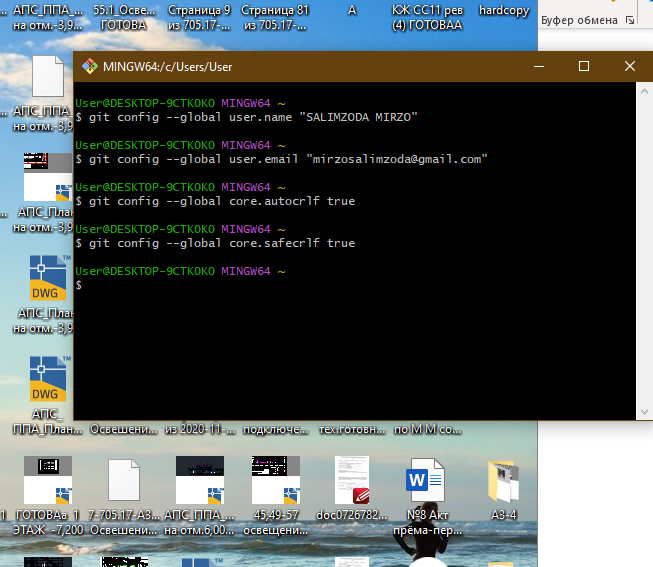
# Работа с Git.

## Подготовка

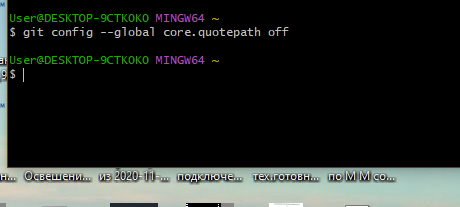
## Установка имени и электронной почты



Параметры окончаний строк

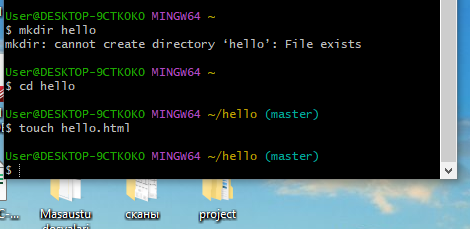


Установка отображения unicode

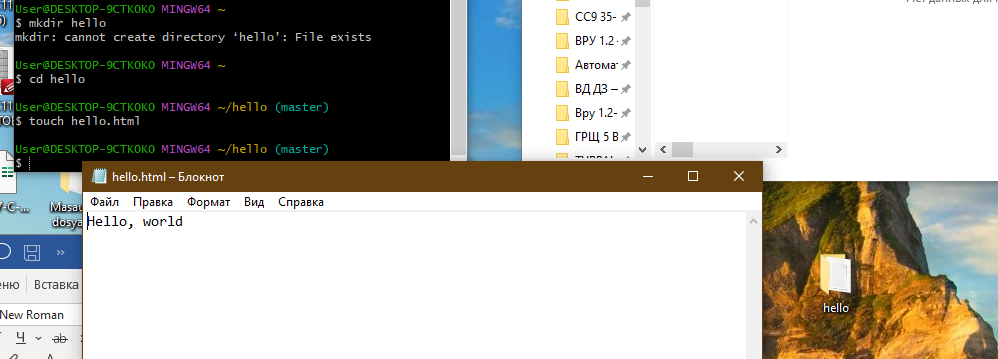


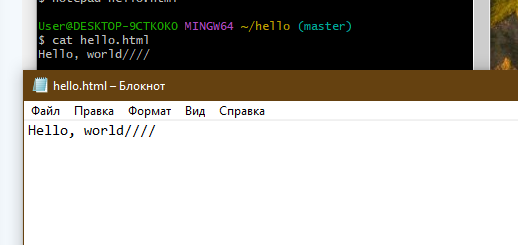
## Создание проекта

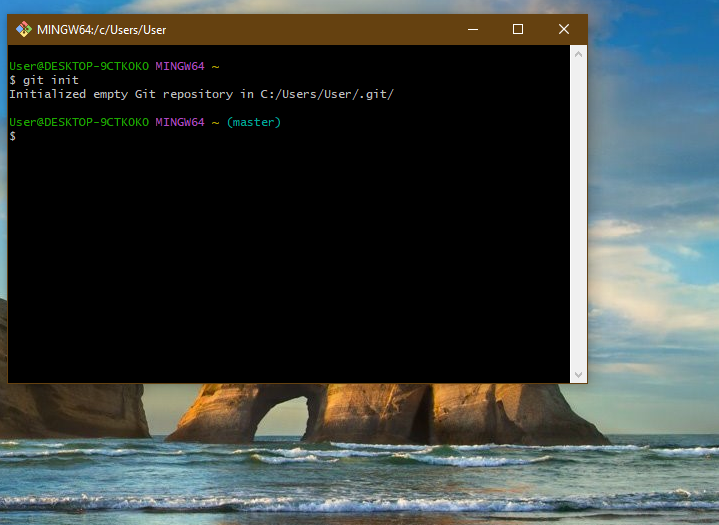
Создание каталога и файла



Создание репозитория





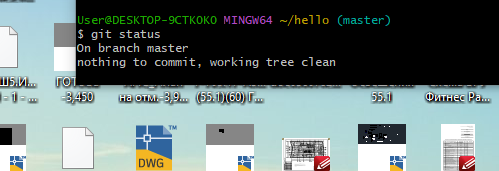


Добавление в репозиторий

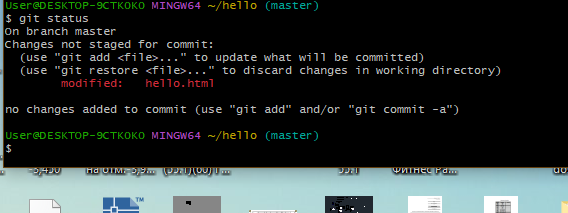


## Проверка состояния

Проверка состояния репозитория

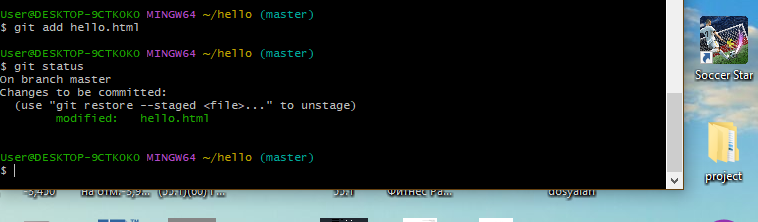


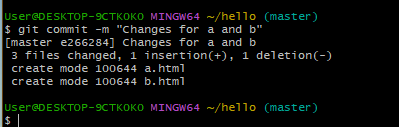
Сообщение о незафиксированных изменениях



## Индексация изменений

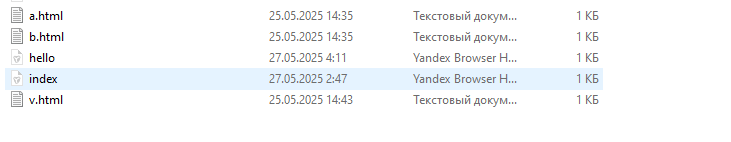
Команды для индексации изменений





## Индексация и коммит

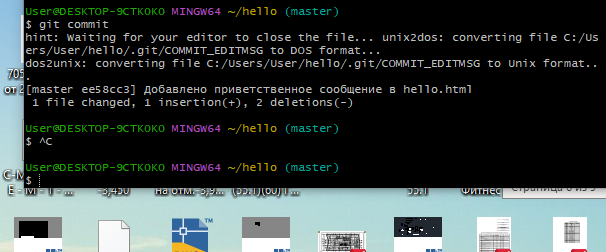
Создано 3 файла

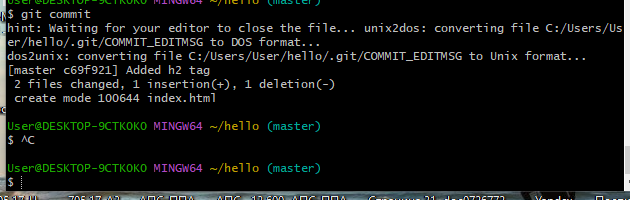


Индексация и коммит

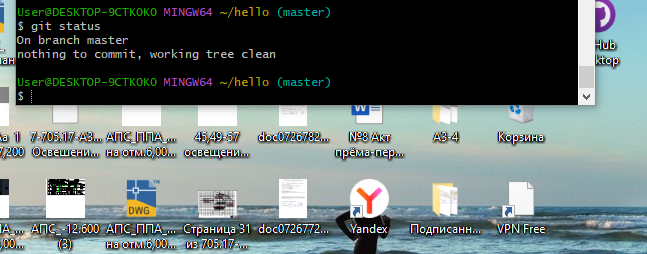


Индексация и коммит



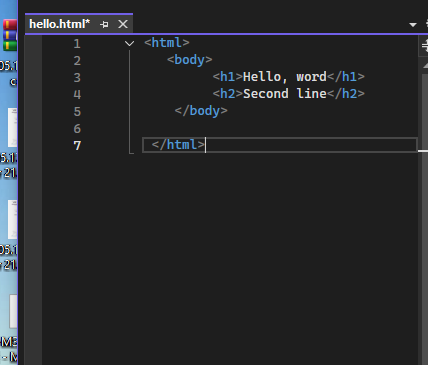


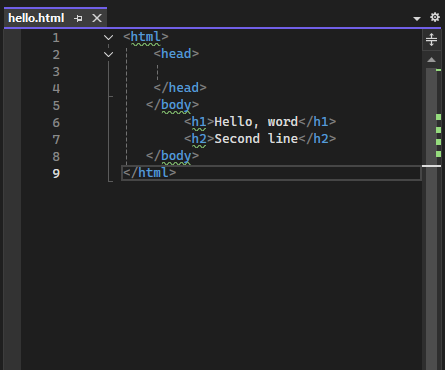
Проверка состояния



## Изменения, а не файлы

Добавление тегов html и body

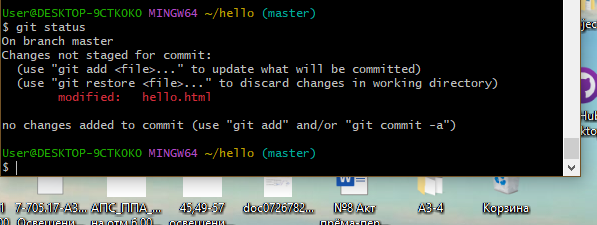




Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Проверка состояния



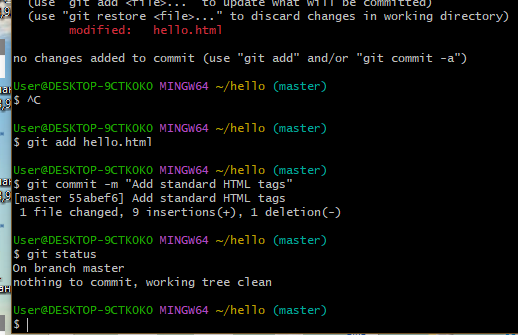
Коммит 1 изменения и проверка состояния

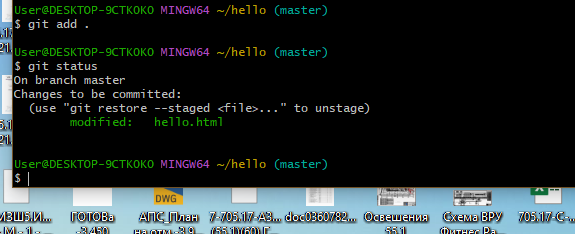
Изображение выглядит как текст, снимок экрана, Шрифт

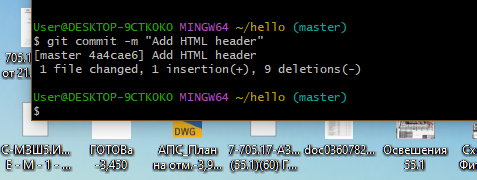
Контент, сгенерированный ИИ, может содержать ошибки.

Индексация 2 изменения и проверка состояния

Коммит 2 изменения



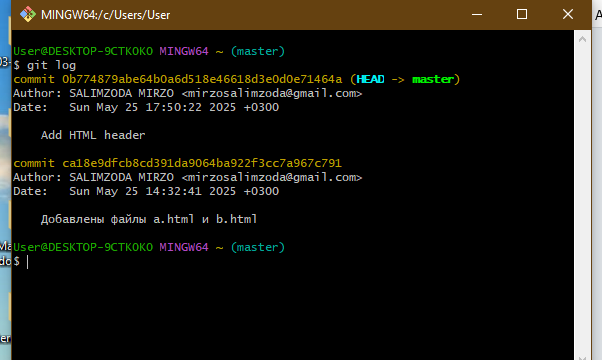


23)

# Работа с Git

## История

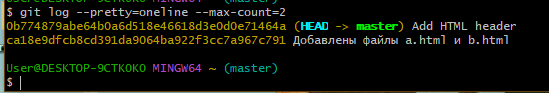
Рисунок 1 - Просмотр истории изменений



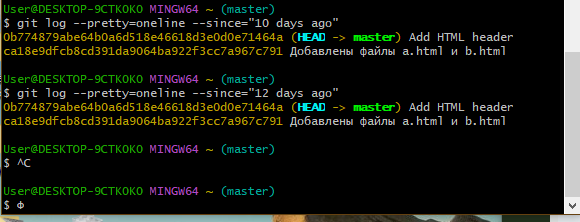
2) Рисунок 2 - Однострочный формат вывода



3) Рисунок 3 - Вывод последних 2 изменений



4) Вывод изменений начиная с определенного времени



5) Рисунок 4 - Вывод изменений до определенного времени

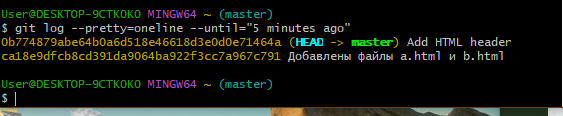
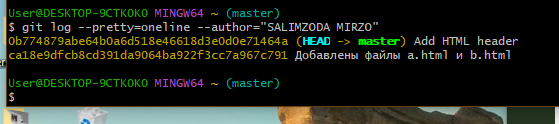
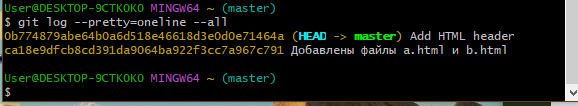


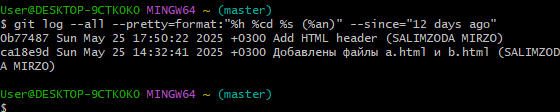
Рисунок 5 - Вывод изменений, внесенных определенным автором



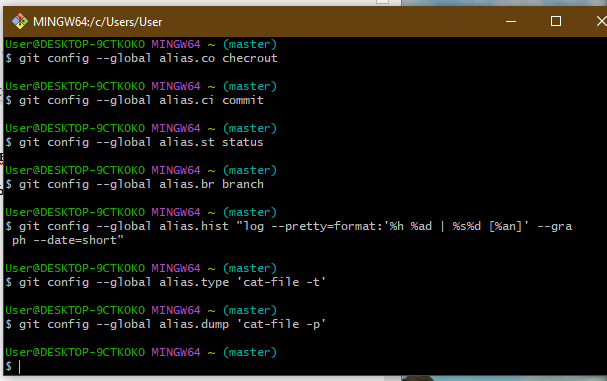
7) Рисунок 6 - Вывод всех изменений



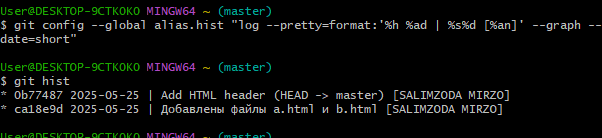
8) Рисунок 7 - Использование нескольких параметров



9) Рисунок 8 - Настройка алиасов для некоторых команд



10) Рисунок 9 - Выполнение алиаса hist



11) Рисунок 10 - Установка и выполнение алиаса gs

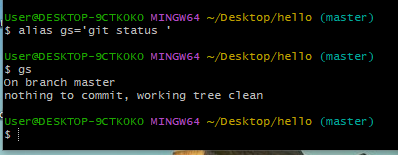
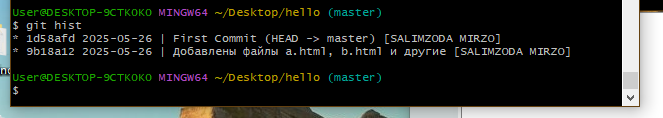
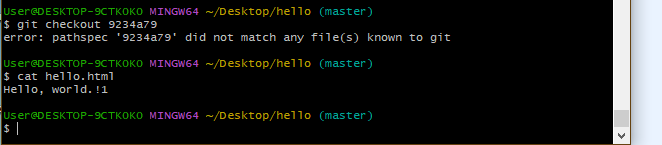


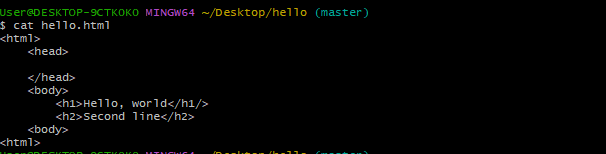
Рисунок 11 - Просмотр хэшей предыдущих версий



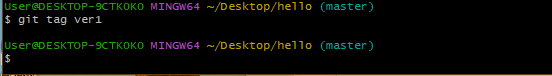
13) Возвращение к нужной версии и просмотр содержимого файла



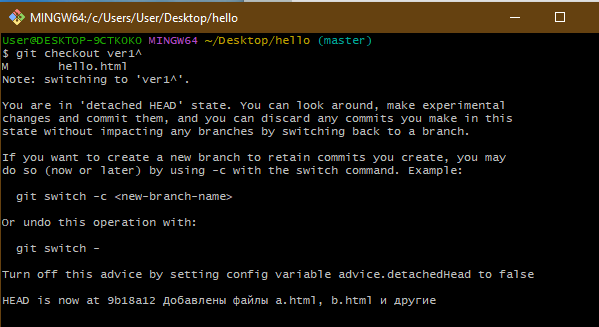
14) Рисунок 12 - Возвращение к последней версии в ветке master



15) Рисунок 13 - Задание тега



16) Рисунок 14 – Переход к предыдущей версии с помощью тега



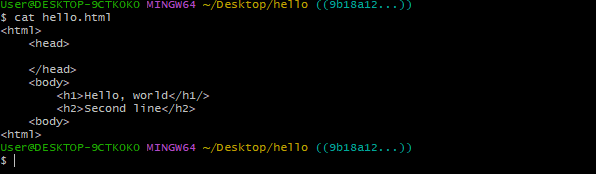
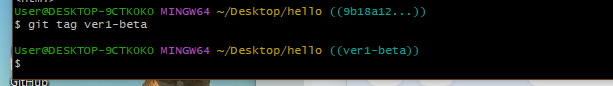
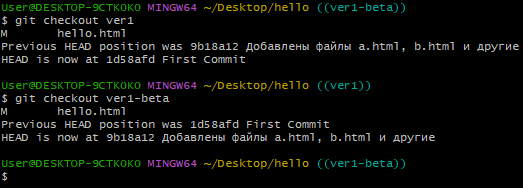


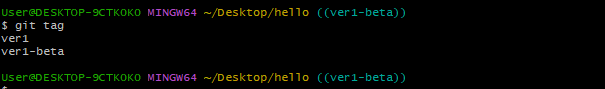
Рисунок 15 - Задание тега предыдущей версии



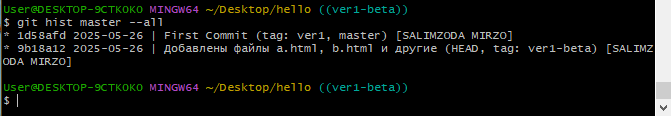
18) Рисунок 16 - Переключение между версиями с помощью тегов



19) Рисунок 17 - Просмотр тегов



20) Рисунок 18 - Просмотр тегов в логе



21) Рисунок 19 - Переключение на последний коммит

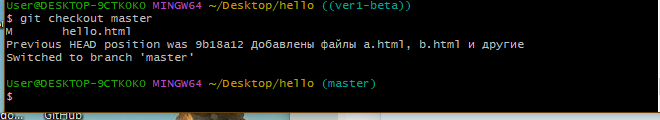
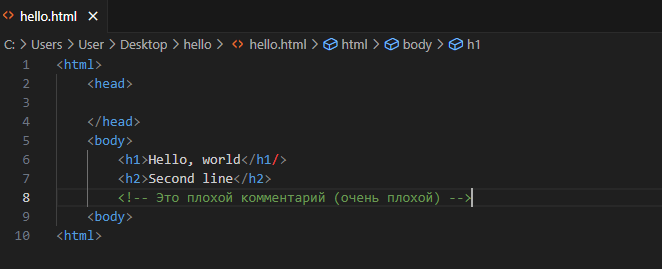
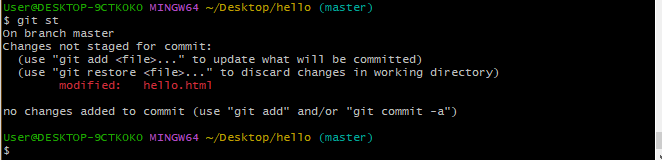


Рисунок 20 - Внесение изменения в файл



23) Рисунок 21 - Не проиндексированное изменение



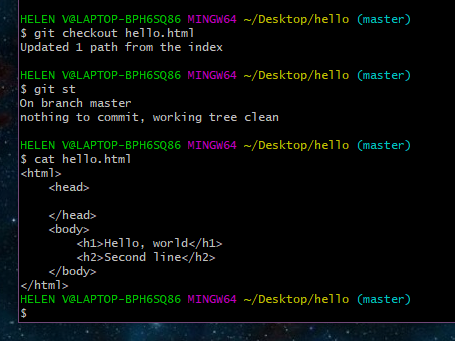
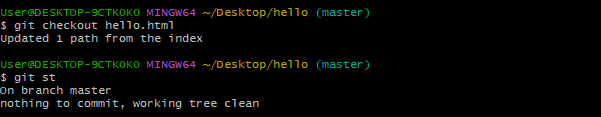
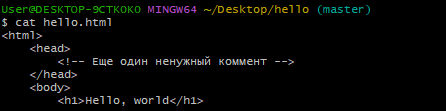
24) 

Рисунок 22 - Возвращение к версии





<html>

    <head>

        <!-- Еще один ненужный коммент -->

    </head>

    <body>

        <h1>Hello, world</h1>

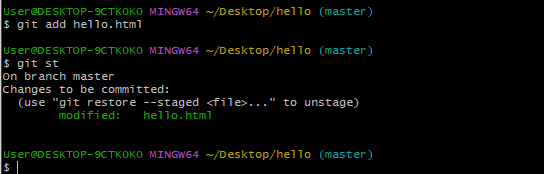
        <h2>Second line</h2>

    </body>

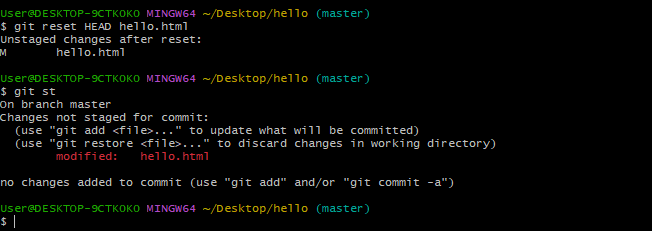
    </html>

Рисунок 23 - Внесение ненужного изменения

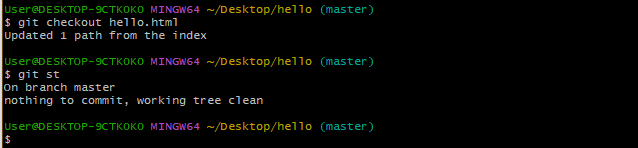
26) Рисунок 24 - Индексация изменения



27) Рисунок 25 - Очистка буферной зоны



28) Рисунок 26 - Удаление ненужных изменений



29) Рисунок 27 - Внесение изменения в файл

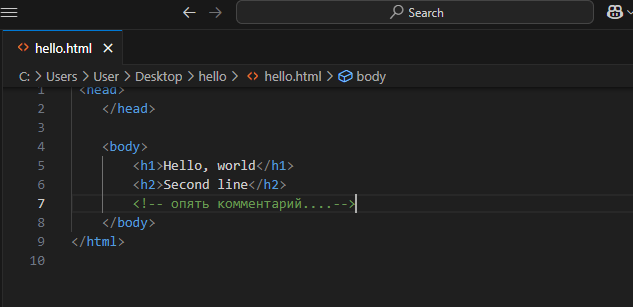


Рисунок 28 - Индексация и коммит

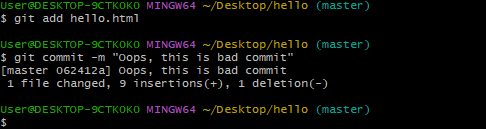
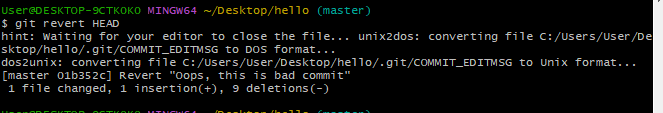


Рисунок 29 - Выполнение команды git revert



32) Рисунок 30 - Коммит сообщение в редакторе

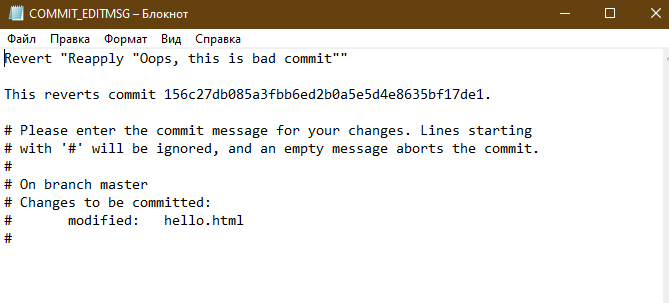


Рисунок 31 - Редактор закрыт

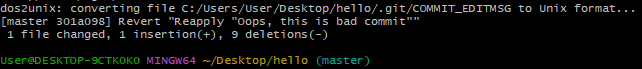


Рисунок 32 - Все коммиты при просмотре лога

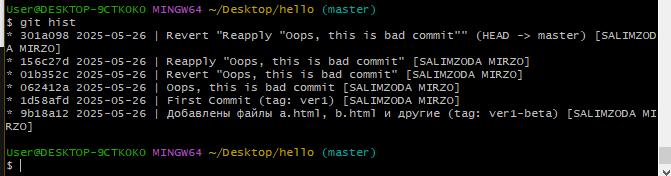


Рисунок 33 - Сброс коммита

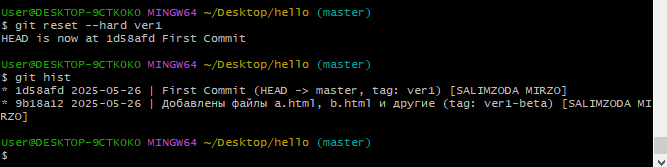
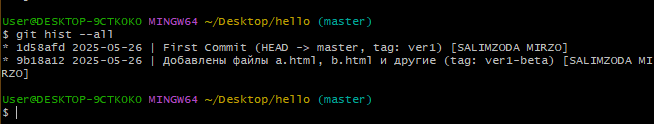
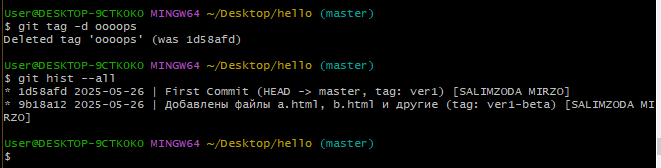


Рисунок 34 - Удаление тега



38)  


# Работа с Git

## Внесение изменений в коммиты

1) Рисунок 35 - Добавление комментария в файл

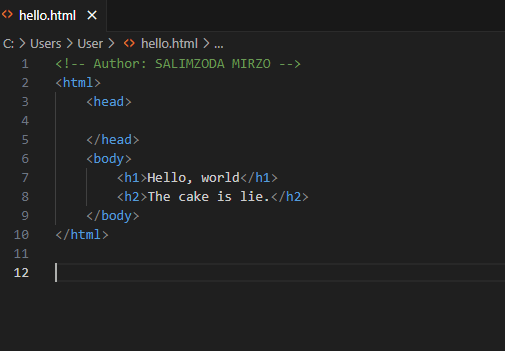


Рисунок 36 - Индексация и коммит

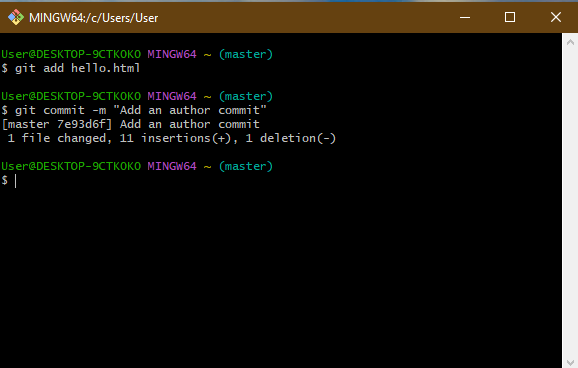


Рисунок 37 - Добавление электронной почты

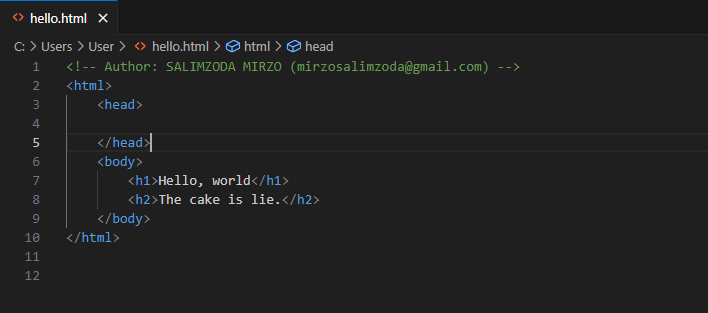
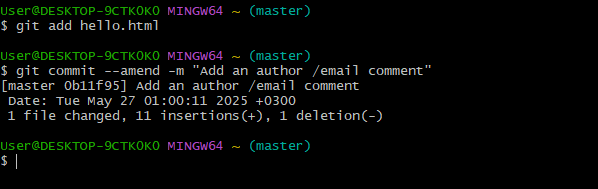
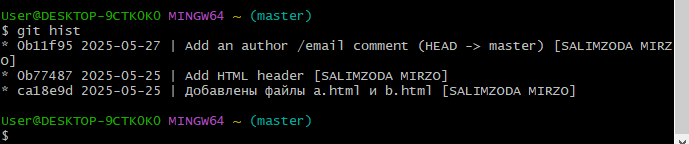


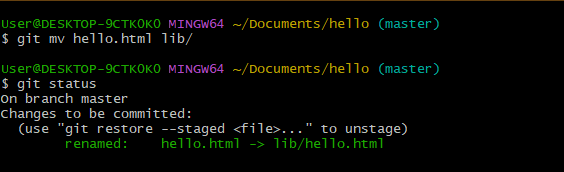
Рисунок 38 - Индексация и изменение коммита



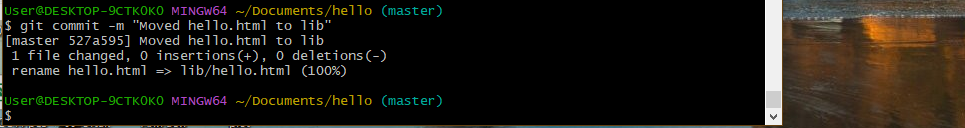
5)



6)



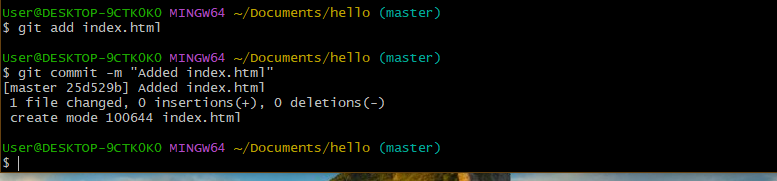
7)



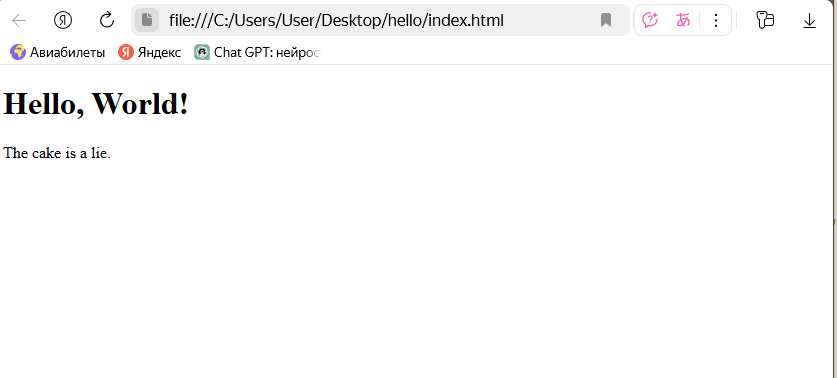
8)



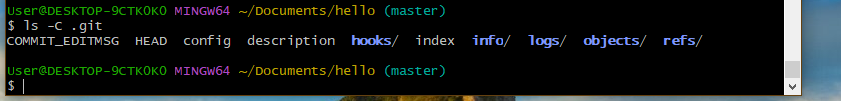
9)



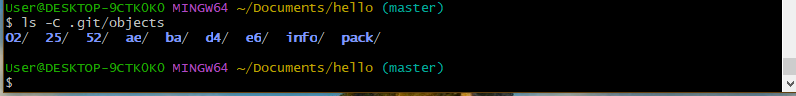
10)



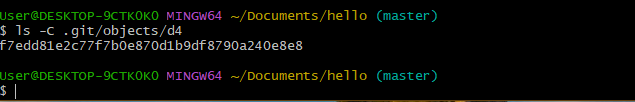
11)



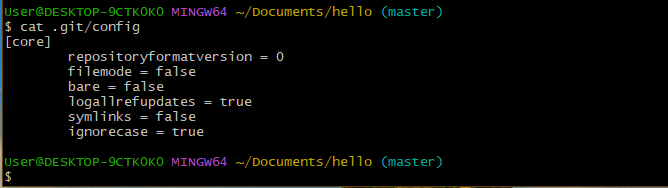
12)



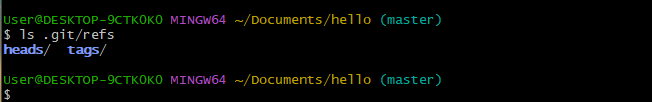
13)



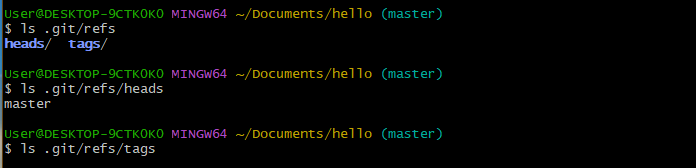
14)

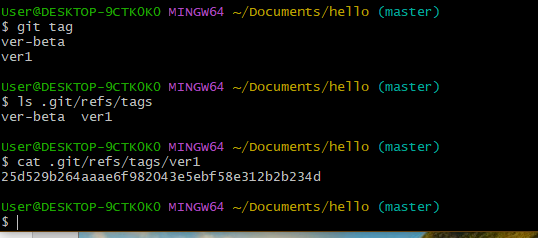


15)

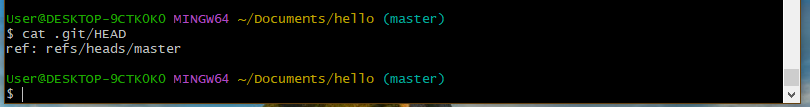


16)

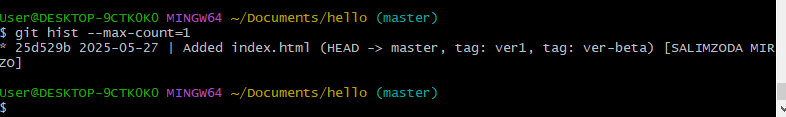




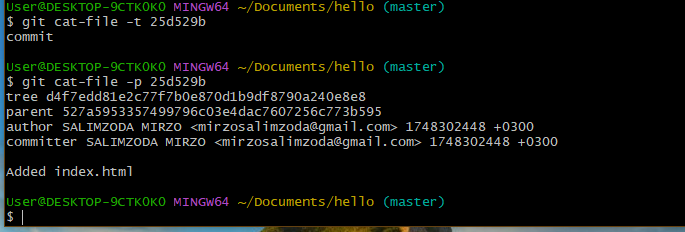
17)



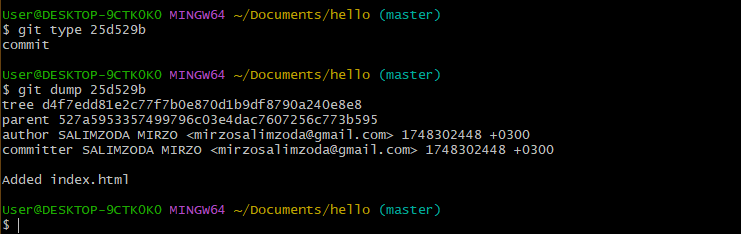
18)



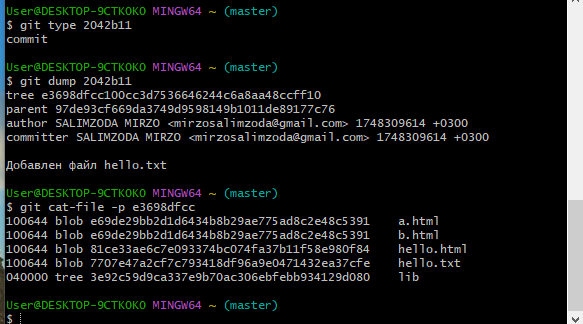
19)

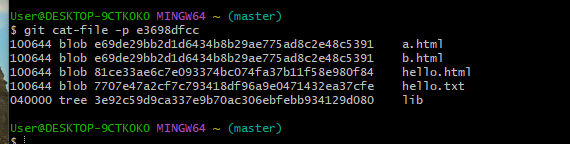


20)

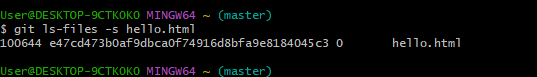


21)

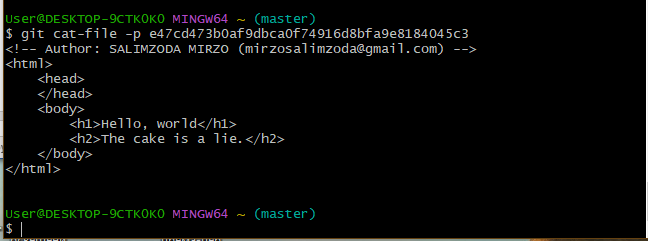




22)



23)



24)

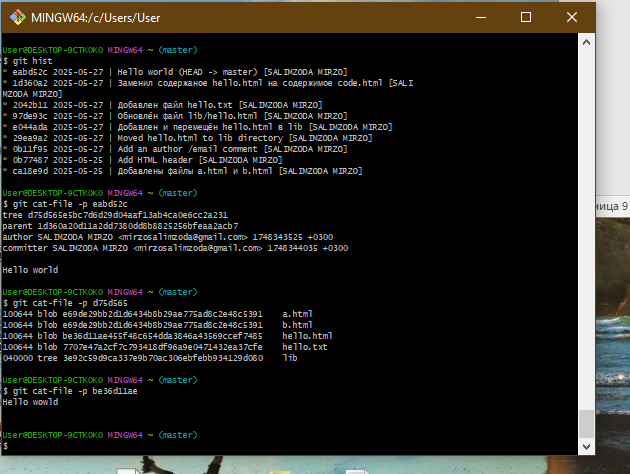


Рисунок 5 - Последний коммит изменен

Перемещение файлов

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

Рисунок 6 - Перемещение файла

Далее надо осуществить коммит данного перемещения.

Рисунок 7 – Коммит перемещения

Подробнее о структуре

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

Рисунок 8 - Содержимое файла index.html

Далее нужно проиндексировать и закоммитить файл (рисунок 9).

Рисунок 9 - Индексация и коммит

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

Рисунок 10 - Файл index.html, открытый в браузере

Каталог .git

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

Рисунок 11 - Содержание каталога .git

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

Рисунок 12 - Содержание каталога objects

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

Рисунок 13 - Просмотр каталога 8d

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

Рисунок 14 - Просмотр файла конфигурации

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

Рисунок 15 - Просмотр файлов и веток

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

Рисунок 16 - Содержимое файла HEAD

Работа с объектами git

Для начала необходимо просмотреть последний коммит (рисунок 17).

Рисунок 17 - Последний коммит

Далее надо использовать хэш последнего коммита, используя команды cat-file -p и cat-file -t (рисунок18) для просмотра объекта коммита. Также вместо длинных команд можно использовать сокращенные type и dump, если данные команды были заданы как алиасы.

Рисунок 18 - Просмотр объекта коммита

Рисунок 19 - Использование алиасов

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

Рисунок 20 - Просмотр дерева каталогов

Затем нужно просмотреть каталог lib (рисунок 21).

Рисунок 21 - Просмотр каталога lib

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

Рисунок 22 - Вывод содержимого файла hello.html

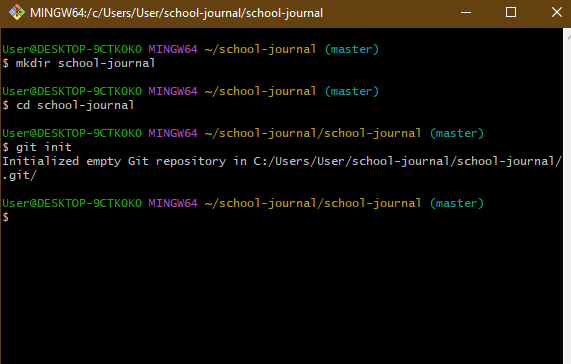
Аналогичным образом можно просмотреть содержимое файла, каким оно было в самом первом коммите, как показано на рисунке 23. Для этого требуется использовать лишь нужные хэши.

Рисунок 23 - Просмотр содержимого файла при первом коммите

### САЛИМЗОДА МИРЗО ДКИП 204 прог

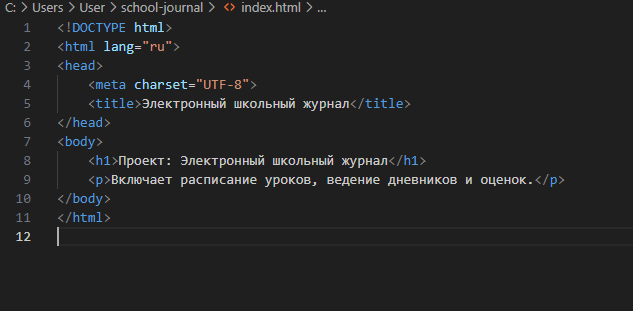
### Шаг 1: Создание репозитория и начальная настройка

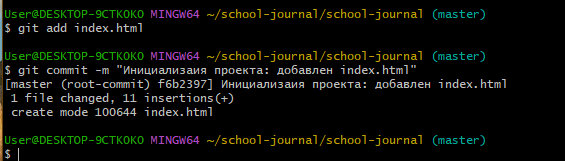
1. **Откройте Git Bash.**
2. **Создайте новую папку проекта и инициализируйте Git-репозиторий:**

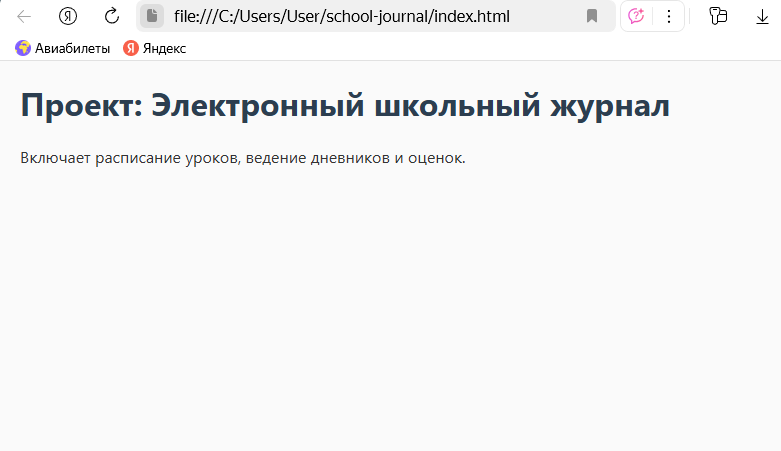


**3.Создайте файл index.html, который будет служить главной страницей проекта. Вставьте в него базовую разметку**

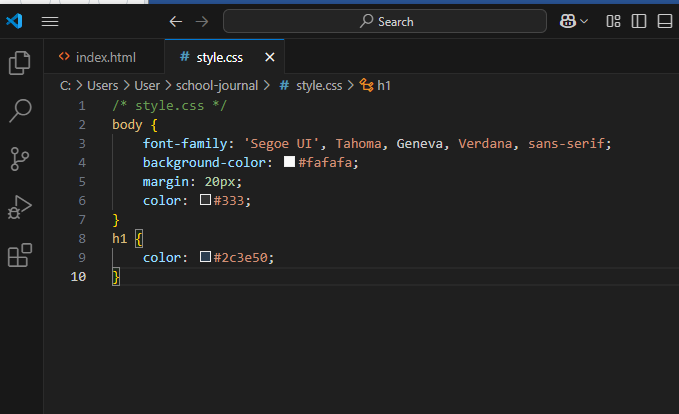






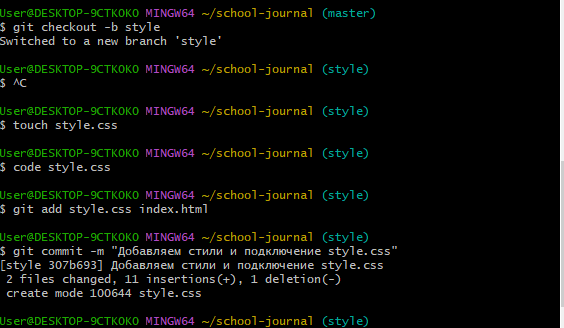


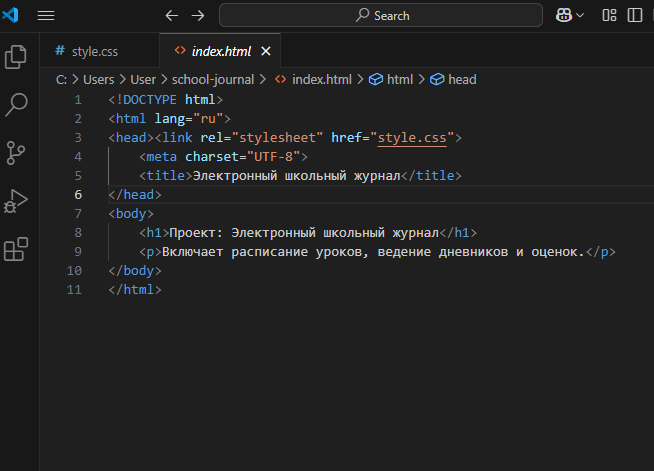
**Шаг 2: Создание ветки для стилей**



### Шаг 2: Создание ветки для стилей

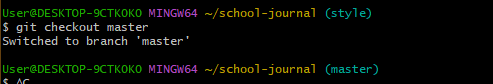
1. **Создайте новую ветку style для разработки CSS-стилей:**



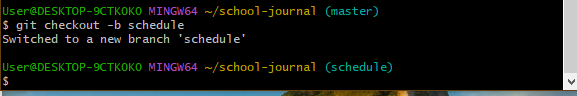


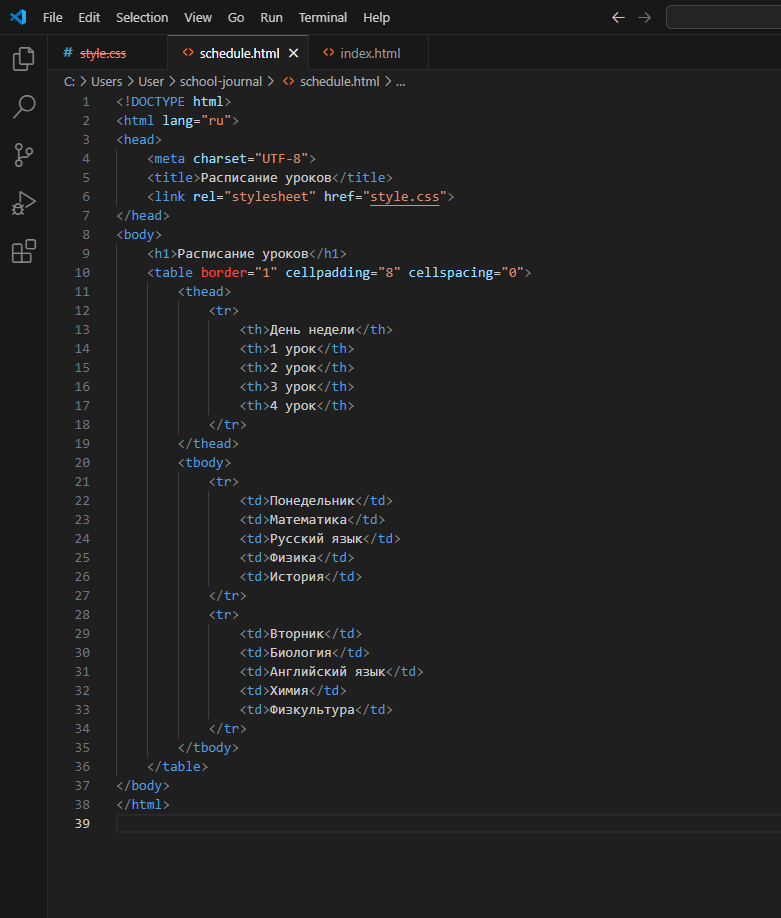
### 

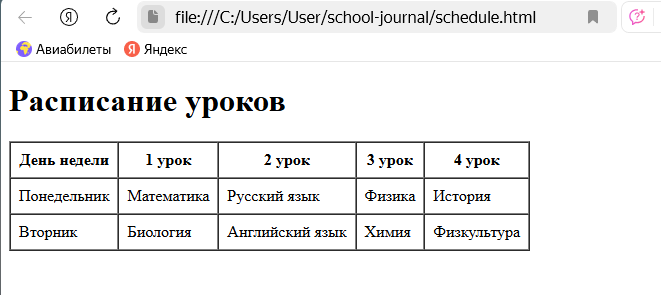
### Шаг 3: Создание ветки для функционала расписания уроков

**Переключитесь на основную ветку master (или main):** 

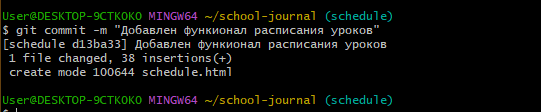
**Создайте новую ветку schedule для реализации расписания уроков:**



Расписнаие уроков  




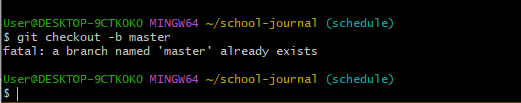
**Добавьте и зафиксируйте изменения**



### 

### Шаг 4: Добавление ссылки на расписание в главную страницу и коммит

1. **Переключитесь на ветку master:**



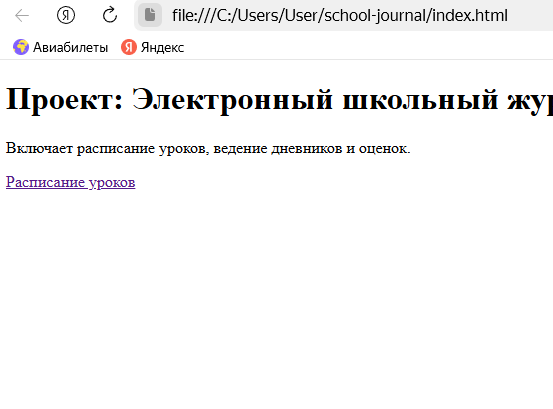
Вставьте следующий код в тело <body>:

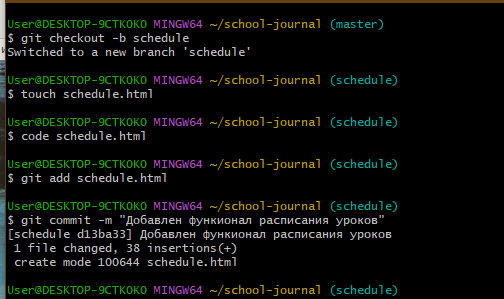
<a href="schedule.html">Расписание уроков</a>

Сохраните изменения и создайте коммит:

git add index.html

git commit -m "Добавлена ссылка на расписание уроков"



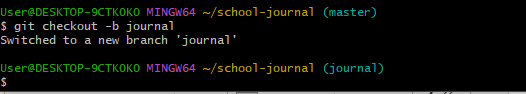


Шаг 5: Создание ветки для электронного журнала

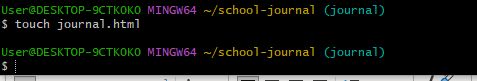
Создайте новую ветку journal для реализации функционала электронного журнала:

git checkout -b journal

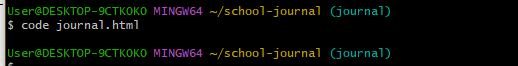
Создайте файл journal.html с разметкой для ввода оценок:

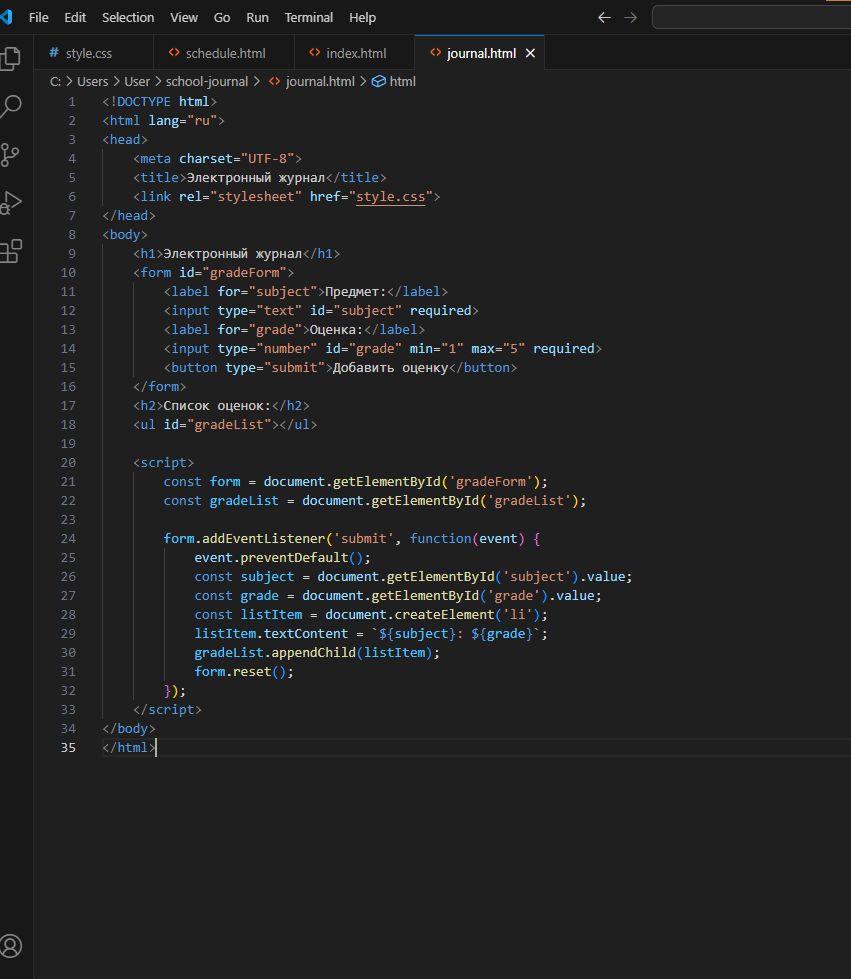


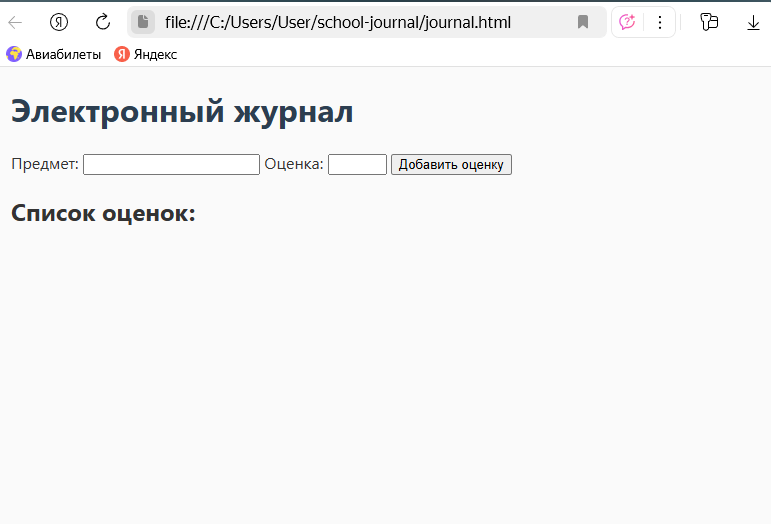
**Создайте файл journal.html с разметкой для ввода оценок:**



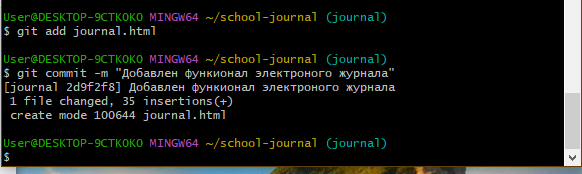
Затем откройте journal.html и добавьте следующий код:







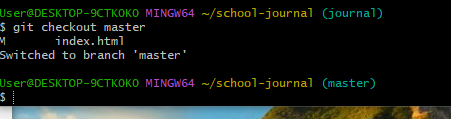
**Добавьте и зафиксируйте изменения:**



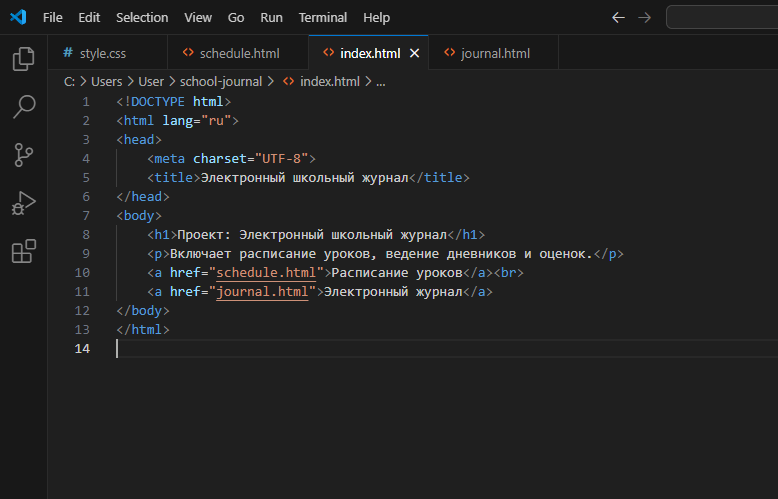
### 

### Шаг 6: Добавление ссылки на электронный журнал в главную страницу и коммит

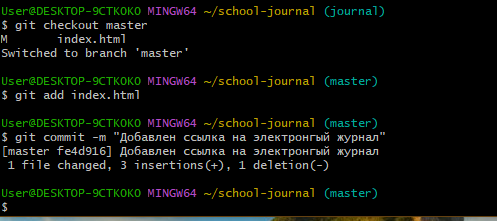
1. **Переключитесь на ветку master:**

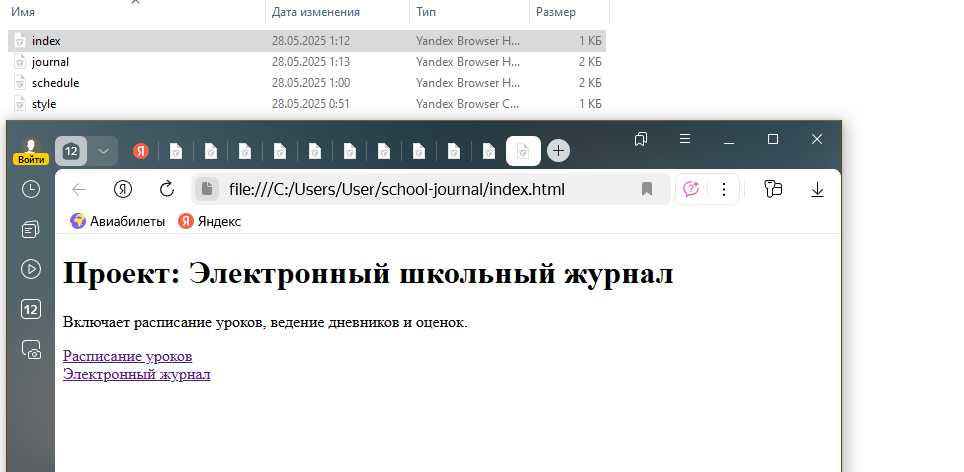


Вставьте следующий код в тело <body>:



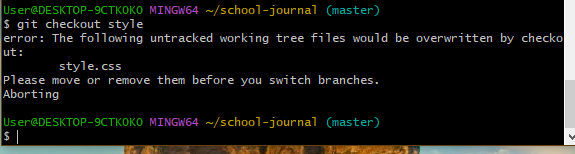
**Сохраните изменения и создайте коммит:**

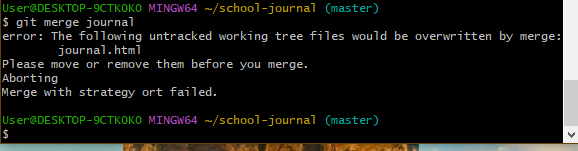


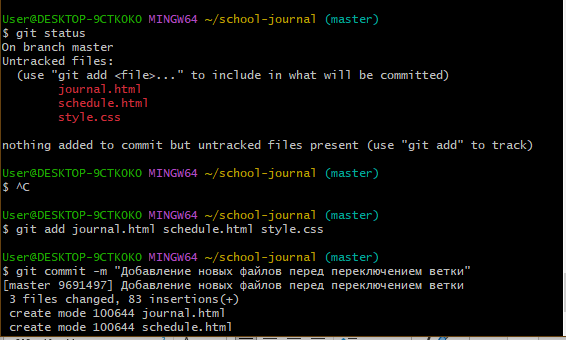


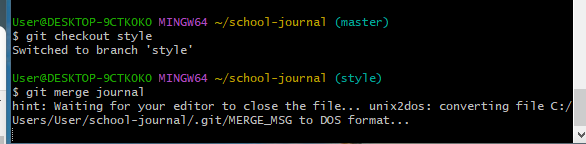
### Шаг 7: Слияние веток journal и style и разрешение конфликта

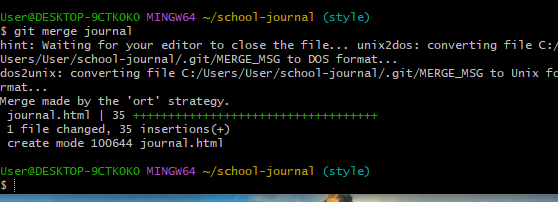
1. **Переключитесь на ветку style:**

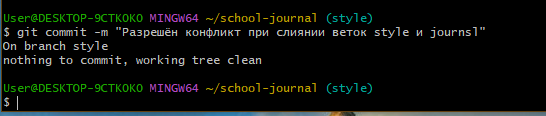


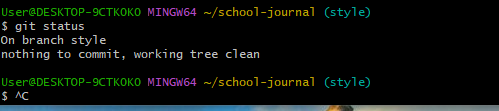












**Сообщение nothing to commit, working tree clean подтверждает, что все изменения были успешно закоммичены, и в твоем рабочем каталоге нет незакоммиченных файлов.**

**Итоги и выводы по проекту электронного школьного журнала:**

**В ходе работы я успешно создал репозиторий с использованием Git, освоил основные операции: инициализация, создание и переключение веток, коммиты и слияния. Создание отдельных веток для разных функций (стили, расписание, электронный журнал) помогло структурировать процесс разработки и упростило тестирование.**

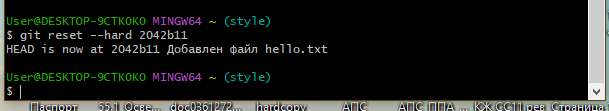
**Я также получил практический опыт разрешения конфликтов при слиянии веток, что улучшило понимание работы Git. В результате я реализовал базовый функционал журнала и повысил свои навыки работы с системами контроля версий, что будет полезно в будущих проектах..**

Практика номер 5  
Салимзода Мирзо

**Шаг 1: Сброс ветки style**

1. **Просмотр истории коммитов:**  
   Чтобы увидеть, какие коммиты были сделаны в ветке style, используйте команду:





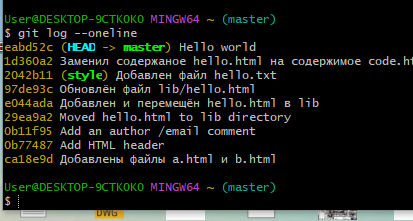
### 

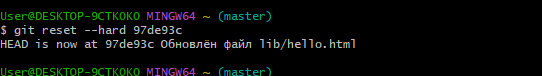
### Шаг 2: Сброс ветки master

1. **Переключитесь на ветку master:**



**Рисунок 3 - Переключение на master и просмотр истории**



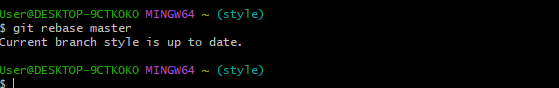


### Шаг 3: Перебазирование ветки style

1. **Переключитесь на ветку style, если вы еще не на ней:**

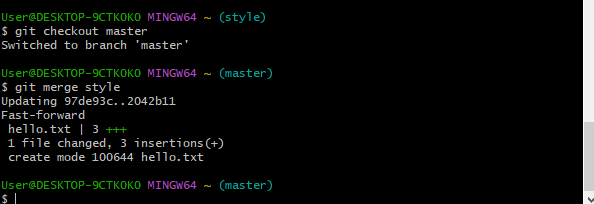


**Перебазируйте ветку style на master:**



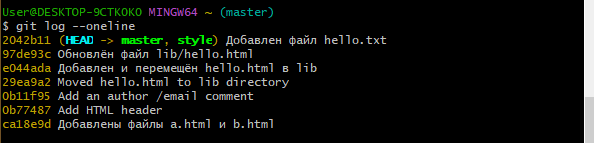
### Шаг 4: Слияние веток в master

1. **Переключитесь на ветку master:**



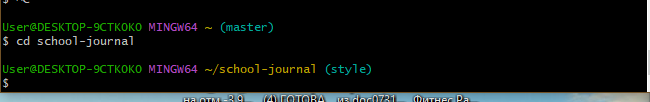
Это объединит изменения из ветки style в master

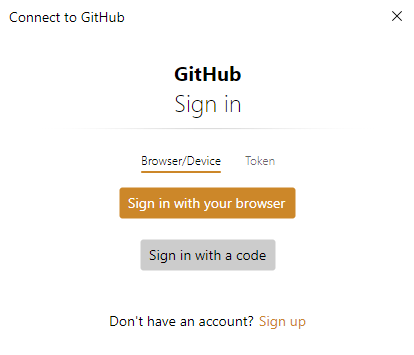
**Просмотр истории после слияния:**

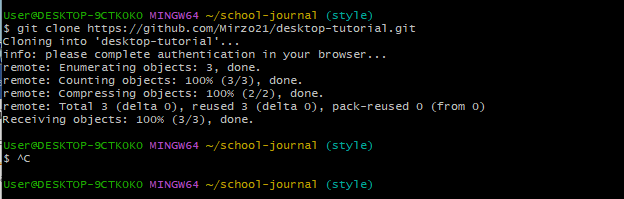


### Шаг 5: Клонирование репозиториев

1. **Перейдите в рабочий каталог, куда хотите клонировать репозиторий:**

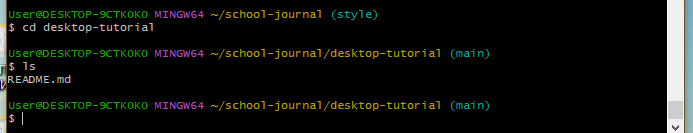


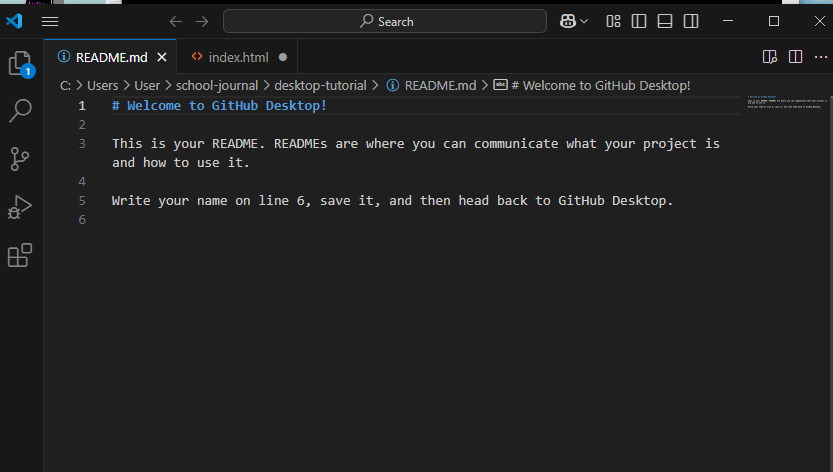


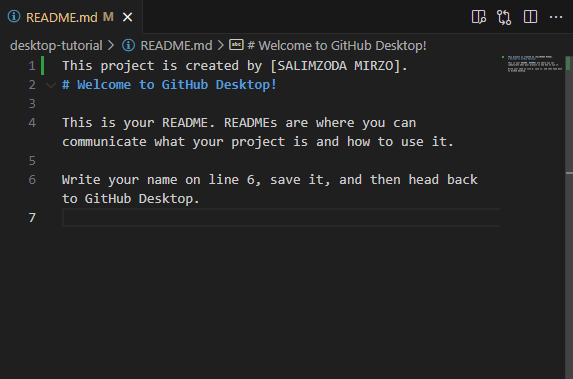
Клонирование репозитория:

### Шаг 6: Просмотр клонированного репозитория

1. **Просмотр содержимого каталога:**







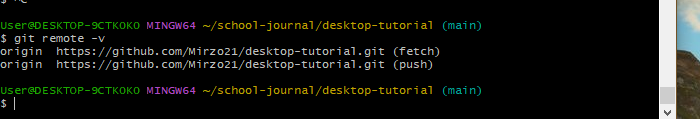
**Просмотр логов репозитория:**

**Просмотр логов клонированного репозитория:**



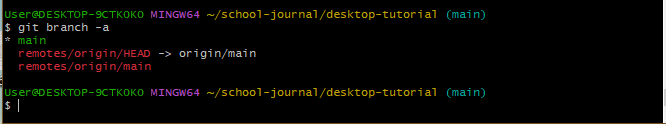
### Шаг 7: Работа с origin

1. Это покажет URL-адреса для операций fetch и push с удаленным репозиторием.



### Шаг 8: Просмотр удаленных веток

**Просмотр всех веток, включая удаленные:**



**Заключение**

Я успешно клонировал репозиторий `desktop-tutorial` на свой компьютер. Теперь я могу просматривать файлы и историю изменений проекта. С помощью команд `git remote -v` и `git branch -a` я проверил настройки удалённого репозитория и увидел все доступные ветки.

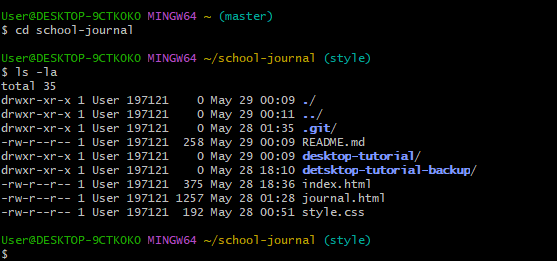
Эти шаги помогли мне лучше понять, как работать с Git и GitHub. В будущем я планирую вносить изменения и отправлять их обратно в репозиторий. Это полезный опыт для разработки проектов!

**САЛИМЗОДА МИРЗО ДКИП прог**

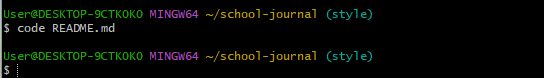
**Работа с Git: Школа — Расписание уроков, ведение дневников, оценки, электронный журнал**

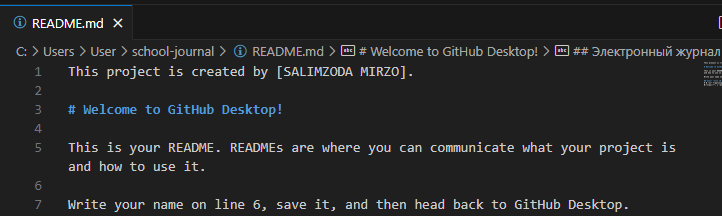
**6. Синхронизация изменений и работа с удалённым репозиторием**

**Внесение изменений в оригинальный репозиторий**

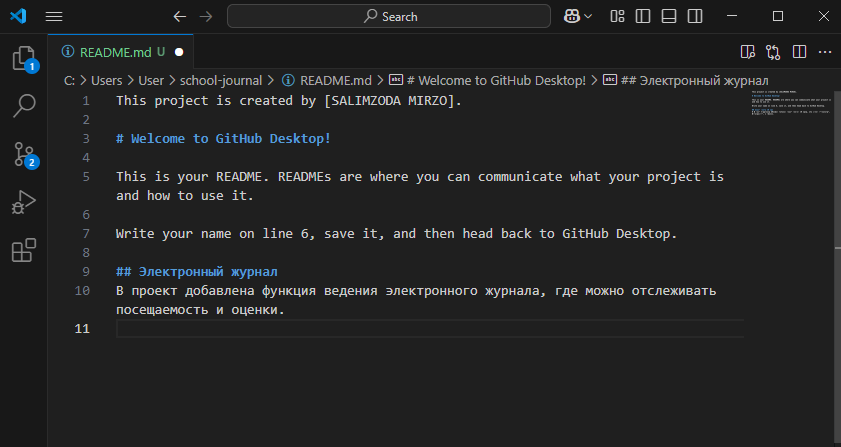


Далее внесите изменения в файл README.md, добавив информацию о новых функциях, например, о добавлении электронного журнала:



Это без изменения  


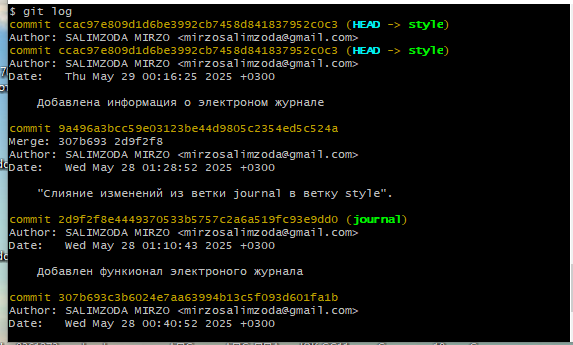
Это с изменения



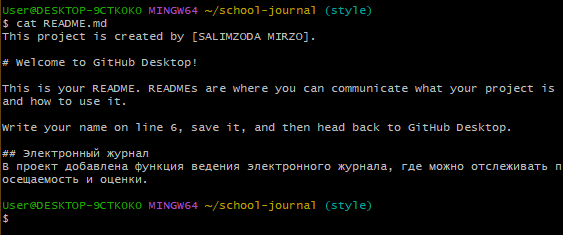
Затем произведите индексацию и коммит изменений:



### Извлечение изменений в клонированном репозитории

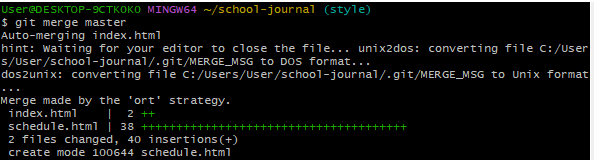


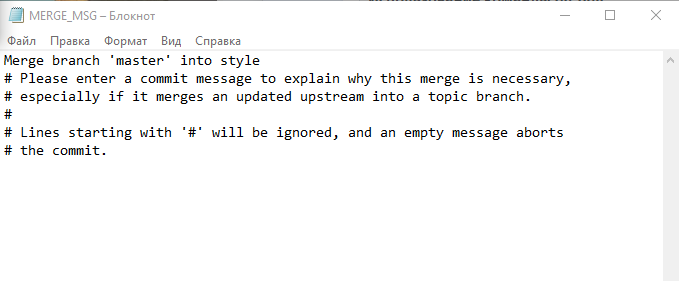
### Проверка содержимого файла

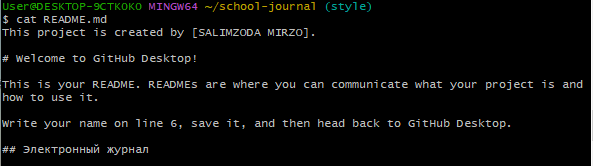


### Слияние изменений

Далее нужно слить извлеченные изменения в ветку master:

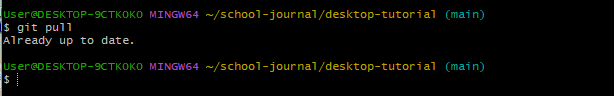






### Использование команды git pull

Также существует команда, объединяющая функции git fetch и git merge, которая показана ниже:

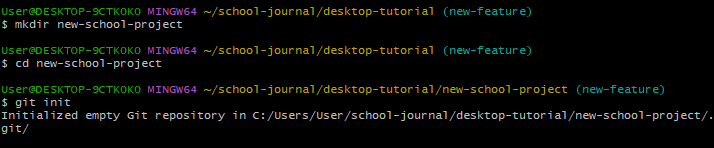


### Добавление локальной ветки

Далее требуется добавить локальную ветку, которая будет отслеживать удалённую ветку:

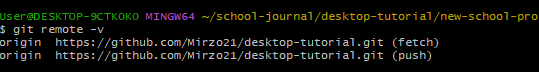


### Создание чистого репозитория

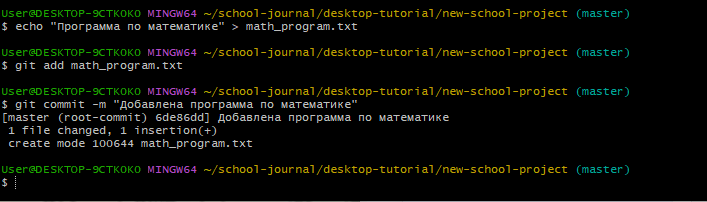


### Добавление удалённого репозитория

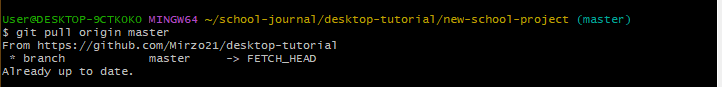




### Отправка изменений в удалённый репозиторий



Теперь отправим изменения в общий репозиторий:



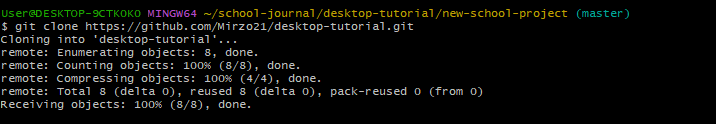
Ссылка тоже работает.

### Извлечение общих изменений

### 

### Настройка git сервера





Клонирование проекта

Итог:

Я научился синхронизировать изменения между локальным и удалённым репозиториями, а также работать с ветками. Теперь я знаю, как вносить изменения в файлы, коммитить их и отправлять в общий репозиторий. Также я освоил команды для извлечения изменений и слияния веток.

# САЛИМЗОДА МИРЗО

Практика номер 7

# Работа с Git

Проверить статус

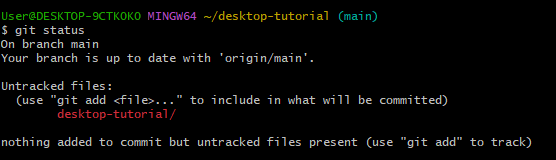
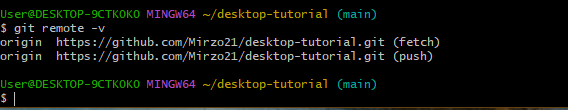


Рисунок - Просмотр удаленных репозиториев с ключом -v



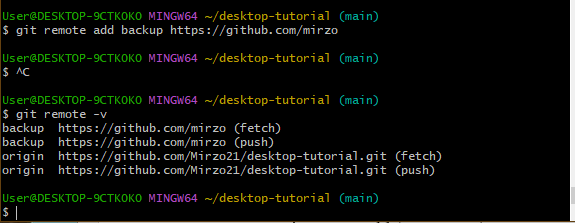
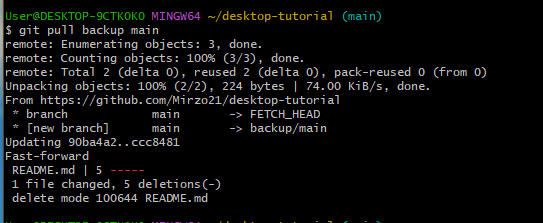
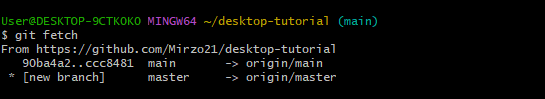
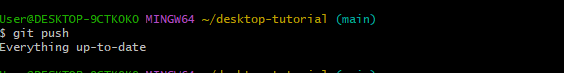


Рисунок - Добавление удаленного репозитория



### Получение и отправка данных





### Информация об удалённом репозитории

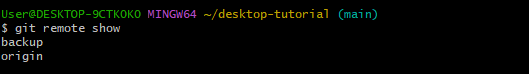


Рисунок 7 - Информация об удалённом репозитории

### 

### Переименование и удаление удалённых репозиториев

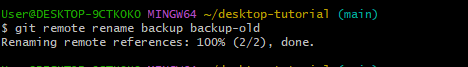
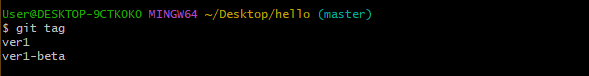


Рисунок 8 - Переименование удалённого репозитория



Рисунок 9 - Удаление удалённого репозитория

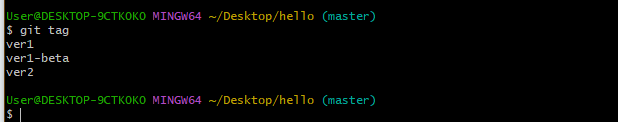
### Работа с тегами

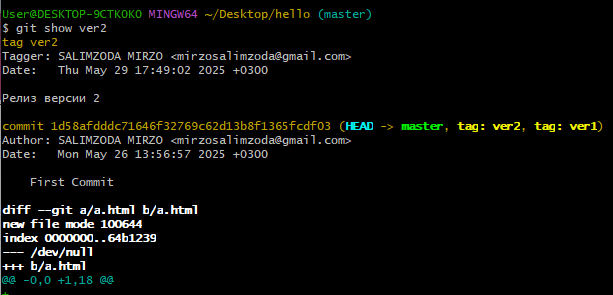


Для создания аннотированной метки:

Рисунок 11 - Создание аннотированной метки







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

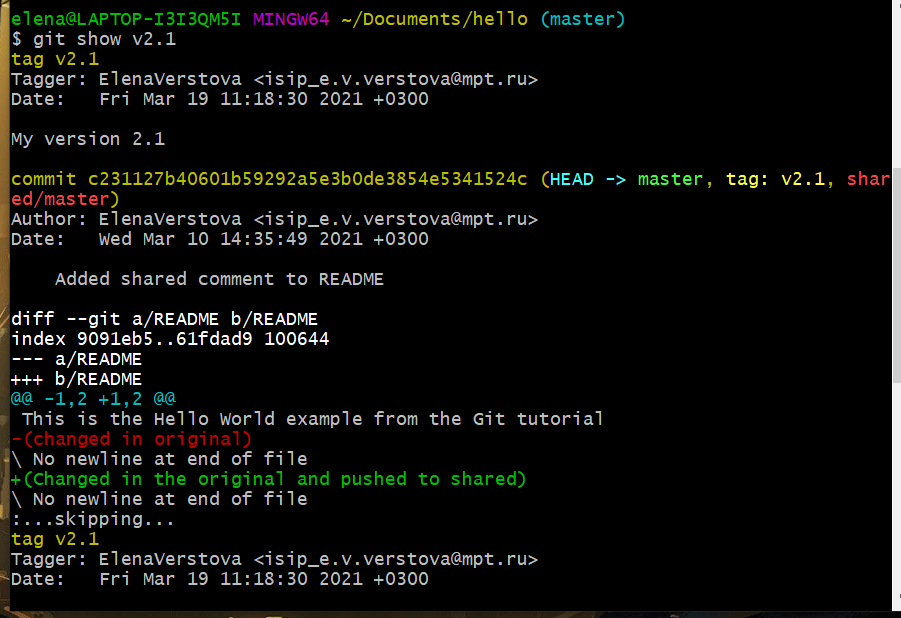


Рисунок - Просмотр данных тега



Рисунок - Создание легковесной метки

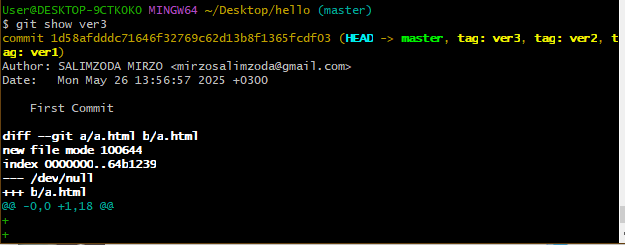


Рисунок - Просмотр данных тега

### Отметка коммита тегом

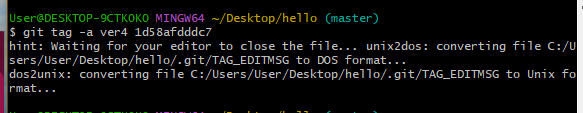


Рисунок 15 - Создание тега для определённого коммита

Данные этого тега можно просмотреть аналогично:

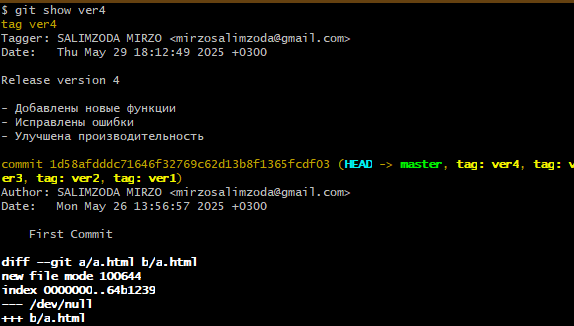


Рисунок 16

Рисунок - Ввод сообщения в текстовом редакторе

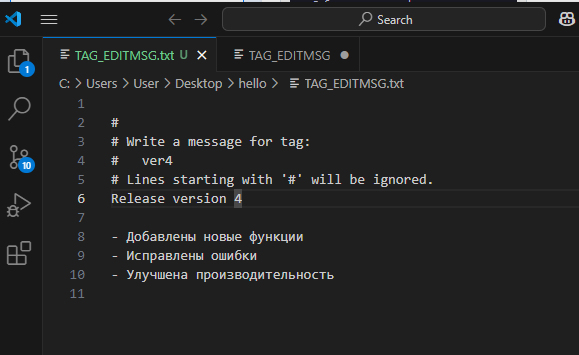
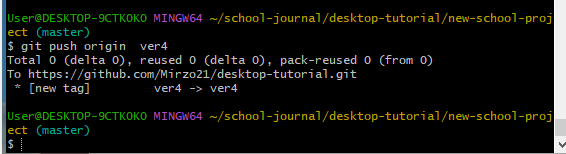


Рисунок 18 - Просмотр данных тега

### Работа с тегами на удалённом сервере



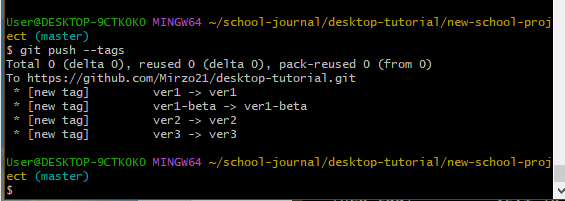


Рисунок - Отправка всех тегов на сервер

### Удаление тегов

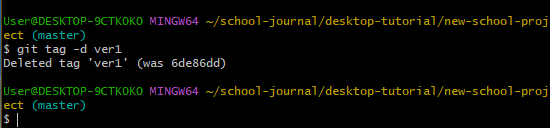


Рисунок - Удаление тега

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

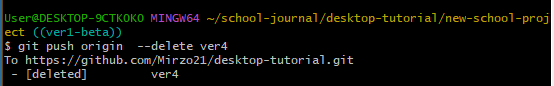


Рисунок - Удаление тегов с сервера

### Работа с тегами и состоянием «detached HEAD»

Чтобы получить версии файлов, на которые указывает тег, выполните:

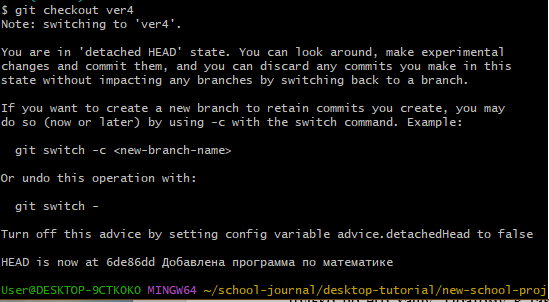


Рисунок - Переключение на метку

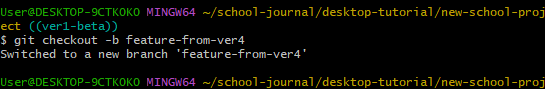


Рисунок - Создание новой ветки

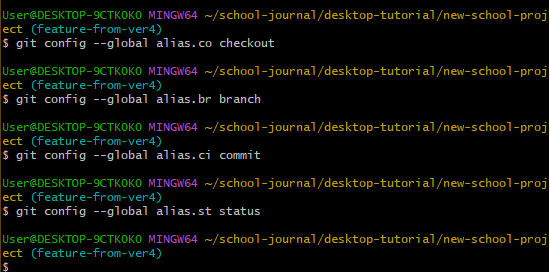


Рисунок - Задание алиасов

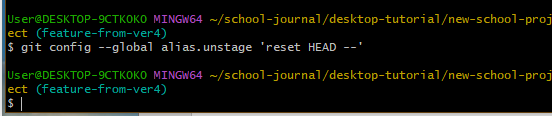


Рисунок - Создание псевдонима исключения файла из индекса

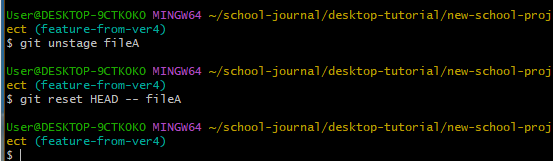
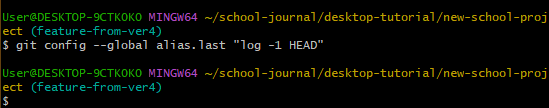


Рисунок - Использование созданного псевдонима



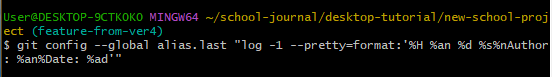
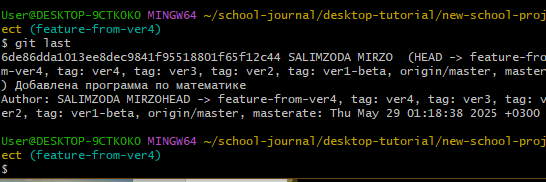


Рисунок - Создание алиаса для просмотра последнего коммита



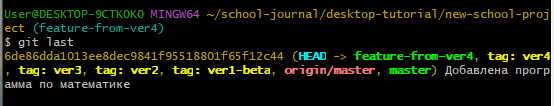


Рисунок - Результат работы созданного алиаса

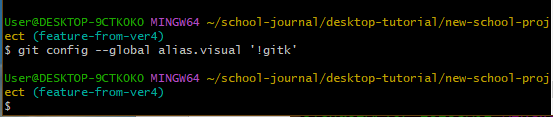
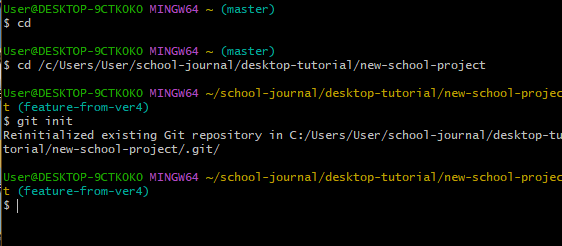


Рисунок - Создание псевдонима внешней команды

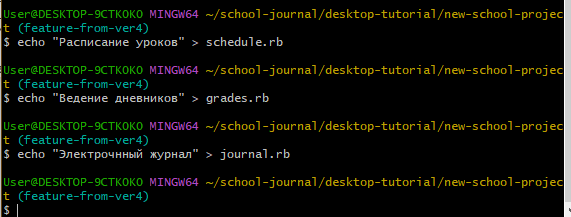
САЛИМЗОДА МИРЗО

**Работа с Git: Проект «Школа»**

**Шаг 1: Создание репозитория**

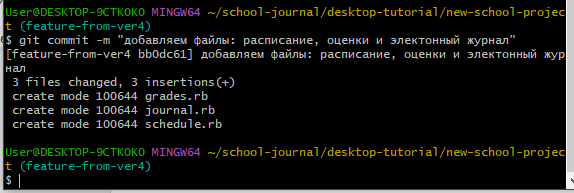


## Шаг 2: Создание файлов

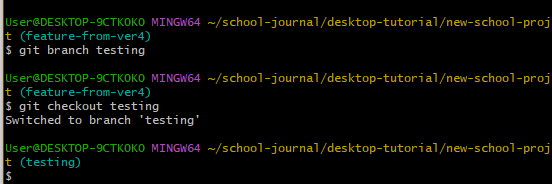


## 

## Шаг 3: Индексация и коммит файлов

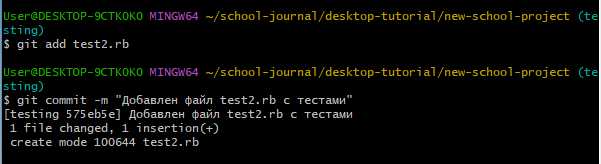


## Шаг 4.5 : Создание ветки testing Шаг 5: Переключение на ветку testing

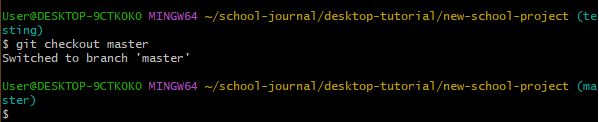


## 

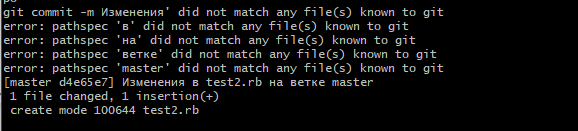
## Шаг 6: Внесение изменений в файл test.rb



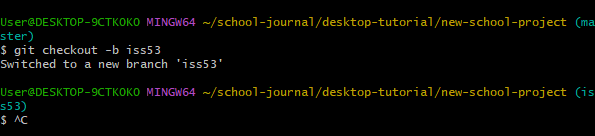
## Шаг 7: Переключение на ветку master



## Шаг 8: Внесение изменений в файл test.rb на ветке master

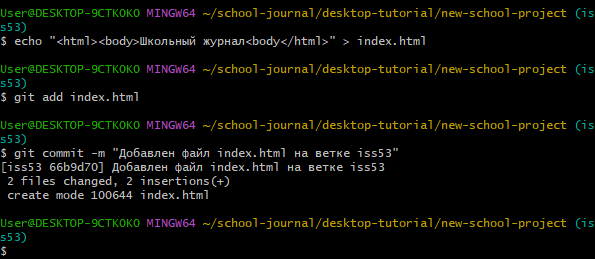


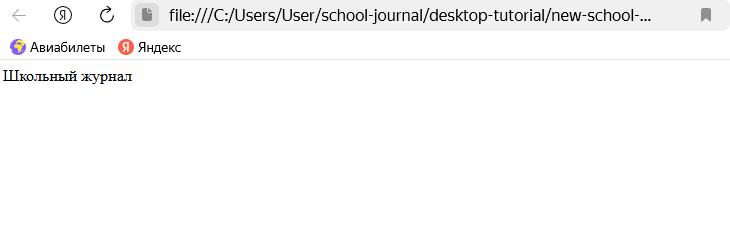
## Шаг 9: Создание и переключение на ветку iss53



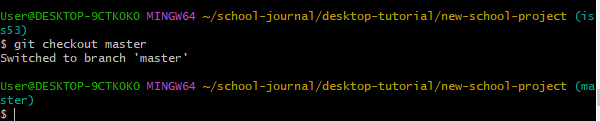
## Шаг 10: Внесение изменений в файл index.html

## 

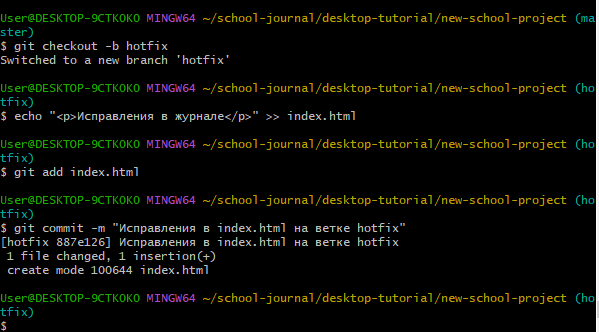


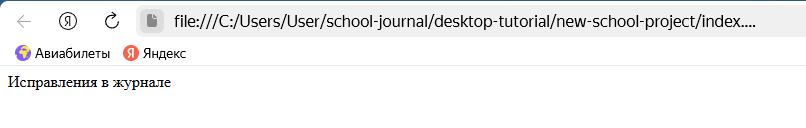


## Шаг 11: Переключение на ветку master

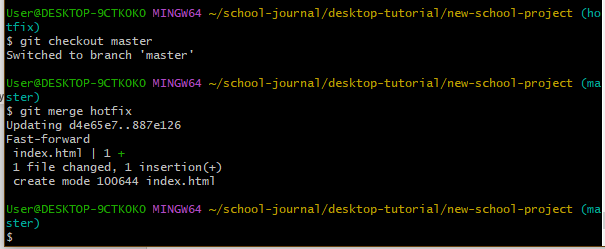


## Шаг 12: Создание ветки hotfix и внесение изменений в index.html

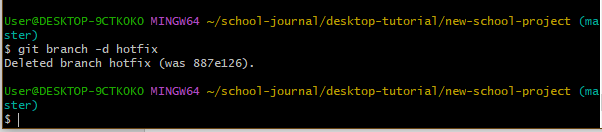




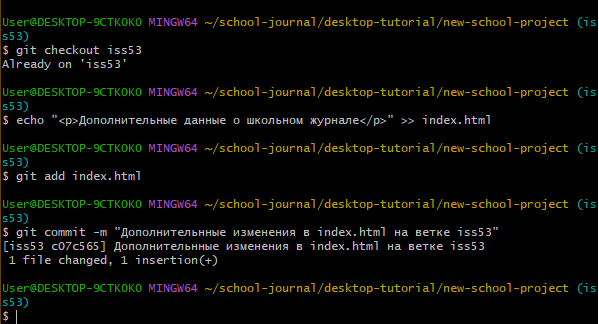
**Шаг 13: Слияние ветки hotfix с master**

****

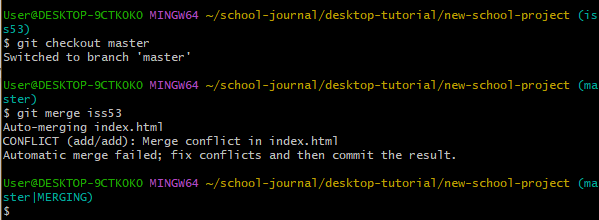
**Шаг 14: Удаление ветки hotfix**

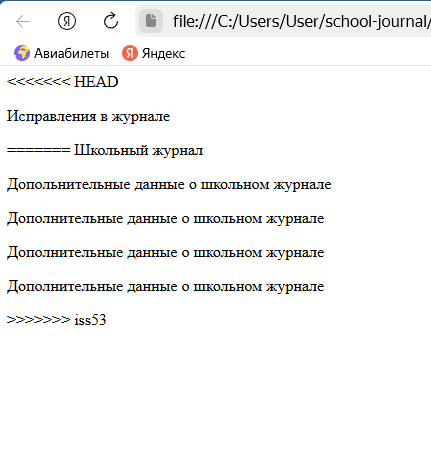


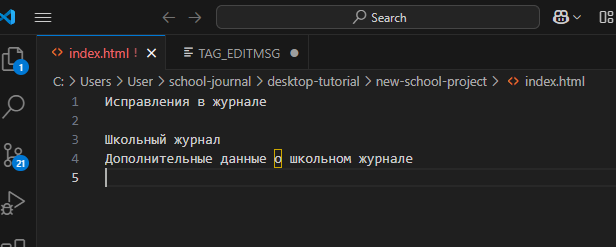
## Шаг 15: Внесение изменений в index.html на ветке iss53



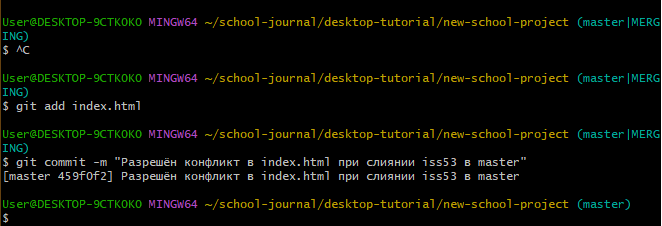
## Шаг 16: Слияние ветки iss53 с master



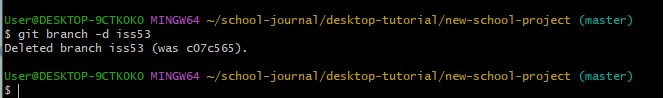


Поменял  


Слиянии ветки.



## Шаг 17: Удаление ветки iss53



## Салимзода Мирзо 9 практика Просмотр списка веток

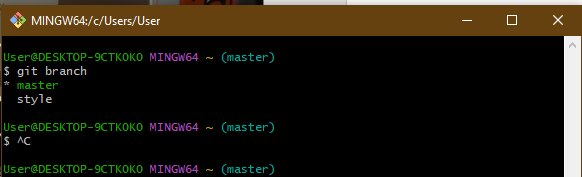
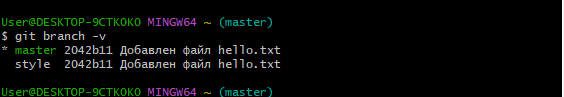


Рисунок 2 – Список веток с последними коммитами



## Фильтрация веток по статусу слияния

Рисунок 3 – Список веток, уже слитых с текущей.

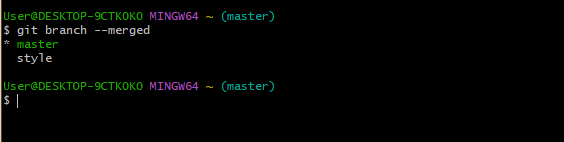


Рисунок 56 - Список веток не слитых с текущей

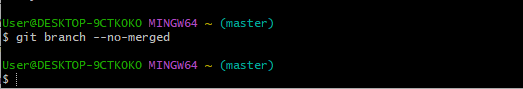


Рисунок 5 – Удаление ветки testing.

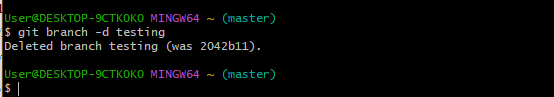
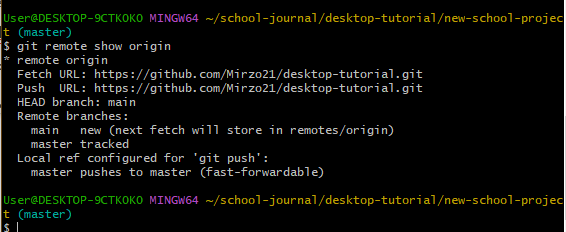


Рисунок 6 – Просмотр удалённых веток



## Отправка изменений на удалённый сервер

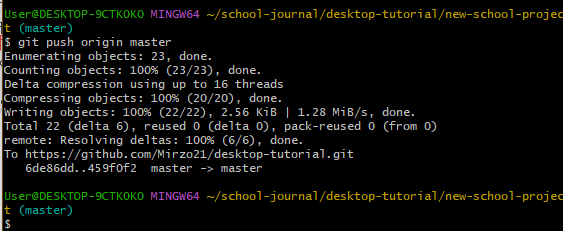
Рисунок 7 – Отправка изменений на сервер.

Рисунок 8 – Выполнение команды git fetch.

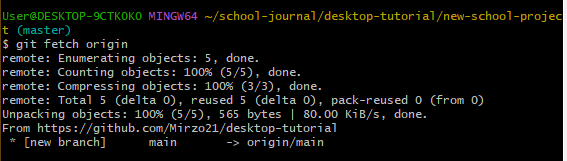


Рисунок 9 – Создание локальной ветки на основе удалённой

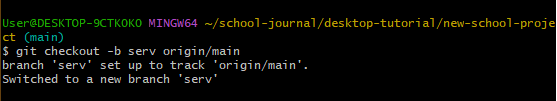
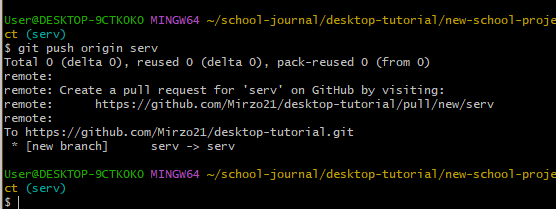


Рисунок 57 - Удаление ветки на сервере



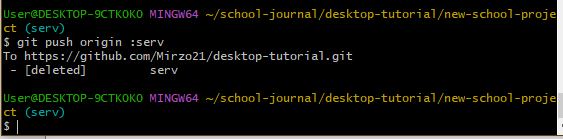
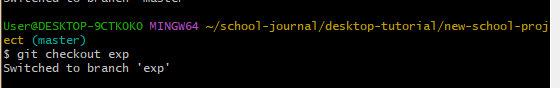


Рисунок 58 - Перемещение изменений



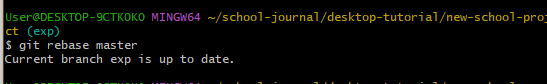


Рисунок 59 - Перемещение изменений с параметром onto

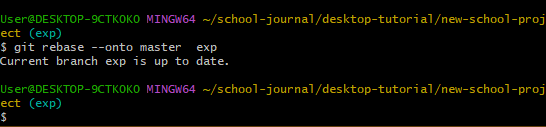


Рисунок 60 - Слияние веток

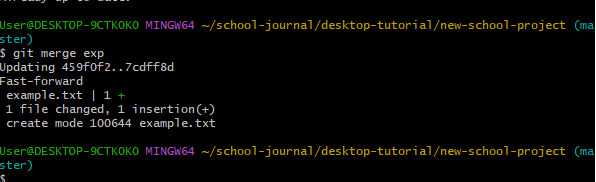


Рисунок 61 - Перемещение изменений

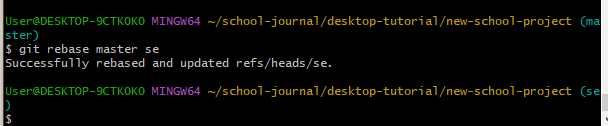
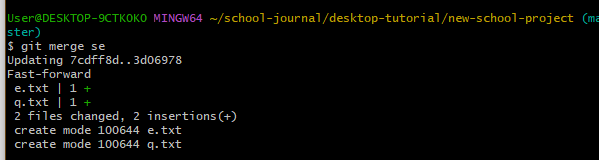


Рисунок 62 - Слияние веток master и se



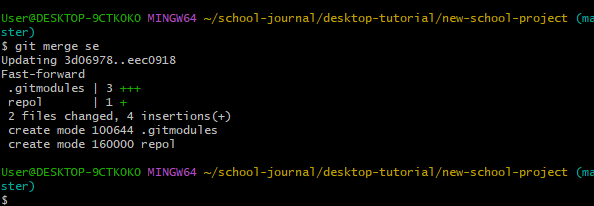


Рисунок 63 - Удаление веток cl и se

