**Git 1.2**

**1. Подготовка**

**Настройка Git**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git config --global user.name "danila.razumnov"**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git config --global user.email "danila.razumnoff@yandex.ru"**

**Параметры окончаний строк**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git config --global core.autocrlf true**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git config --global core.autocrlf input**

**Поддержка Unicode :**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git config --global core.quotepath off**

**2. Создание проекта**

**Шаг 1: Инициализация репозитория**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ mkdir my\_project**

**mkdir: cannot create directory ‘my\_project’: File exists**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ cd my\_project**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ echo "Hello, world" > hello.html**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ git init**

**Reinitialized existing Git repository in C:/Users/Acer/my\_project/.git/**

**Шаг 2: Добавление файла в репозиторий**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ git add hello.html**

**4. Внесение изменений**

**Шаг 1: Редактирование файла**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ cat hello.html**

**Hello, world**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ nano hello.html**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ cat hello.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Hello</title>**

**</head>**

**<body>**

**Hello, world!**

**</body>**

**</html>**

**Шаг 2: Индексация изменений**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ git add hello.html**

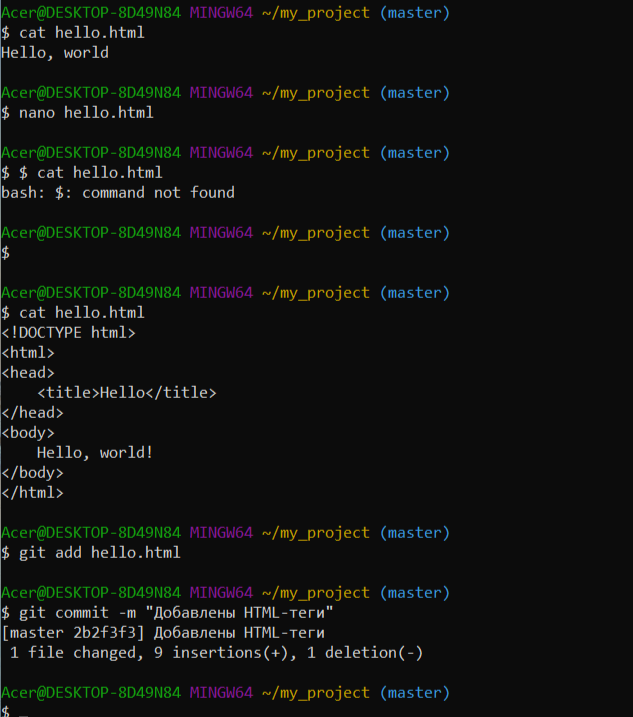
**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**Шаг 3: Коммит изменений**

**$ git commit -m "Добавлены HTML-теги"**

**[master 2b2f3f3] Добавлены HTML-теги**

**1 file changed, 9 insertions(+), 1 deletion(-)**

****

**5. Просмотр истории коммитов**

**Основные команды**

**Полная история**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ git log**

**Краткая история в одну строку**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ git log --pretty=oneline**

**Фильтрация по времени**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ git log --since="2 days ago"**

**commit 2b2f3f3e3dc2227607ad6bb4caa4c526690fd9e7 (HEAD -> master)**

**Author: danila.razumnov <danila.razumnoff@yandex.ru>**

**Date: Thu Jun 5 19:30:48 2025 +0300**

**Добавлены HTML-теги**

**commit 5b426b07ee8616b337712f33deecc7a1bdd412d7**

**Author: Danila Razumnov <danila.razumnoff@yandex.ru>**

**Date: Thu Jun 5 00:55:54 2025 +0300**

**Добавлен тег head**

**commit a898f2c257c8936e745de4321e8530a584c2a15f**

**Author: Danila Razumnov <danila.razumnoff@yandex.ru>**

**Date: Thu Jun 5 00:53:29 2025 +0300**

**Добавлены теги html и body**

**commit 695f46851ee45159602fda7f2d5e1c1e91cb3187**

**Author: Danila Razumnov <danila.razumnoff@yandex.ru>**

**Date: Thu Jun 5 00:46:52 2025 +0300**

**Добавлен тег head**

**commit f66a3531ebdc79a492710bbaa8c09c8df2687936**

**Author: Danila Razumnov <danila.razumnoff@yandex.ru>**

**Date: Thu Jun 5 00:46:32 2025 +0300**

**Добавлены теги html и body**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ git log --until="2021-01-31"**

**По автору**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ git log --author "danila.razumnov"**

**commit 2b2f3f3e3dc2227607ad6bb4caa4c526690fd9e7 (HEAD -> master)**

**Author: danila.razumnov <danila.razumnoff@yandex.ru>**

**Date: Thu Jun 5 19:30:48 2025 +0300**

**Добавлены HTML-теги**

**С отображением изменений**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ git log --stat**

**commit 2b2f3f3e3dc2227607ad6bb4caa4c526690fd9e7 (HEAD -> master)**

**Author: danila.razumnov <danila.razumnoff@yandex.ru>**

**Date: Thu Jun 5 19:30:48 2025 +0300**

**Добавлены HTML-теги**

**hello.html | 10 +++++++++-**

**1 file changed, 9 insertions(+), 1 deletion(-)**

**commit 5b426b07ee8616b337712f33deecc7a1bdd412d7**

**Author: Danila Razumnov <danila.razumnoff@yandex.ru>**

**Date: Thu Jun 5 00:55:54 2025 +0300**

**Добавлен тег head**

**hello.html | 2 +-**

**1 file changed, 1 insertion(+), 1 deletion(-)**

**commit a898f2c257c8936e745de4321e8530a584c2a15f**

**Author: Danila Razumnov <danila.razumnoff@yandex.ru>**

**Date: Thu Jun 5 00:53:29 2025 +0300**

**Добавлены теги html и body**

**hello.html | 2 +-**

**1 file changed, 1 insertion(+), 1 deletion(-)**

**commit 695f46851ee45159602fda7f2d5e1c1e91cb3187**

**Author: Danila Razumnov <danila.razumnoff@yandex.ru>**

**Date: Thu Jun 5 00:46:52 2025 +0300**

**Добавлен тег head**

**hello.html | 2 +-**

**1 file changed, 1 insertion(+), 1 deletion(-)**

**commit f66a3531ebdc79a492710bbaa8c09c8df2687936**

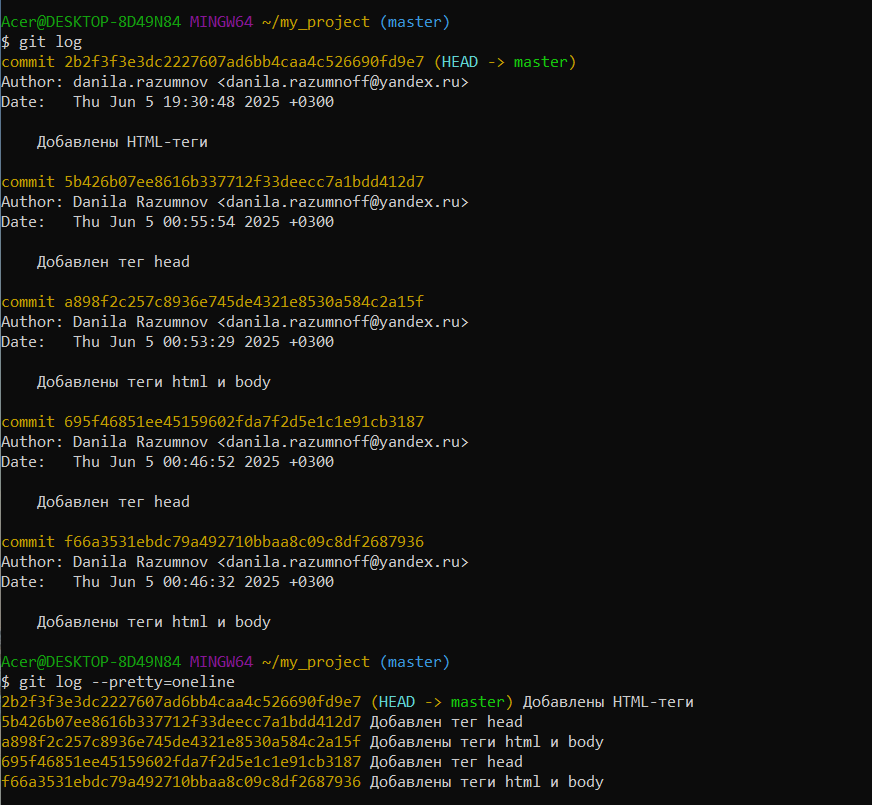
**Author: Danila Razumnov <danila.razumnoff@yandex.ru>**

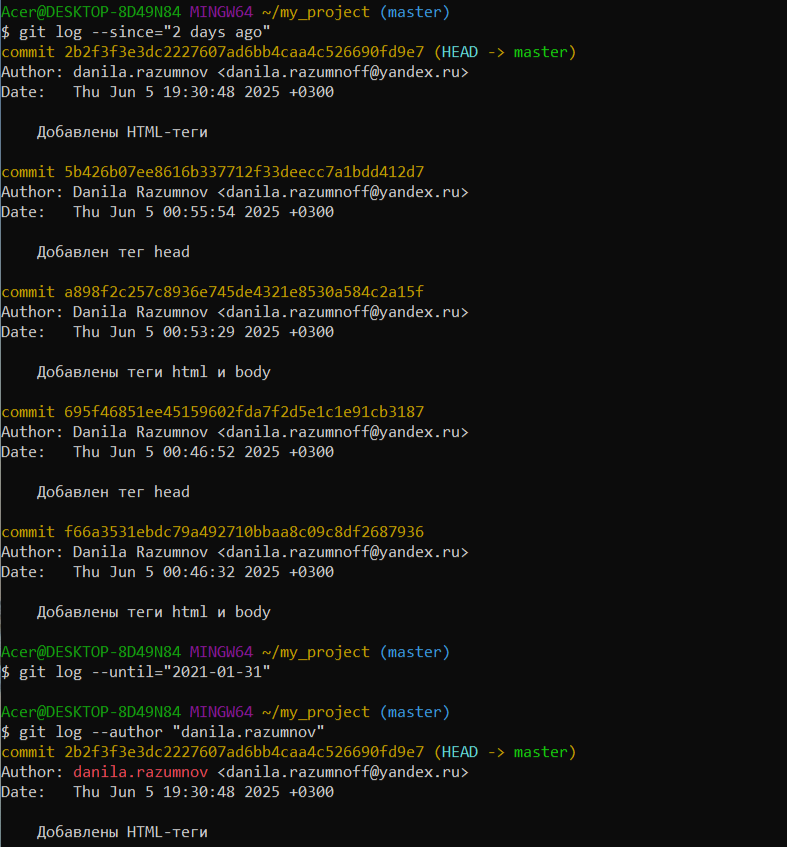
**Date: Thu Jun 5 00:46:32 2025 +0300**

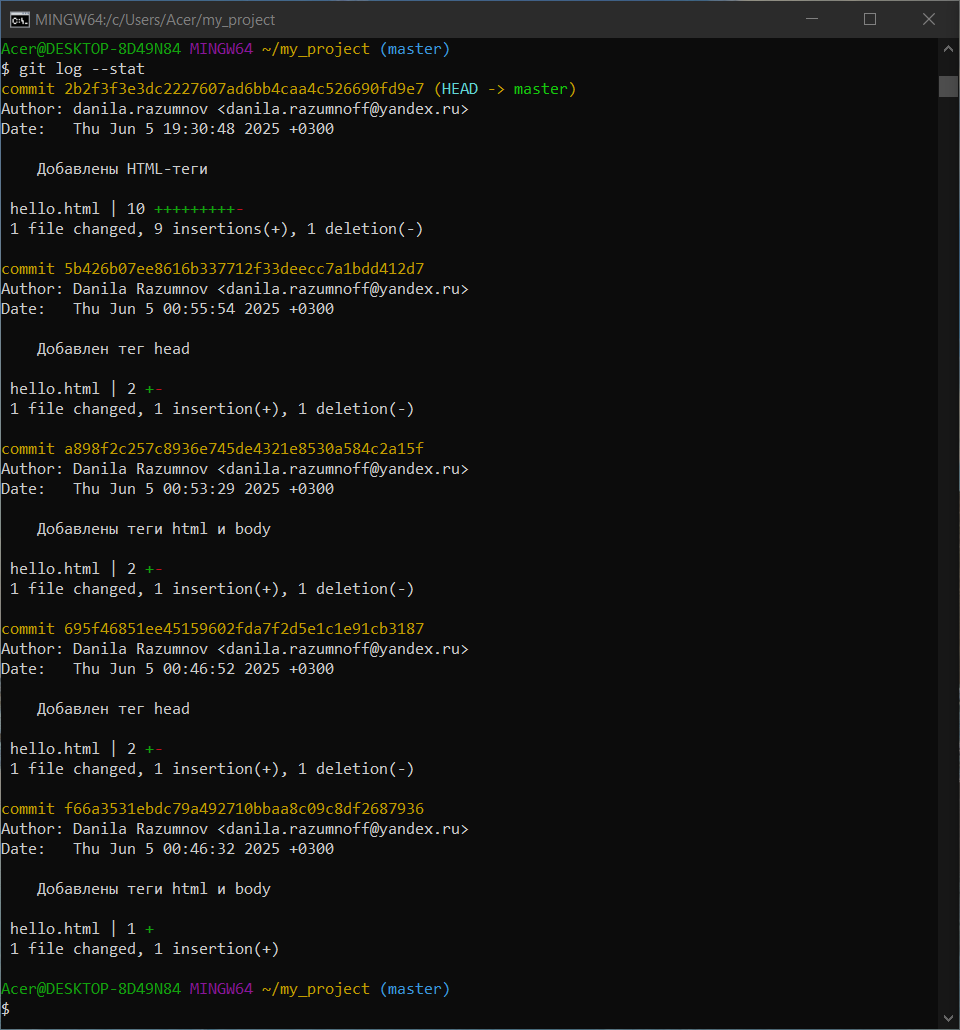
**Добавлены теги html и body**

**hello.html | 1 +**

**1 file changed, 1 insertion(+)**

****

****

****

**6. Работа с тегами**

**Создание тега**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ git tag ver1**

**Просмотр всех тегов**

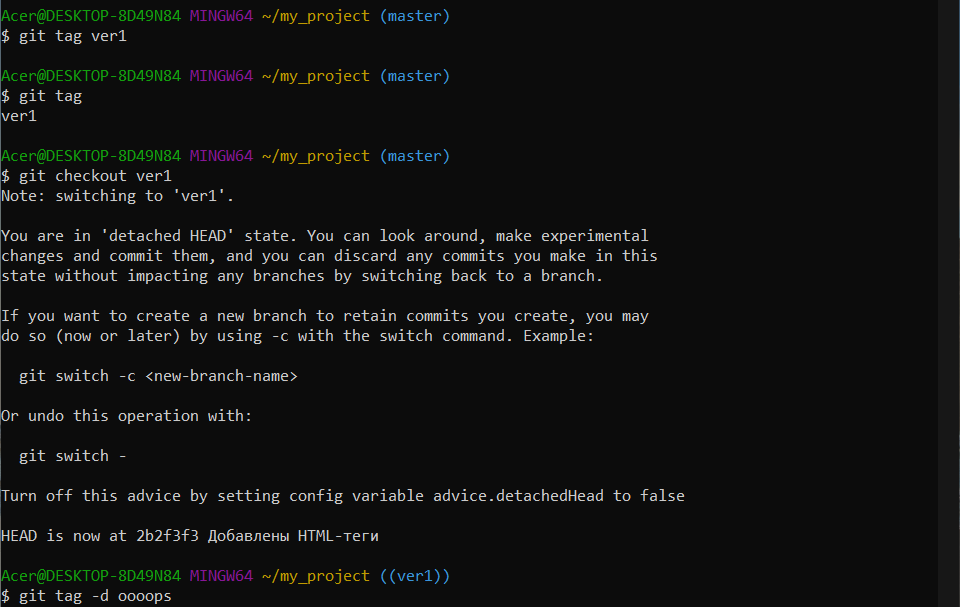
**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ git tag**

**Переключение на тег**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project (master)**

**$ git checkout ver1**

****

**7. Отмена изменений**

**Неиндексированные изменения**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((ver1))**

**$ git checkout hello.html**

**Индексированные изменения**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((ver1))**

**$ git reset HEAD hello.html**

**Отмена последнего коммита**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((ver1))**

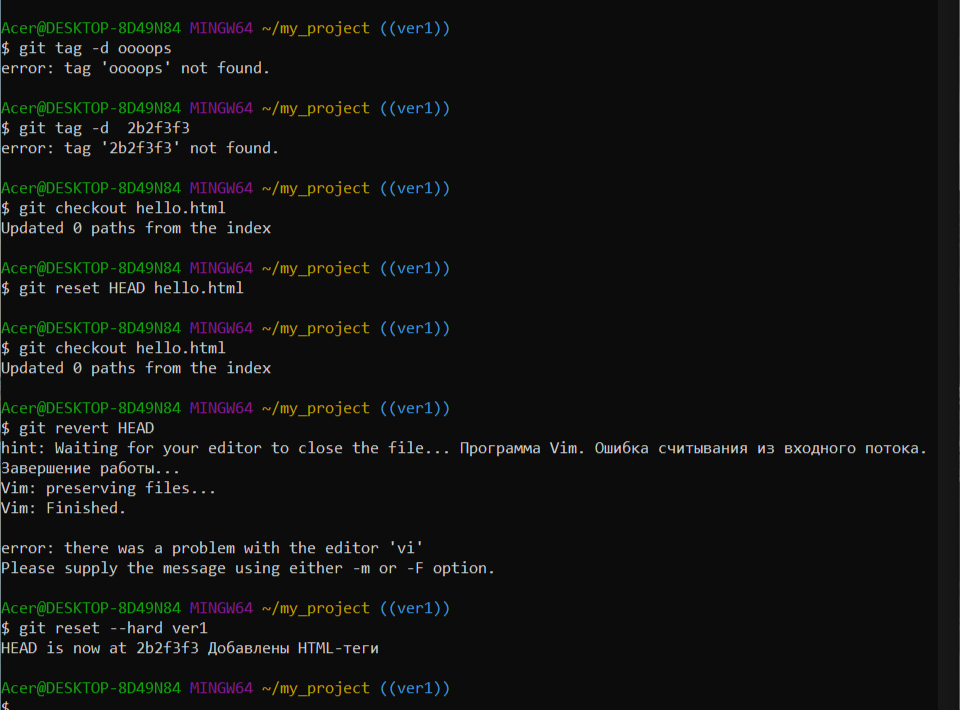
**$ git revert HEAD**

**#Открывается редактор для подтверждения отката**

**Полный сброс до предыдущего коммита**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((ver1))**

**$ git reset --hard ver1**

****

**8. Алиасы для упрощения работы**

**короткие псевдонимы для часто используемых команд:**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((ver1))**

**$ git config --global alias.co checkout**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((ver1))**

**$ git config --global alias.ci commit**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((ver1))**

**$ git config --global alias.st status**

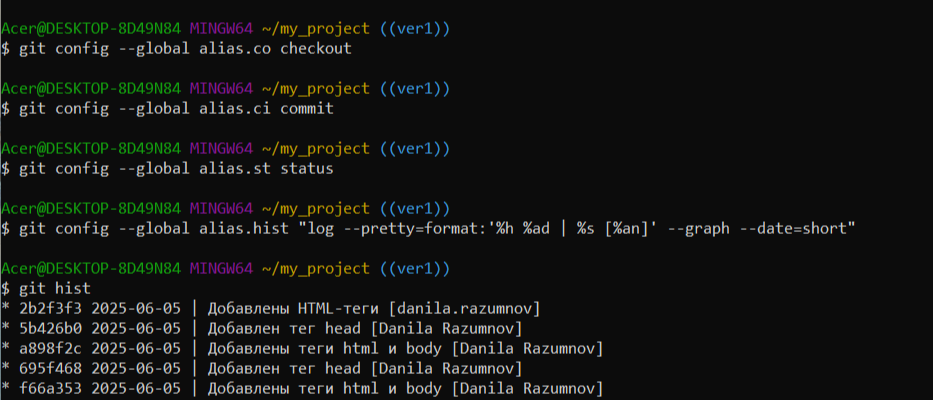
**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((ver1))**

**$ git config --global alias.hist "log --pretty=format:'%h %ad | %s [%an]' --graph --date=short"**

**Теперь вместо git log --pretty=format... Используем**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((ver1))**

**$ git hist**

****

**9. Примеры работы с историей**

**Просмотр изменений**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((ver1))**

**$ git hist**

**Поиск коммита по хэшу**

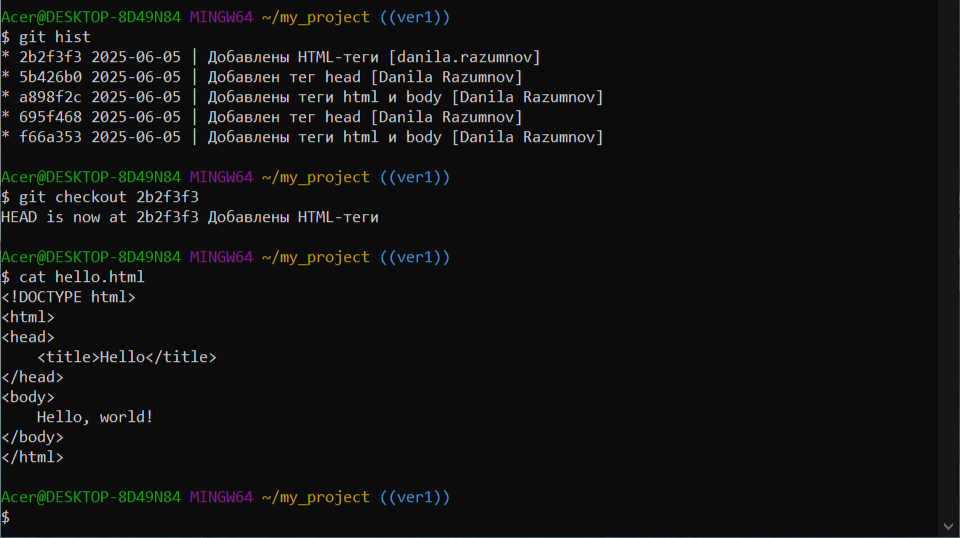
**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((ver1))**

**$ git checkout 2b2f3f3**

**HEAD is now at 2b2f3f3 Добавлены HTML-теги**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((ver1))**

**$ cat hello.html**

****

**10. Удаление коммитов**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((ver1))**

**$ git reset --hard HEAD~1**

**HEAD is now at 5b426b0 Добавлен тег head**

**11. Восстановление удалённых коммитов.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((5b426b0...))**

**$ git reflog**

**5b426b0 (HEAD) HEAD@{0}: reset: moving to HEAD~1**

**2b2f3f3 (tag: ver1, master) HEAD@{1}: checkout: moving from master to ver1**

**2b2f3f3 (tag: ver1, master) HEAD@{2}: commit: Добавлены HTML-теги**

**5b426b0 (HEAD) HEAD@{3}: commit: Добавлен тег head**

**a898f2c HEAD@{4}: commit: Добавлены теги html и body**

**695f468 HEAD@{5}: commit: Добавлен тег head**

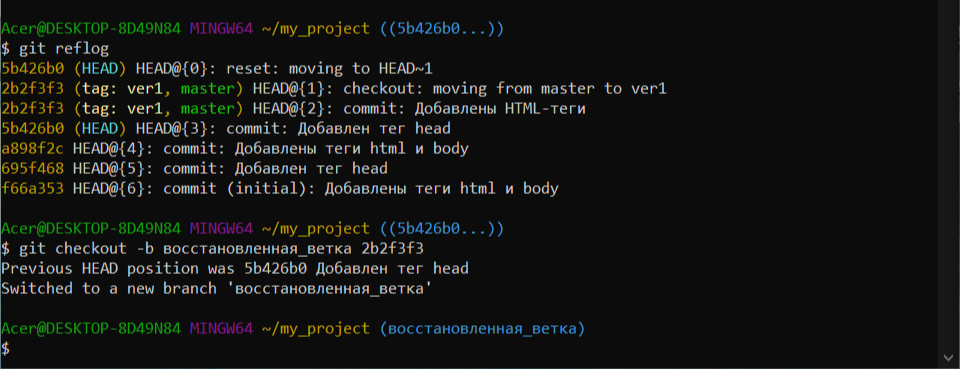
**f66a353 HEAD@{6}: commit (initial): Добавлены теги html и body**

**Acer@DESKTOP-8D49N84 MINGW64 ~/my\_project ((5b426b0...))**

**$ git checkout -b восстановленная\_ветка 2b2f3f3**

**Previous HEAD position was 5b426b0 Добавлен тег head**

**Switched to a new branch 'восстановленная\_ветка'**

****

**Git 1.3**

**1 Добавление автора и email в комментарии**

**Открываю файл hello.html и добавляю комментарий.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ cat hello.html**

**<html><body>Hello, world</body></html>**

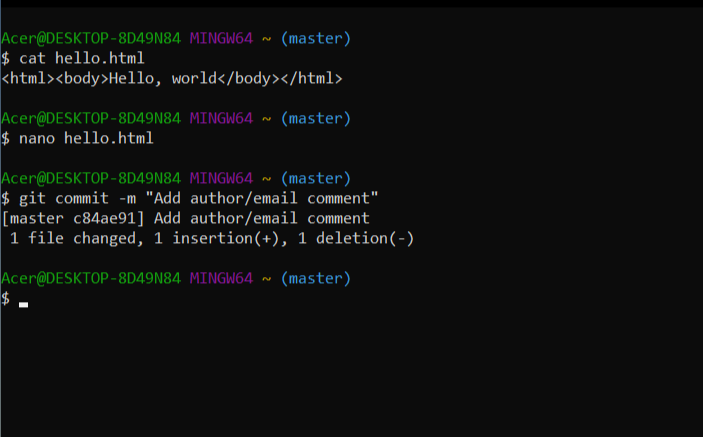
**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ nano hello.html**

**Проиндексирую изменения и создам коммит.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git commit -m "Add author/email comment"**

****

**2. Перемещение файлов**

**Создаю новую директорию:**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ mkdir lib**

**Перемещаю файл с помощью Git:**

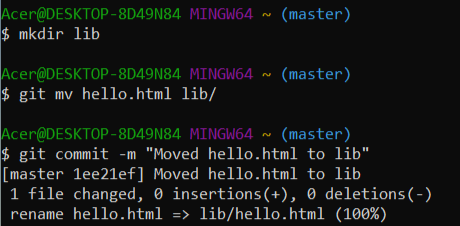
**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git mv hello.html lib/**

**Зафиксирую изменения:**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git commit -m "Moved hello.html to lib"**

****

**3. Создание файла index.html**

**Создаю файл index.html с содержимым.**

**<html>**

**<body>**

**<iframe src="lib/hello.html" width="200" height="200"></iframe>**

**</body>**

**</html>**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ nano index.html**

**Добавляю файл в репозиторий и создаю коммит.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git add index.html**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git commit -m "Added index.html"**

****

**4. Исследование структуры .git**

**Просмотр содержимого .git:**

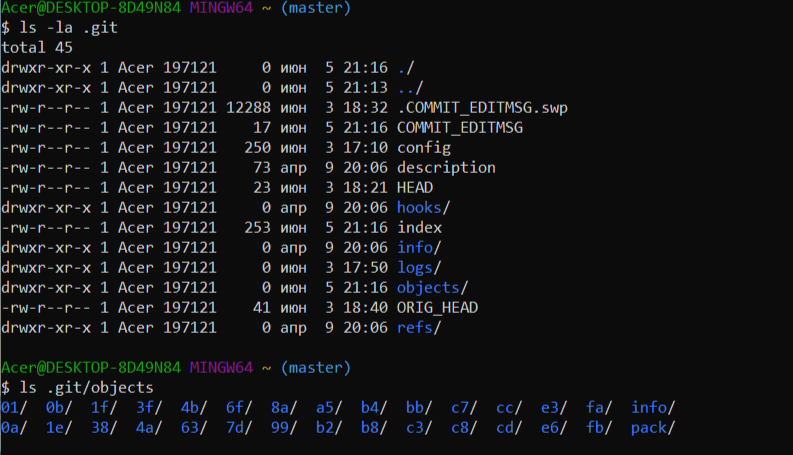
**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ ls -la .git**

**Просмотр хэшей объектов:**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ ls .git/objects**

****

**5. Работа с тегами**

**Создайте тег для текущего коммита:**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git tag ver2**

**Проверьте список тегов:**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git tag**

**Переключитесь на тег:**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git checkout ver2**

****

**7. Просмотр истории коммитов**

**Полная история :**

**Acer@DESKTOP-8D49N84 MINGW64 ~ ((ver2))**

**$ git log**

**Краткий формат**

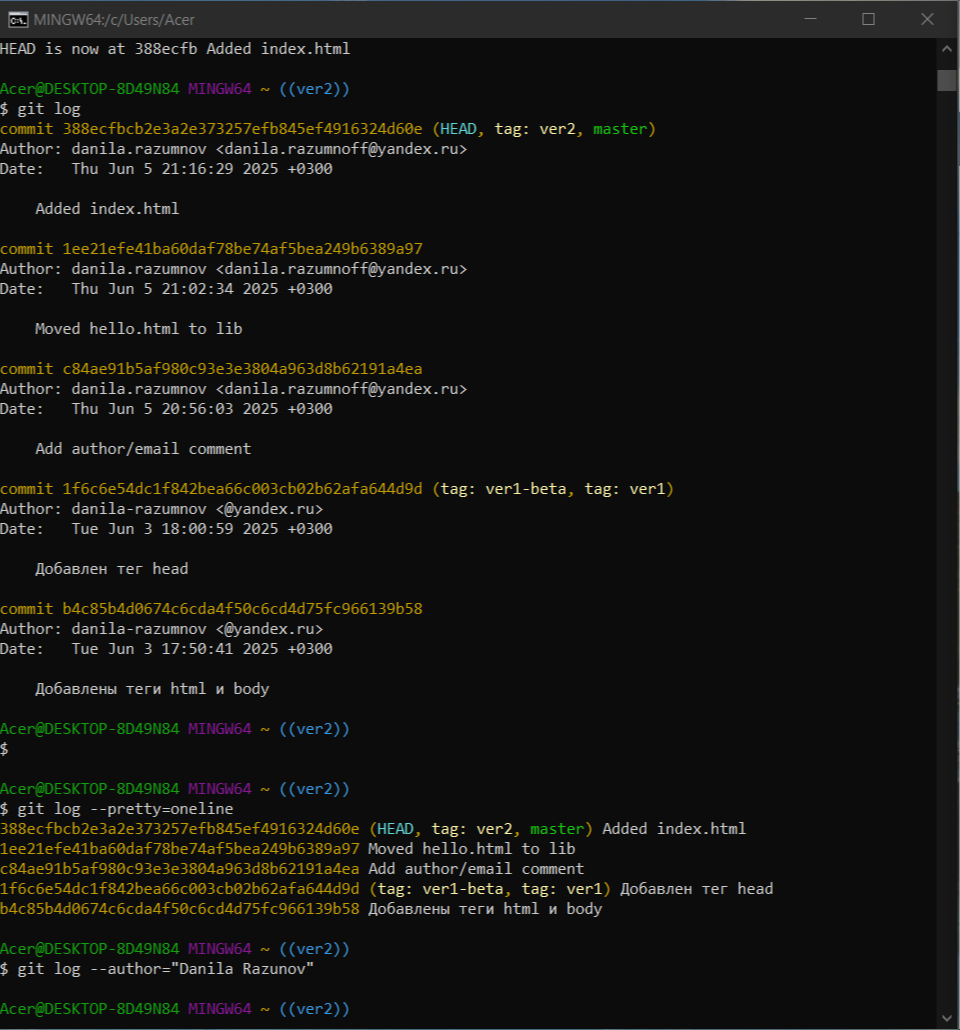
**Acer@DESKTOP-8D49N84 MINGW64 ~ ((ver2))**

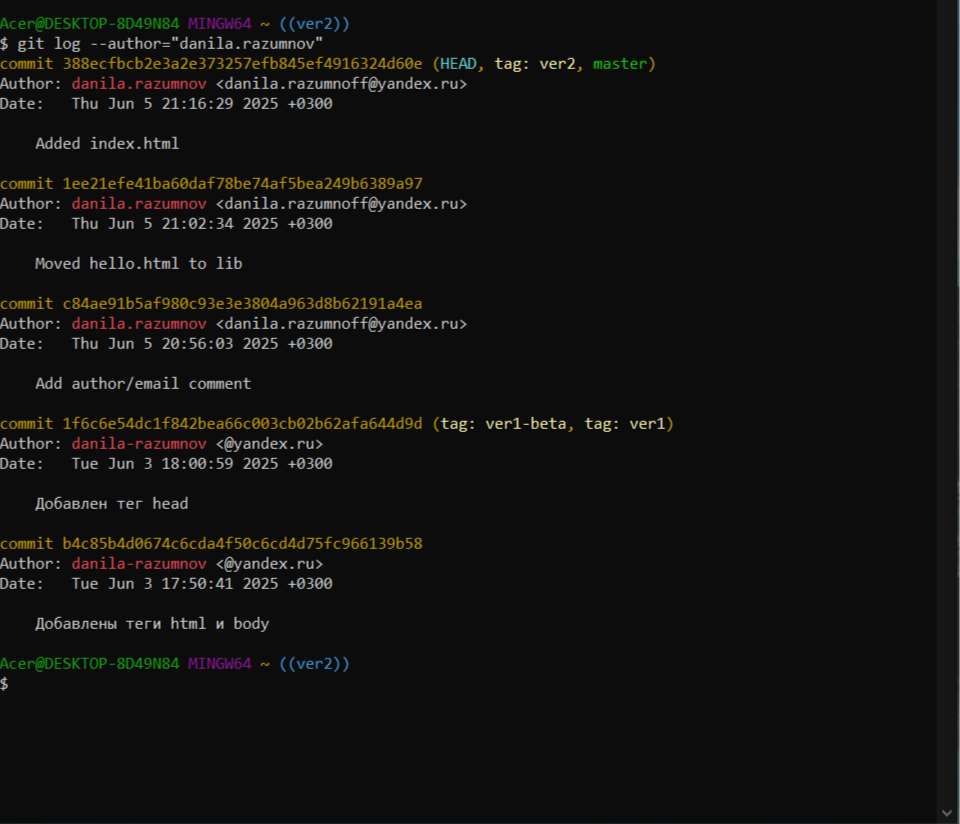
**$ git log --pretty=oneline**

**С фильтром по автору**

**Acer@DESKTOP-8D49N84 MINGW64 ~ ((ver2))**

**$ git log --author="danila.razumnov"**

****

****

**Git 1.4**

**Шаг 1: Подготовка**

**# Создаю папку для проекта.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ mkdir hello**

**mkdir: cannot create directory ‘hello’: File exists**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ cd hello**

**# Инициализирую Git в этой папке.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (new\_branch\_name)**

**$ git init**

**Reinitialized existing Git repository in C:/Users/Acer/hello/.git/**

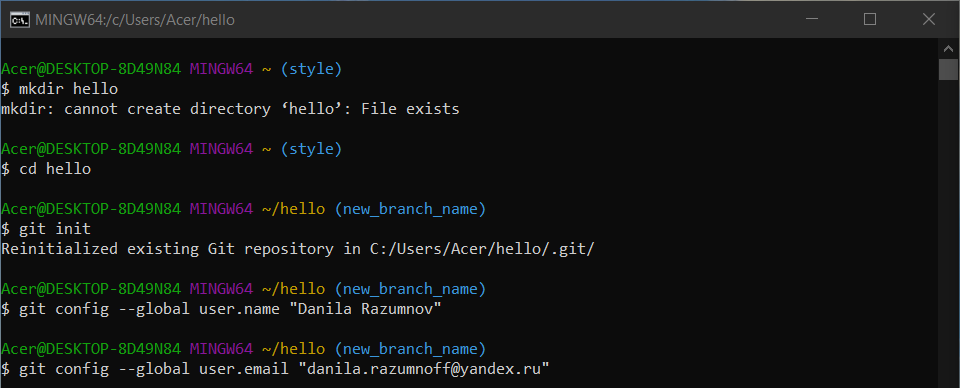
**# Настраиваю имя и email.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (new\_branch\_name)**

**$ git config --global user.name "Danila Razumnov"**

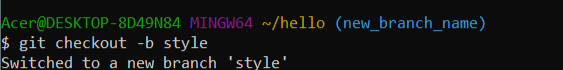
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (new\_branch\_name)**

**$ git config --global user.email "danila.razumnoff@yandex.ru"**

****

**Шаг 2: Создание ветки style**

**# Создаю новую ветку и переключаюсь на неё.**

****

**Шаг 3: Добавление CSS-файла**

**#Создаю директорию lib и файл style.css.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ mkdir -p lib**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ echo "body { color: red; }" > lib/style.css**

**#Добавляю файл в Git**

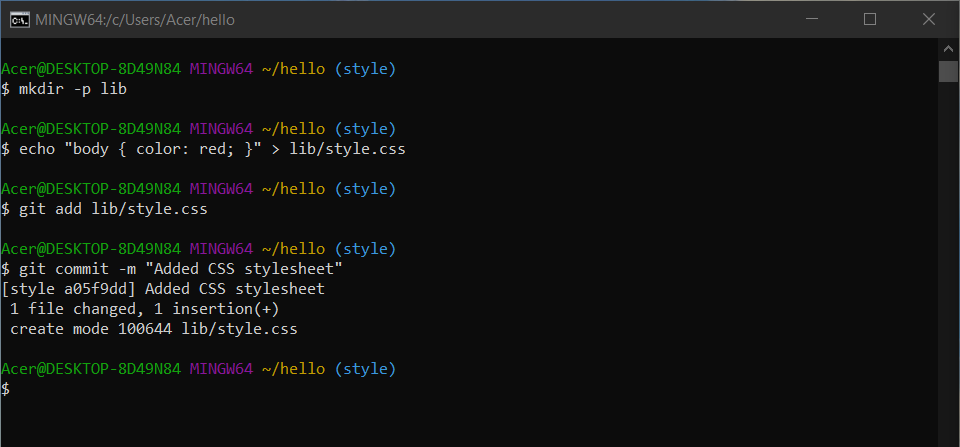
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git add lib/style.css**

**#делаю коммит.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git commit -m "Added CSS stylesheet"**

****

**Шаг 4: Изменение hello.html**

**#Создаю файл lib/hello.html с базовым HTML.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/helloo (style)**

**$ mkdir -p lib**

**Acer@DESKTOP-8D49N84 MINGW64 ~/helloo (style)**

**$ echo "<\!-- Author: Danila Razumnov -->**

**> <html>**

**> <head>**

**> <link rel=\"stylesheet\" type=\"text/css\" href=\"style.css\">**

**> </head>**

**> <body>**

**> <h1>Hello, world</h1>**

**> <h2>The cake is a lie.</h2>**

**> </body>**

**> </html>" > lib/hello.html**

**#Добавляю файл в Git.**

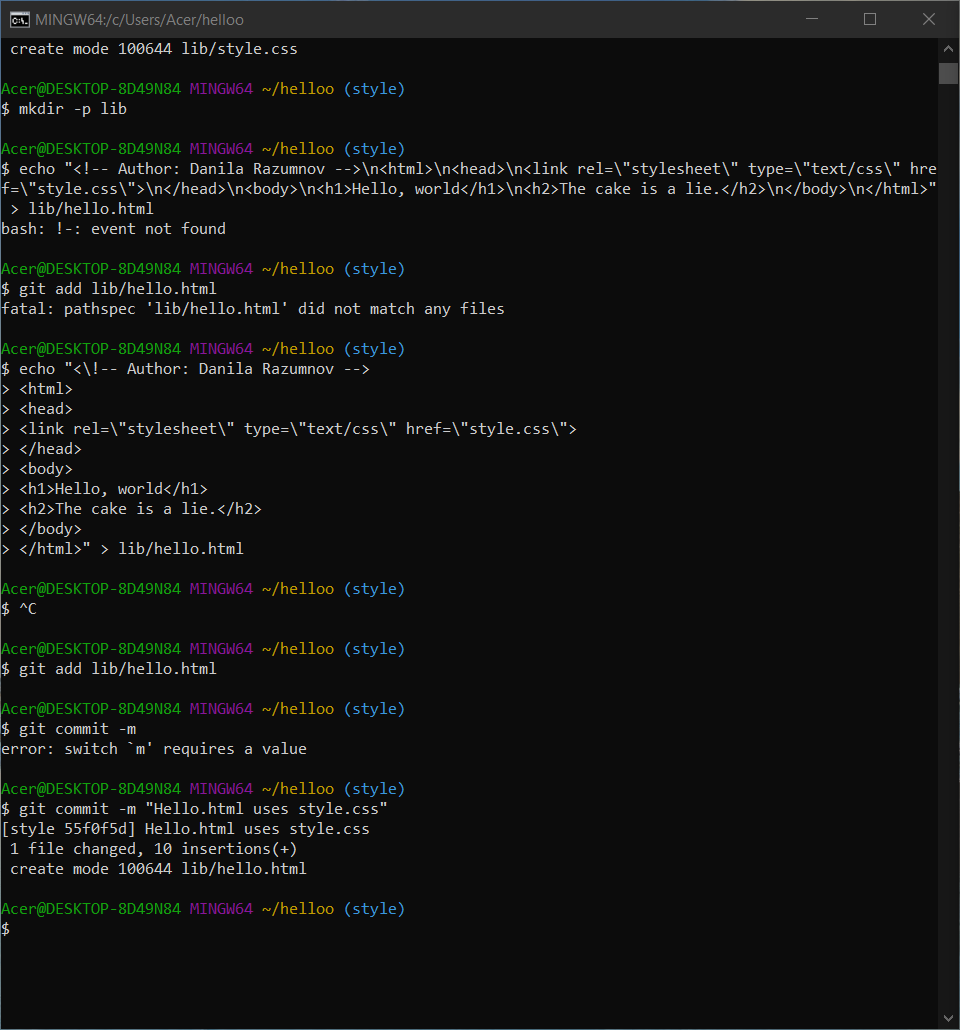
**Acer@DESKTOP-8D49N84 MINGW64 ~/helloo (style)**

**$ git add lib/hello.html**

**#Делаю коммит.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/helloo (style)**

**$ git commit -m "Hello.html uses style.css"**

****

**Шаг 5: Обновление index.html**

**#Создаю файл index.html**

**Acer@DESKTOP-8D49N84 MINGW64 ~/helloo (style)**

**$ echo "<html>\n<body>\n<iframe src=\"lib/hello.html\" width=\"200\" height=\"200\"></iframe>\n</body>\n</html>" > index.html**

**#Добавляю файл в Git**

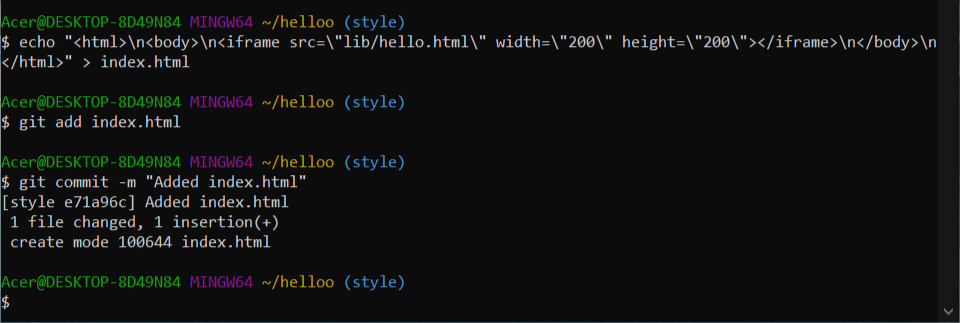
**Acer@DESKTOP-8D49N84 MINGW64 ~/helloo (style)**

**$ git add index.html**

**#Делаю коммит.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/helloo (style)**

**$ git commit -m "Added index.html"**

****

**Шаг 6: Просмотр истории**

**#Устанавливаю алиас для истории**

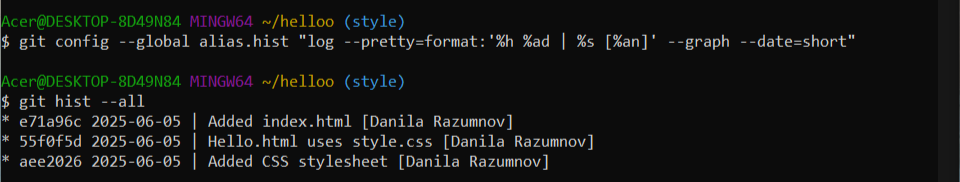
**Acer@DESKTOP-8D49N84 MINGW64 ~/helloo (style)**

**$ git config --global alias.hist "log --pretty=format:'%h %ad | %s [%an]' --graph --date=short"**

**#Проверяю историю.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/helloo (style)**

**$ git hist --all**

****

**Шаг 7: Добавление файла README**

**#Создал файл README.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/helloo (style)**

**$ echo "This is the Hello World example from the Git tutorial" > README**

**#Добавил его в Git.**

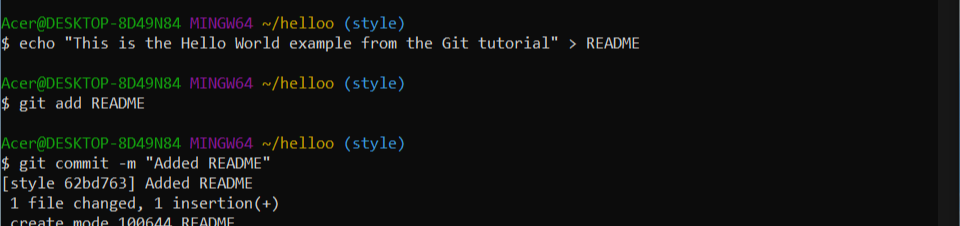
**Acer@DESKTOP-8D49N84 MINGW64 ~/helloo (style)**

**$ git add README**

**#Сделал коммит.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/helloo (style)**

**$ git commit -m "Added README"**

****

**Git 1.5**

**1. Сброс ветки style**

**Цель: Вернуть ветку style к определённому коммиту.**

**1. Проверяю историю коммитов.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ git hist**

**2. Переключаюсь на ветку style.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ git checkout style**

**3. Сбрасываю ветку до нужного коммита.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ git reset --hard b4c85b4**

****

**2. Сброс ветки master**

**Цель: Вернуть ветку master к определённому коммиту.**

1. **Переключаюсь на ветку master.**

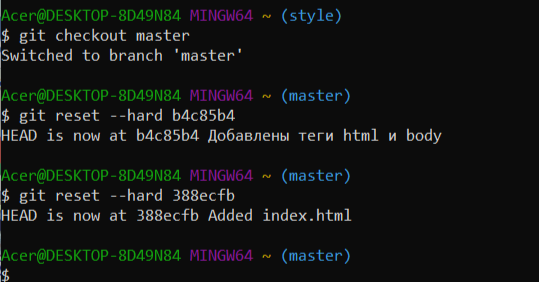
**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ git checkout master**

1. **Сбрасываю ветку до нужного коммита.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git reset --hard 388ecfb**

****

**3. Перебазирование (Rebase)**

**Цель: Обновить ветку style, перенеся её коммиты поверх коммитов ветки master.**

**1. Переключаюсь на ветку style.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git checkout style**

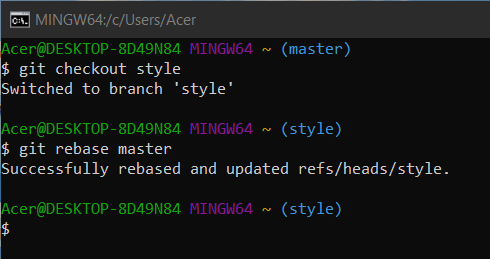
**Switched to branch 'style'**

**2. Выполняю rebase.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ git rebase master**

**Successfully rebased and updated refs/heads/style.**

****

**4. Слияние веток**

**Цель: Слить изменения из style в master.**

**1. Слил ветку style.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ git checkout master**

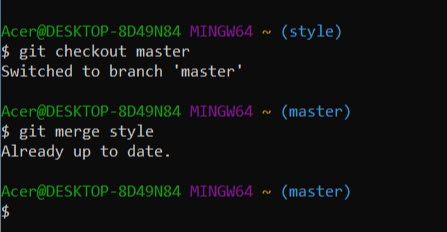
**Switched to branch 'master'**

**2. Переключился на ветку master.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git merge style**

**Already up to date.**

****

**5. . Просмотр клонированного репозитория**

**Цель: Проверить содержимое и историю клонированного репозитория.**

1. **Перехожу в папку репозитория.**

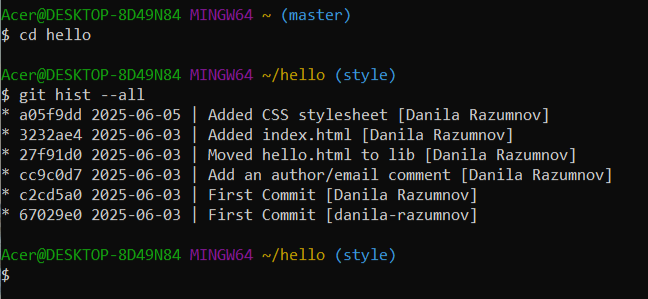
**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ cd hello**

1. **Проверяю историю.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git hist --all**

****

**6 Работа с origin**

**Цель: Просмотр информации об удалённом репозитории (origin).**

1. **Проверяю remote-ссылки.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git remote -v**

1. **Получаю данные о origin.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git remote show origin**

****

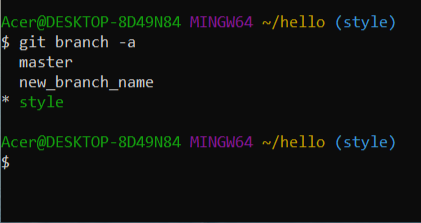
**7. Работа с удалёнными ветками**

**Цель: Просмотр локальных и удалённых веток.**

**Показать все ветки.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git branch -a**

****

**Git 1.5**

**1. Сброс ветки style**

**Цель: Вернуть ветку style к определённому коммиту.**

**1. Проверяю историю коммитов.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ git hist**

**2. Переключаюсь на ветку style.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ git checkout style**

**3. Сбрасываю ветку до нужного коммита.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ git reset --hard b4c85b4**

****

**2. Сброс ветки master**

**Цель: Вернуть ветку master к определённому коммиту.**

1. **Переключаюсь на ветку master.**

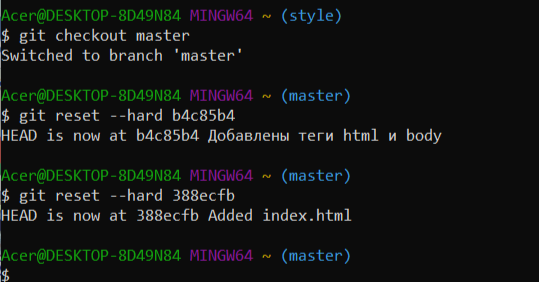
**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ git checkout master**

1. **Сбрасываю ветку до нужного коммита.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git reset --hard 388ecfb**

****

**3. Перебазирование (Rebase)**

**Цель: Обновить ветку style, перенеся её коммиты поверх коммитов ветки master.**

**1. Переключаюсь на ветку style.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git checkout style**

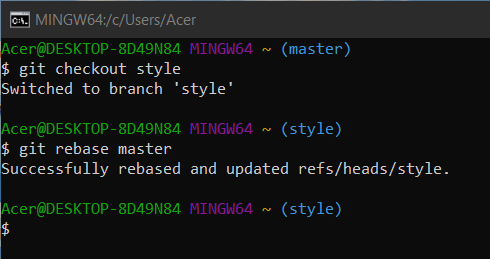
**Switched to branch 'style'**

**2. Выполняю rebase.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ git rebase master**

**Successfully rebased and updated refs/heads/style.**

****

**4. Слияние веток**

**Цель: Слить изменения из style в master.**

**1. Слил ветку style.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (style)**

**$ git checkout master**

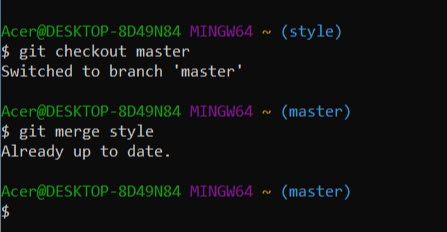
**Switched to branch 'master'**

**2. Переключился на ветку master.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git merge style**

**Already up to date.**

****

**5. . Просмотр клонированного репозитория**

**Цель: Проверить содержимое и историю клонированного репозитория.**

1. **Перехожу в папку репозитория.**

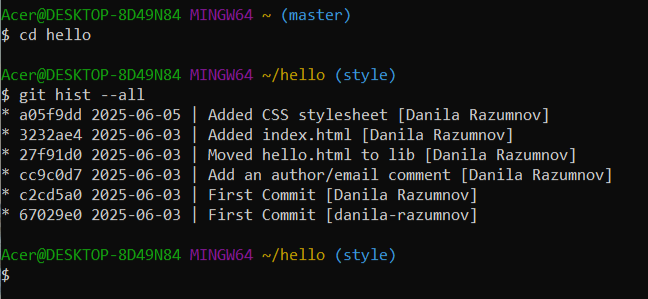
**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ cd hello**

1. **Проверяю историю.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git hist --all**

****

**6 Работа с origin**

**Цель: Просмотр информации об удалённом репозитории (origin).**

1. **Проверяю remote-ссылки.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git remote -v**

1. **Получаю данные о origin.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git remote show origin**

****

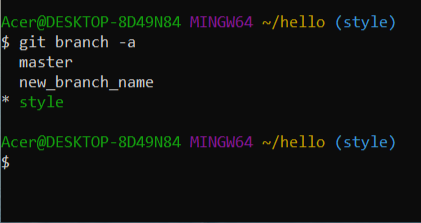
**7. Работа с удалёнными ветками**

**Цель: Просмотр локальных и удалённых веток.**

**Показать все ветки.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git branch -a**

****

**Git 1.6**

**1. Настройка удалённого репозитория**

**Цель: Связать локальный репозиторий с удалённым (например, GitHub, GitLab).**

**Добавляю удалённый репозиторий.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git remote remove origin**

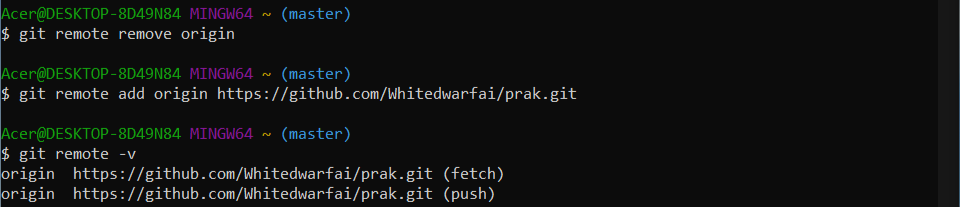
**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git remote add origin** [**https://github.com/Whitedwarfai/prak.git**](https://github.com/Whitedwarfai/prak.git)

**Проверяю список удалённых репозиториев.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git remote -v**

****

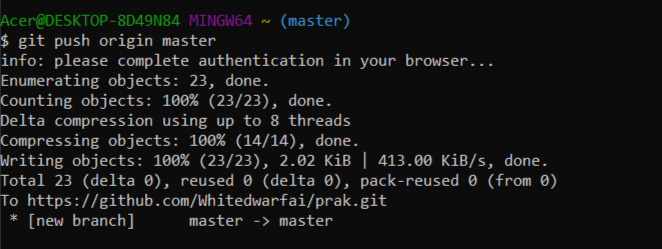
**2. Отправка изменений на удалённый репозиторий**

**Цель: Загрузите локальные коммиты на сервер**

**Отправляю изменения на ветку master.**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ git push origin master**

****

#### **3. Работа с общей директорией (shared repository)**

**Цель: Использовать общий репозиторий для командной работы.**

#### **я создал общий репозиторий**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ mkdir ../hello.git**

**Acer@DESKTOP-8D49N84 MINGW64 ~ (master)**

**$ cd ../hello.git**

**Acer@DESKTOP-8D49N84 MINGW64 /c/Users/hello.git**

**$ git init --bare**

**Я связал локальный репозиторий с shared.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello.git (BARE:master)**

**$ cd ../hello**

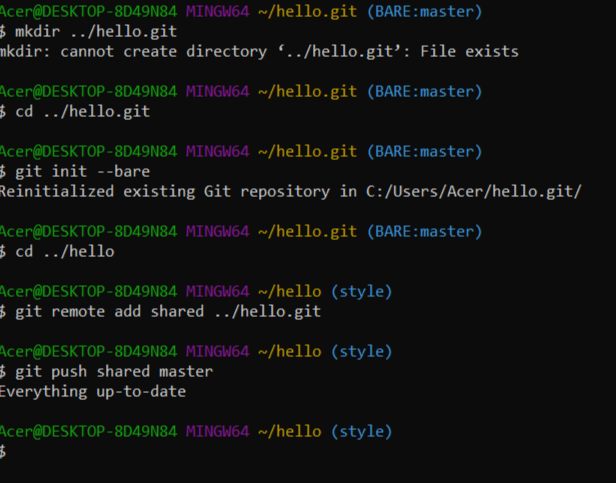
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git remote add shared ../hello.git**

**Я отправил изменения в shared.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git push shared master**

****

**4. Обновление локального репозитория**

**Цель: Получаю изменения от других участников.**

#### **Забираю изменения с удалённого репозитория.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git pull origin master**

**Когда использую shared repository.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git pull shared master**

****

#### **5. Работа с ветками в команде**

**Цель: Я создаю ветки для задач и сливаю их в основную ветку.**

#### **# Создаю новую ветку для задачи**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (style)**

**$ git checkout -b feature-login**

**# Вношу изменения, индексирую и делаю коммит**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (feature-login)**

**$ git add .**

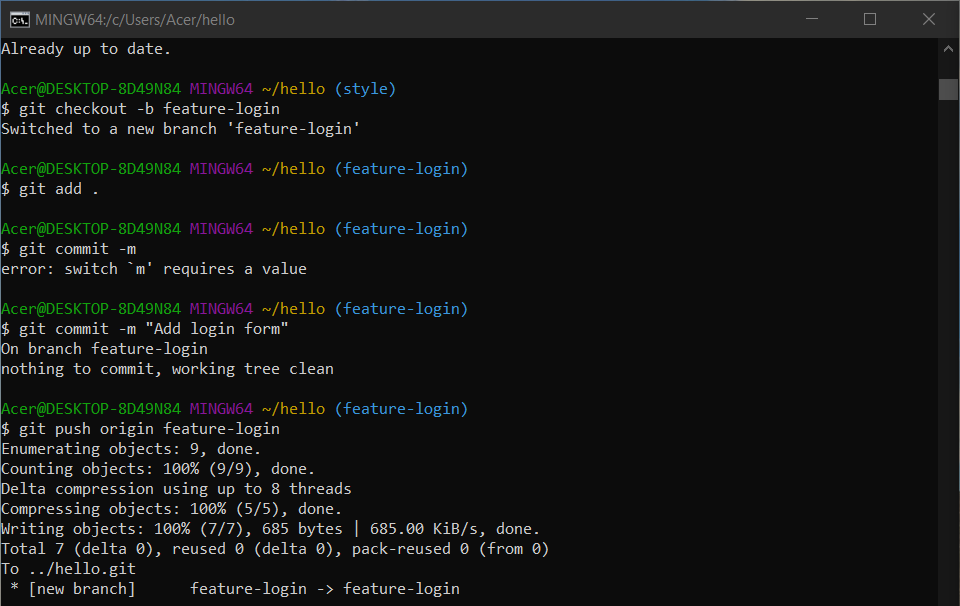
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (feature-login)**

**$ git commit -m "Add login form"**

**# Отправляю ветку на удалённый репозиторий.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (feature-login)**

**$ git push origin feature-login**

****

#### **6. Разрешение конфликтов при совместной работе**

**Цель: Я устраняю конфликты, возникшие при слиянии изменений.**

#### **Попытайтесь слить ветки**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (feature-login)**

**$ git merge feature-login**

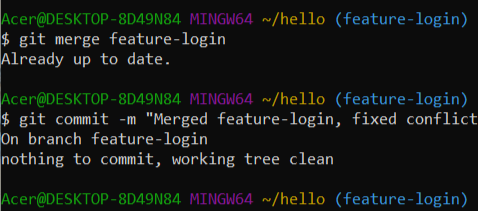
**Already up to date.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (feature-login)**

**$ git commit -m "Merged feature-login, fixed conflict"**

**On branch feature-login**

**nothing to commit, working tree clean**

****

#### 

#### **7. Просмотр изменений из удалённого репозитория**

**Цель: Я проверяю историю и ветки из удалённого репозитория.**

**# Показать историю из origin**

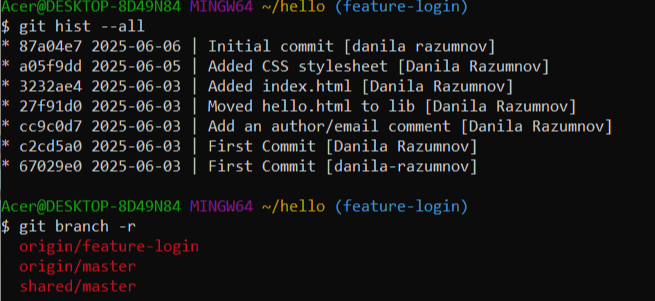
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (feature-login)**

**$ git hist --all**

**# Показать ветки из origin**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (feature-login)**

**$ git branch -r**

****

#### **8. Работа с ветками других участников**

**Цель: Получить и протестировать чужие ветки.**

#### **Получите данные о всех ветках**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (feature-login)**

**$ git fetch origin**

**Для начала я создам локальную ветку из удалённой.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (feature-login)**

**$ git checkout -b remote-feature origin/feature-login**

****

#### **9. Отслеживание изменений в реальном времени**

**Цель: Я слежу за активностью в репозитории.**

### **# Проверяю обновления на origin**

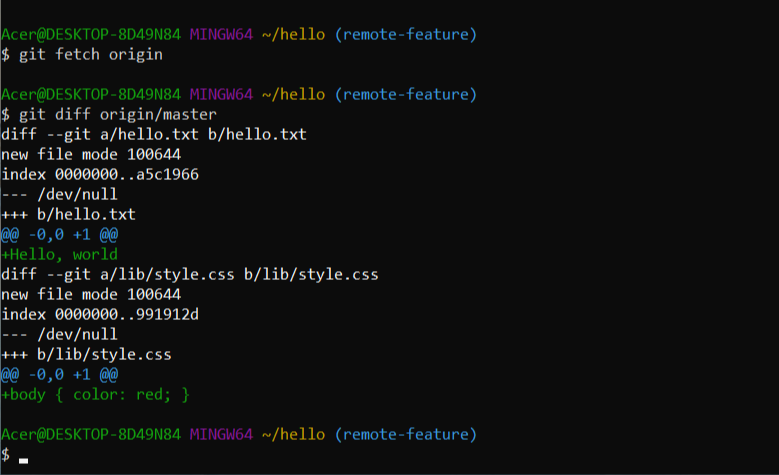
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (remote-feature)**

**$ git fetch origin**

**# Я сравниваю локальную ветку с origin/master.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (remote-feature)**

**$ git diff origin/master**

****

**Git 1.7**

#### **1. Добавление удалённого репозитория (remote)**

**Цель: Привяжу локальный репозиторий к удалённому (например, GitHub).**

**# Добавляю удалённый репозиторий с именем `origin`**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (remote-feature)**

**$ git remote remove origin**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (remote-feature)**

**$ git remote add origin https://github.com/Whitedwarfai/prak.git**

**# Проверяю список удалённых репозиториев**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (remote-feature)**

**$ git remote -v**

****

#### **2. Получение данных из удалённого репозитория (fetch)**

**Цель: Заберу данные из удалённого репозитория без слияния.**

**# Получаю данные из `origin`**

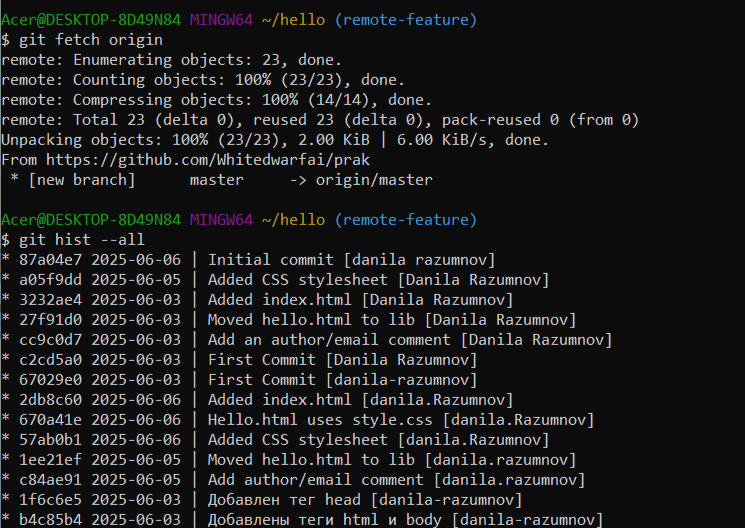
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (remote-feature)**

**$ git fetch origin**

**# Проверяю историю после fetch**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (remote-feature)**

**$ git hist --all**

****

#### **3. Отправка изменений на удалённый репозиторий (push)**

**Цель: Загружу локальные коммиты на сервер.**

**# Отправляю ветку `master` на `origin`**

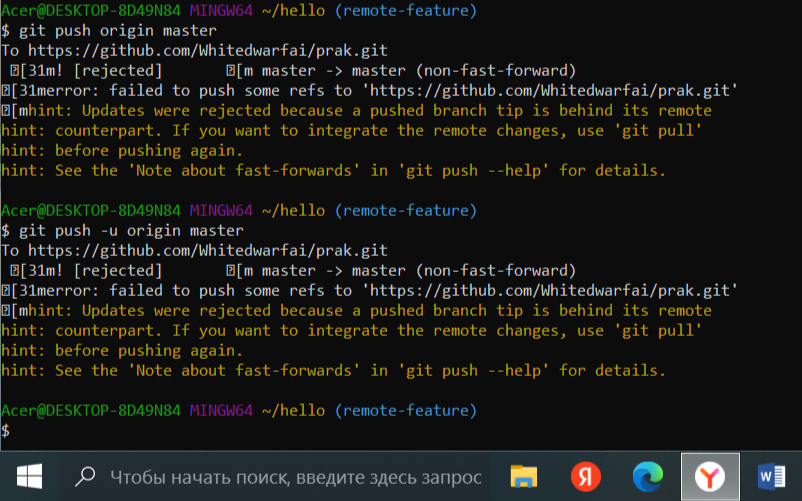
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (remote-feature)**

**$ git push origin master**

**# Если ветка ещё не существует на сервере, использую:**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (remote-feature)**

**$ git push -u origin master**

****

#### **4. Работа с тегами**

**Цель: Создаю теги, переключаюсь на них и создаю ветки из тегов.**

**# Создаю аннотированный тег**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (remote-feature)**

**$ git tag -a v2.1 -m "Release version 2.1"**

**# Проверяю список тегов**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (remote-feature)**

**$ git tag**

**# Переключаюсь на коммит по тегу**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (remote-feature)**

**$ git checkout v2.1**

**# Создаю новую ветку из тега v2.1**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello ((v2.1))**

**$ git switch -c release-2.1 v2.1**

****

#### **5. Работа с несколькими удалёнными репозиториями**

**Цель: Добавляю второй удалённый репозиторий.**

**# Добавляю второй remote**

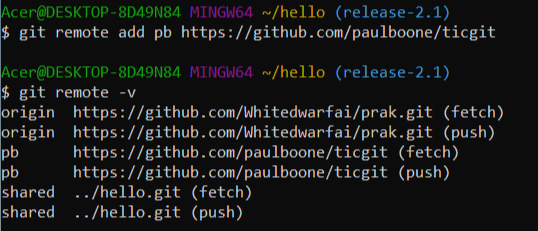
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (release-2.1)**

**$ git remote add pb https://github.com/paulboone/ticgit**

**# Проверяю список remote**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (release-2.1)**

**$ git remote -v**

****

##### **Получение данных из второго remote**

**#Забираю изменения из pb.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (release-2.1)**

**$ git fetch pb**

**# Проверяю ветки из `pb`**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (release-2.1)**

**$ git branch -r**

****

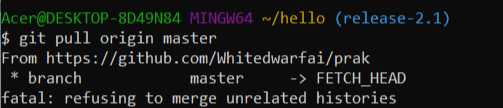
#### **6. Синхронизация с удалённым репозиторием (pull)**

**Цель: Получаю изменения из удалённого репозитория и сливаю их с текущей веткой.**

**# Синхронизирую текущую ветку с `origin`**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (release-2.1)**

**$ git pull origin master**

****

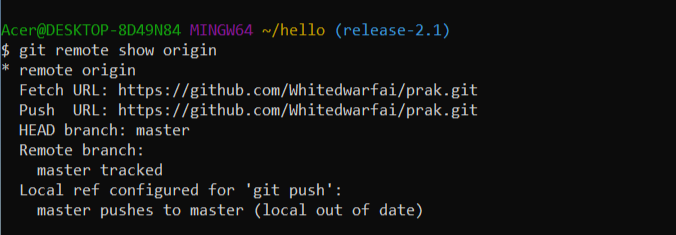
#### **7. Просмотр информации о remote**

**Цель: Узнаю детали о удалённых репозиториях.**

**# Показываю информацию о `origin`**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (release-2.1)**

**$ git remote show origin**

****

#### **8. Удаление тега**

**Цель: Удаляю ненужный тег.**

**# Удаляю локальный тег**

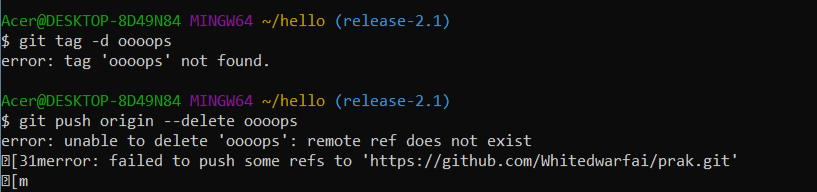
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (release-2.1)**

**$ git tag -d oooops**

**# Удаляю тег из удалённого репозитория**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (release-2.1)**

**$ git push origin --delete oooops**

****

#### **9. Работа с detached HEAD**

**Цель: Выход из состояния detached HEAD или создание ветки из тега.**

**# Переключаюсь на тег**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (release-2.1)**

**$ git checkout v2.1**

**# Создаю новую ветку**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello ((v2.1))**

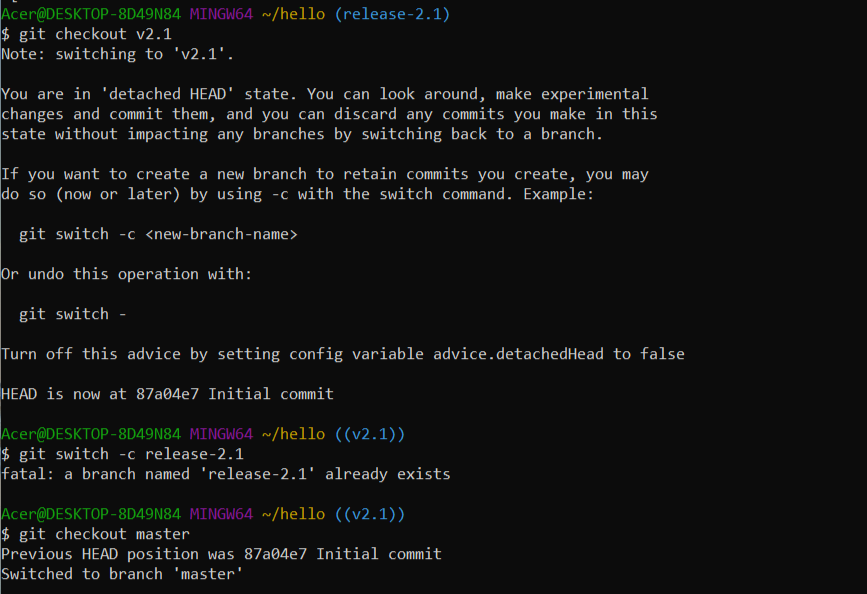
**$ git switch -c release-2.1**

##### **Выход из detached HEAD**

**# Возвращаюсь к ветке `master`**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello ((v2.1))**

**$ git checkout master**

****

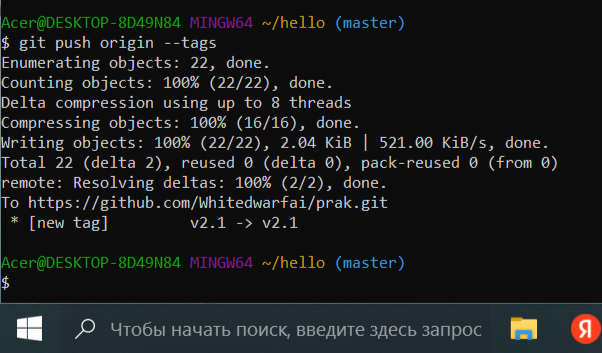
#### **10. Отправка тегов на удалённый репозиторий**

**Цель: Загружаю теги на сервер.**

**# Отправляю все теги на `origin`**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (master)**

**$ git push origin --tags**

****

**Git 1,8**

#### **1. Создание проекта**

##### **Шаг 1: Инициализация репозитория**

**Я инициализирую репозиторий.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (master)**

**$ mkdir hello\_two**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello (master)**

**$ cd hello\_two**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (master)**

**$ git init**

##### **Шаг 2: Добавление файлов**

**Я создаю файлы README, test.rb, LICENSE и добавляю их в Git.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (master)**

**$ echo "This is the Hello World example from the Git tutorial" > README**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (master)**

**$ echo "# Test file" > test.rb**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (master)**

**$ echo "MIT License" > LICENSE**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (master)**

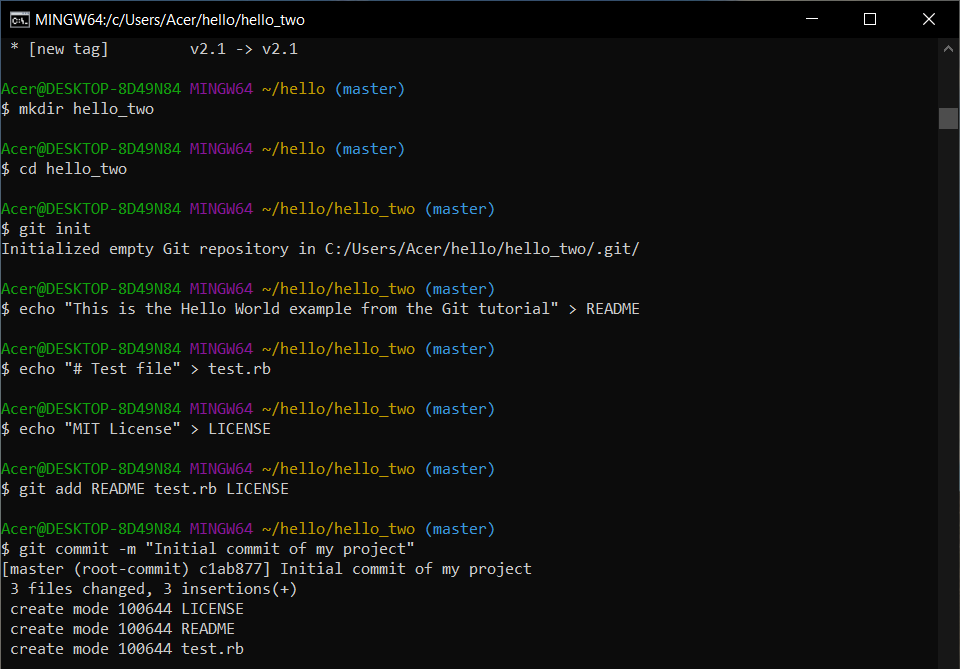
**$ git add README test.rb LICENSE**

##### **Шаг 3: Коммит изменений**

**Я фиксирую начальные файлы.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (master)**

**$ git commit -m "Initial commit of my project"**

****

#### **2. Создание ветки для задачи**

##### **Цель: Создать ветку iss53 для работы над конкретной задачей.**

**# Я создаю и переключаюсь на ветку iss53.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (master)**

**$ git checkout -b iss53**

##### **Пример работы с веткой iss53:**

**Я добавляю файл index.html.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (iss53)**

**$ echo "<html><body><h1>Hello, world</h1></body></html>" > index.html**

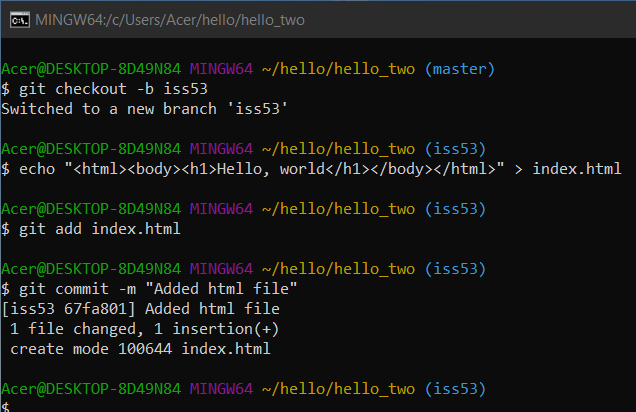
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (iss53)**

**$ git add index.html**

**Я фиксирую изменения.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (iss53)**

**$ git commit -m "Added html file"**

****

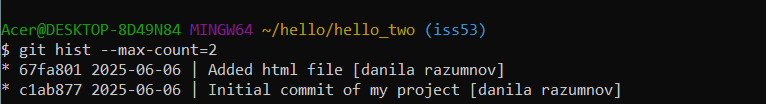
**3. Просмотр истории коммитов**

##### **Команда git hist**

**Я проверяю историю коммитов через алиас hist.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (iss53)**

**$ git hist --max-count=2**

****

#### **4. Переключение между ветками**

##### **Вернитесь к ветке master:**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (iss53)**

**$ git checkout master**

##### **Создайте ветку из удалённой ветки**

**# Я связываю локальную ветку с удалённой.**

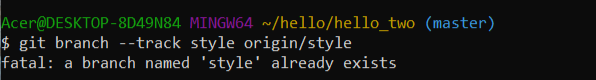
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (master)**

**$ git branch --track style origin/style**

**# Я переключаюсь на ветку.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (master)**

**$ git checkout style**

****

****

#### **5. Работа с удалёнными репозиториями**

##### **Добавление удалённого репозитория (shared)**

**# Я связываю локальный репозиторий с shared.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (style)**

**$ git remote add shared ../hello.git**

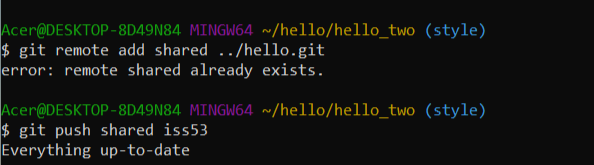
**error: remote shared already exists.**

**# Я отправляю изменения в shared.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (style)**

**$ git push shared iss53**

**Everything up-to-date**

****

#### **6. Обновление локального репозитория**

##### **Получение данных из удалённого репозитория**

**# Я забираю изменения из shared.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (style)**

**$ git fetch shared**

**# Я сливаю ветку shared/iss53 с текущей.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (style)**

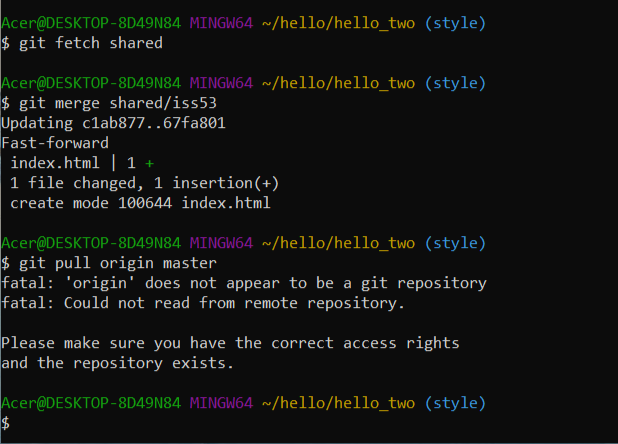
**$ git merge shared/iss53**

##### **Синхронизация с удалённым репозиторием**

**# Я обновляю локальный репозиторий.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (style)**

**$ git pull origin master**

****

**Git 1,9**

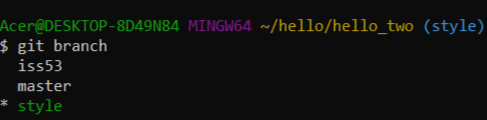
**1 Просмотр списка веток**

**Проверяю, какие ветки существуют в моём репозитории.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (style)**

**$ git branch**

**\* Показываю ветки, которые были слиты в master или ещё не слиты.**

****

**2 Фильтрация веток по истории коммитов**

* **Показать ветки, слитые в master:**

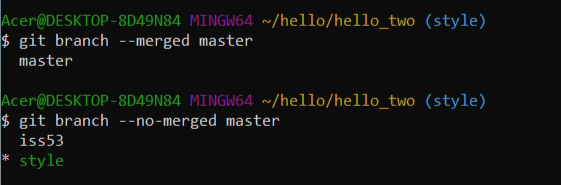
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (style)**

**$ git branch --merged master**

* **Показать ветки, не слитые в master:**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (style)**

**$ git branch --no-merged mster**

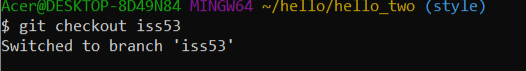
****

**3 Переключение между ветками**

**Перехожу на нужную ветку для работы.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (style)**

**$ git checkout iss53**

****

**4 Перебазирование (rebase)**

**Обновляю ветку, перенося её изменения поверх другой ветки (например, master).**

**Перебазировать style на master:**

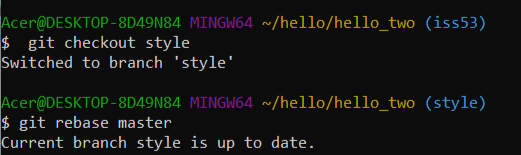
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (iss53)**

**$ git checkout style**

**Switched to branch 'style'**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (style)**

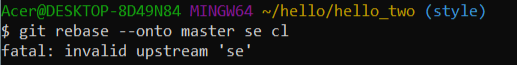
**$ git rebase master**

****

**Перебазирование с параметром --onto (для конкретных коммитов):**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (style)**

**$ git rebase --onto master se cl**

****

**Описание:**

* **se — стартовая точка.**
* **cl — конечная точка.**
* **master — целевая ветка.**

**5 Слияние веток (merge)**

**Объединяю изменения из одной ветки в другую.**

**Слить cl в master:**

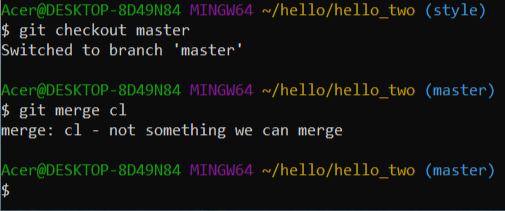
**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (style)**

**$ git checkout master**

**Switched to branch 'master'**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (master)**

**$ git merge cl**

****

**6 Удаление веток**

**Удаляю ветку после успешного слияния.**

**Acer@DESKTOP-8D49N84 MINGW64 ~/hello/hello\_two (master)**

**$ git branch -d iss53**

****