

Цель работы

Вспомнить основные команды Git Bash для работы с системой контроля версий git. Выполнить тренировочные задания по работе с проектом(репозиторием) с помощью git.

План работы

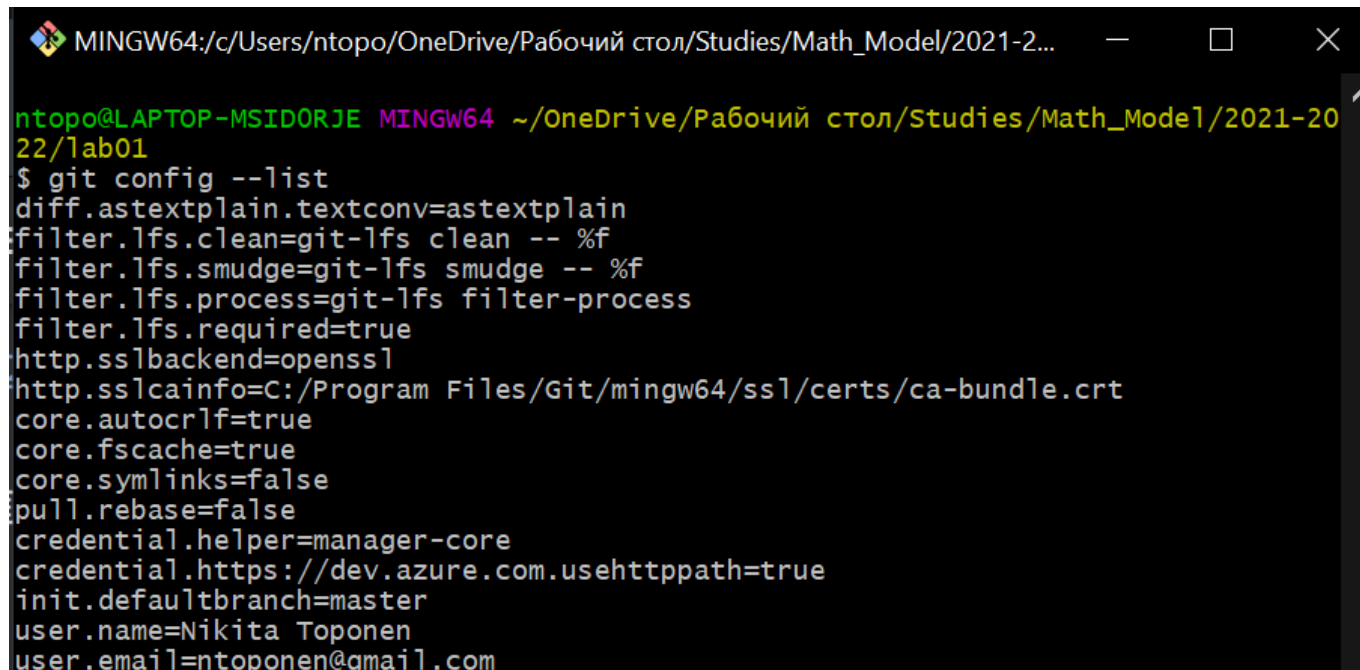
- Подготовка
- Создание проекта
- Внесение изменений
- Индексация изменений
- Отмена локальных изменений (до индексации)
- Отмена проиндексированных изменений (перед коммитом)
- Отмена коммитов
- Удаление коммитов из ветки
- Удаление тега oops
- Внесение изменений в коммиты
- Перемещение файлов
- Второй способ перемещения файлов
- Подробнее о структуре
- Git внутри: Каталог .git
- Работа непосредственно с объектами git
- Создание ветки
- Навигация по веткам
- Изменения в ветке master
- Коммит изменений README.md в ветку master

- Слияние
- Создание конфликта
- Разрешение конфликтов
- Сброс ветки `style`
- Сброс ветки `master`
- Перебазирование
- Слияние в ветку `master`
- Клонирование репозитория
- Просмотр клонированного репозитория
- Что такое `origin`?
- Удаленные ветки
- Изменение оригинального репозитория
- Слияние извлеченных изменений
- Добавление ветки наблюдения
- Чистые репозитории
- Создание чистого репозитория
- Добавление удаленного репозитория
- Отправка изменений
- Извлечение общих изменений

Выполнение лабораторной работы

Подготовка к работе с `git`

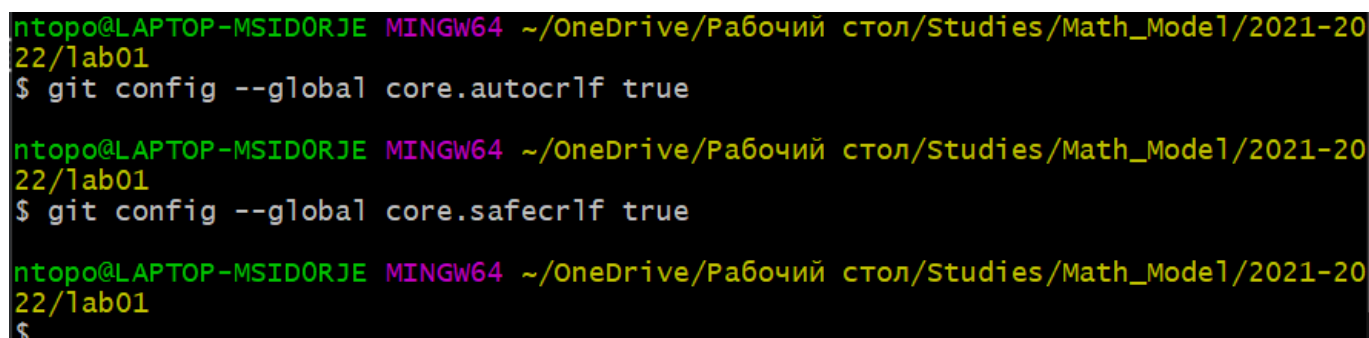
Для начала работы необходимо выполнить команды `git config`, чтобы `git` узнал ваше имя и электронную почту. Однако, так как я ранее уже использовал `git`, я сделал это раньше. Проверим это с помощью команды `git config --list`, которая выводит всю информацию о конфиге вашего `git`. В ней видно данные пользователя в последних двух строках. (рис. [-@fig:001])

A terminal window with a black background and green text. The title bar shows the path 'MINGW64:/c/Users/ntopo/OneDrive/Рабочий стол/Studies/Math_Model/2021-2...'. The prompt is 'ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01'. The command '\$ git config --list' has been executed, displaying various Git configuration settings.

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Nikita Toponen
user.email=ntoponen@gmail.com
```

{#fig:001 width=70%}

Также необходимо установить параметры окончаний строк. Настройка `core.autocrlf` с параметрами `true` и `input` делает все переводы строк текстовых файлов в главном репозитории одинаковыми. (рис. [-@fig:002])

A terminal window with a black background and green text. The prompt is 'ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01'. Two commands are shown: '\$ git config --global core.autocrlf true' and '\$ git config --global core.safecrlf true'.

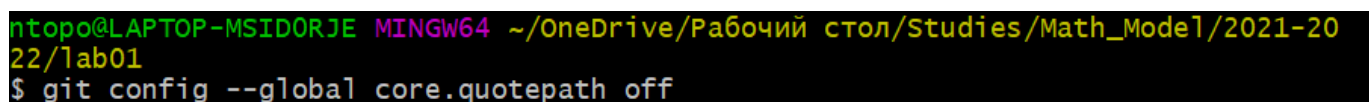
```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01
$ git config --global core.autocrlf true

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01
$ git config --global core.safecrlf true

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01
$
```

{#fig:002 width=70%}

Так как по умолчанию, git будет печатать не-ASCII символов в именах файлов в виде восьмеричных последовательностей `\nnn`. Что бы избежать нечитаемых строк, установим соответствующий флаг. (рис. [-@fig:003])

A terminal window with a black background and green text. The prompt is 'ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01'. The command '\$ git config --global core.quotepath off' is shown.

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01
$ git config --global core.quotepath off
```

{#fig:003 width=70%}

Создание проекта

Создадим пустой каталог `hello`, затем войдем него и создадим файл с именем `hello.html`, содержащий приветствие "Hello world!".

Чтобы создать git репозиторий из этого каталога, выполним команду `git init`. (рис. [-@fig:004])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello
$ git init
Initialized empty Git repository in C:/Users/ntopo/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello/.git/

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$
```

{#fig:004 width=70%}

Добавим файл в репозиторий. (рис. [-@fig:005])

```
$ git add hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit -m "Initial Commit"
[master (root-commit) 7d021d9] Initial Commit
1 file changed, 1 insertion(+)
create mode 100644 hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ |
```

{#fig:005 width=70%}

Используем команду git status, чтобы проверить текущее состояние репозитория. (рис. [-@fig:006])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git status
On branch master
nothing to commit, working tree clean
```

{#fig:006 width=70%}

Внесение изменений

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

Hello, World!

И проверим состояние рабочего каталога. (рис. [-@fig:007])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.html

no changes added to commit (use "git add" and/or "git commit -a")
```

{#fig:007 width=70%}

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

Индексация изменений

Теперь выполним команду git, чтобы проиндексировать изменения. А затем проверим состояние репозитория. (рис. [-@fig:008] рис. [-@fig:009])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.html
```

{#fig:008 width=70%}

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git status
On branch master
nothing to commit, working tree clean
```

{#fig:009 width=70%}

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

Сделаем коммит. (рис. [-@fig:010])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit
[master 0e873e7] Added h1 tag
1 file changed, 1 insertion(+), 1 deletion(-)
```

{#fig:010 width=70%}

Открылся редактор.

В первой строке введем комментарий: «Added h1 tag». Сохраним файл и выйдем из редактора.

Снова проверим состояние. (рис. [-@fig:011])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git status
On branch master
nothing to commit, working tree clean
```

{#fig:011 width=70%}

Рабочий каталог чистый, можно продолжить работу.

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

```
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Теперь добавим это изменение в индекс git.

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

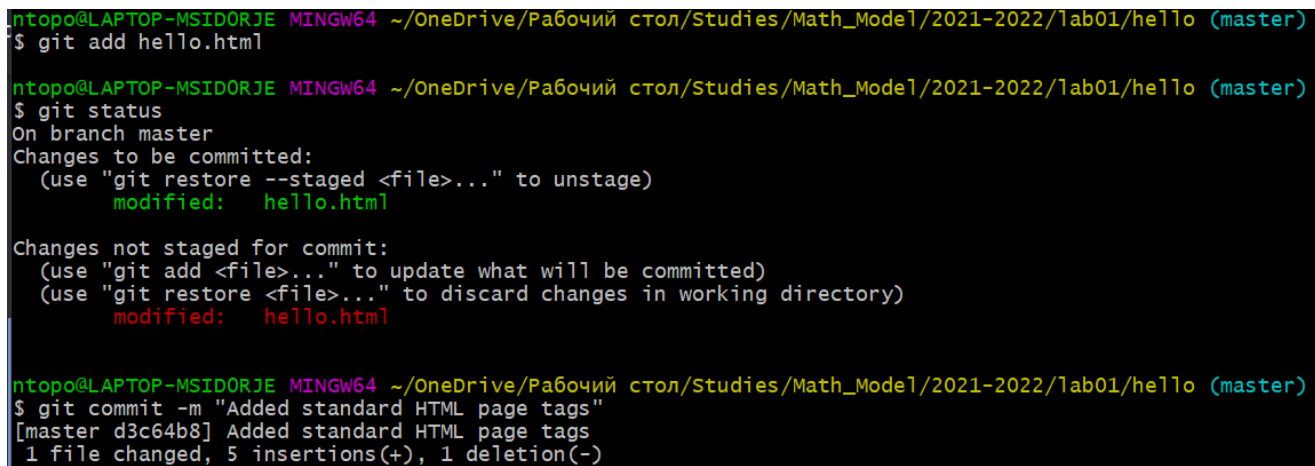
```
<html>
  <head>
```

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

Проверим текущий статус

hello.html указан дважды в состоянии. Первое изменение (добавление стандартных тегов) проиндексировано и готово к коммиту. Второе изменение (добавление заголовков HTML) является неиндексированным. Если бы мы делали коммит сейчас, заголовки не были бы сохранены в репозиторий.

Произведем коммит проиндексированного изменения (значение по умолчанию), а затем еще раз проверим состояние. (рис. [-@fig:012])



```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit -m "Added standard HTML page tags"
[master d3c64b8] Added standard HTML page tags
1 file changed, 5 insertions(+), 1 deletion(-)
```

{#fig:012 width=70%}

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

Теперь добавим второе изменение в индекс, а затем проверим состояние.

В качестве файла для добавления, мы используем текущий каталог (.). Это краткий и удобный путь для добавления всех изменений в файлы текущего каталога и его подкаталоги. Но поскольку он добавляет все, не лишним будет проверить состояние перед запуском add, просто чтобы убедиться, что мы не добавили какой-то файл, который добавлять было не нужно.

Второе изменение было проиндексировано и готово к коммиту. Выполним коммит. (рис. [-@fig:013])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.html

no changes added to commit (use "git add" and/or "git commit -a")

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add .

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit -m "Added HTML header"
[master 0a9bfad] Added HTML header
1 file changed, 2 insertions(+)

```

{#fig:013 width=70%}

Получим список произведенных изменений: (рис. [-@fig:014])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log
commit 0a9bfada91fe47814cee1a0787f1d34e508784e0 (HEAD -> master)
Author: Nikita Toponen <ntoponen@gmail.com>
Date:   Tue Feb 8 16:00:14 2022 +0300

    Added HTML header

commit d3c64b8a36319a50b15822061ef6d996a4239a01
Author: Nikita Toponen <ntoponen@gmail.com>
Date:   Tue Feb 8 15:58:42 2022 +0300

    Added standard HTML page tags

commit 0e873e7de95fbb3ba172cfe9a06053d443b07ba1
Author: Nikita Toponen <ntoponen@gmail.com>
Date:   Tue Feb 8 15:50:49 2022 +0300

    Added h1 tag

commit 7d021d976b4c2a45847dd2b48d90c58191d07eb1
Author: Nikita Toponen <ntoponen@gmail.com>
Date:   Tue Feb 8 15:45:05 2022 +0300

    Initial Commit

```

{#fig:014 width=70%}

Однострочный формат истории: (рис. [-@fig:015])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log --pretty=oneline
0a9bfada91fe47814cee1a0787f1d34e508784e0 (HEAD -> master) Added HTML header
d3c64b8a36319a50b15822061ef6d996a4239a01 Added standard HTML page tags
0e873e7de95fbb3ba172cfe9a06053d443b07ba1 Added h1 tag
7d021d976b4c2a45847dd2b48d90c58191d07eb1 Initial Commit

```

{#fig:015 width=70%}

Получим хэши предыдущих версий.

Изучил данные лога и нашел хэш для первого коммита - 7d210...

Использовал этот хэш-код для перехода к изначальному снимку репозитория с помощью команды checkout. (рис. [-@fig:016])

Затем вернулся к последней версии ветки master.

```
commit 7d021d976b4c2a45847dd2b48d90c58191d07eb1
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:45:05 2022 +0300

Initial Commit

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ AC

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git checkout 7d021d976
Note: switching to '7d021d976'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

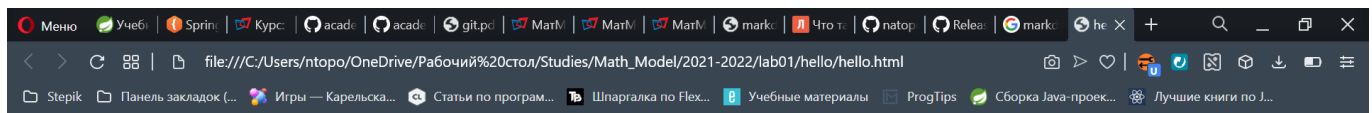
  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 7d021d9 Initial Commit

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello ((7d021d9...))
$ git checkout master
Previous HEAD position was 7d021d9 Initial Commit
Switched to branch 'master'
```

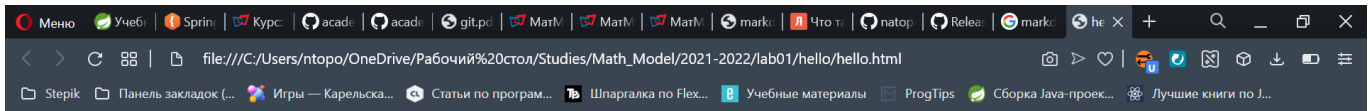
{#fig:016 width=70%}



Hello world!



{#fig:017 width=70%}



Hello world!



{#fig:018 width=70%}

Назовем текущую версию страницы hello первой (v1). Для этого создадим тег с помощью команды `git tag v1`. Теперь текущая версия страницы называется v1.

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

Это версия с тегами и , но еще пока без . Давайте сделаем ее версией v1-beta с помощью команды `git tag v1-beta`. (рис. [-@fig:019])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git tag v1

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git checkout v1^
Note: switching to 'v1^'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false
HEAD is now at d3c64b8 Added standard HTML page tags
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello ((d3c64b8...))
$ git tag v1-beta
```

{#fig:019 width=70%}

Теперь попробуйте попереключаться между двумя отмеченными версиями командой `git checkout имя соответствующего тега`. (рис. [-@fig:020])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello ((v1-beta))
$ git checkout v1
Previous HEAD position was d3c64b8 Added standard HTML page tags
HEAD is now at 0a9bfad Added HTML header

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello ((v1))
$ git checkout v1-beta
Previous HEAD position was 0a9bfad Added HTML header
HEAD is now at d3c64b8 Added standard HTML page tags
```

{#fig:020 width=70%}

Просмотрим доступные теги с помощью команды `git tag`.

Также теги можно посмотреть в логе. (рис. [-@fig:021])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello ((v1-beta))
$ git tag
v1
v1-beta

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello ((v1-beta))
$ git log master --all
commit 0a9bfada91fe47814cee1a0787f1d34e508784e0 (tag: v1, master)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:00:14 2022 +0300

    Added HTML header

commit d3c64b8a36319a50b15822061ef6d996a4239a01 (HEAD, tag: v1-beta)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:58:42 2022 +0300

    Added standard HTML page tags

commit 0e873e7de95fbb3ba172cfe9a06053d443b07ba1
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:50:49 2022 +0300

    Added h1 tag

commit 7d021d976b4c2a45847dd2b48d90c58191d07eb1
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:45:05 2022 +0300

    Initial Commit
```

{#fig:021 width=70%}

Отмена локальных изменений (до индексации)

Иногда случается, что вы изменили файл в рабочем каталоге, и хотите отменить последние коммиты. С этим справится команда `git checkout`.

Внесем изменение в файл `hello.html` в виде нежелательного комментария. (рис. [-@fig:022])



```
hello.txt - Блокнот
Файл Правка Формат Вид Справка
<html>
  <head>
  </head>
  <body>
    <h1>Hello world!</h1>
    <!-- This is a bad comment. We want to revert it. -->
  </body>
</html>
```

{#fig:022 width=70%}

Сначала проверим состояние рабочего каталога. (рис. [-@fig:023])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.html

no changes added to commit (use "git add" and/or "git commit -a")
```

{#fig:023 width=70%}

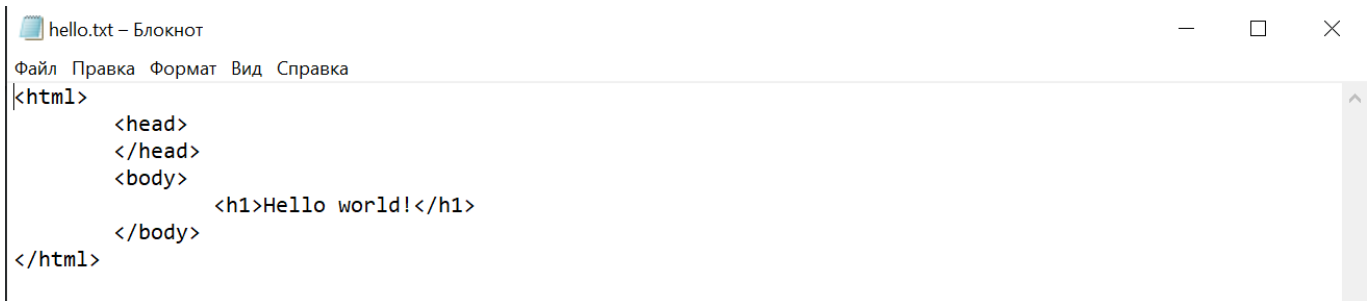
Мы видим, что файл hello.html был изменен, но еще не проиндексирован.

Используем команду git checkout для переключения версии файла hello.html в репозитории. (рис. [-@fig:024] рис. [-@fig:025])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git checkout hello.html
Updated 1 path from the index

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git status
On branch master
nothing to commit, working tree clean
```

{#fig:024 width=70%}



```
hello.txt – Блокнот
Файл Правка Формат Вид Справка
<html>
  <head>
</head>
  <body>
    <h1>Hello world!</h1>
  </body>
</html>
```

{#fig:025 width=70%}

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

Отмена проиндексированных изменений (перед коммитом)

Внесем изменение в файл hello.html в виде нежелательного комментария. (рис. [-@fig:026])



```
hello.txt – Блокнот
Файл Правка Формат Вид Справка
<html>
  <head>
    <!-- This is an unwanted but staged comment -->
  </head>
  <body>
    <h1>Hello world!</h1>
  </body>
</html>
```

{#fig:026 width=70%}

Проиндексируем это изменение. (рис. [-@fig:027])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add hello.html
```

{#fig:027 width=70%}

Проверим состояние после нежелательного изменения. (рис. [-@fig:028])

```
ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

{#fig:028 width=70%}

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

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

Команда `git reset` сбрасывает буферную зону к HEAD. Это очищает буферную зону от изменений, которые мы только что проиндексировали. (рис. [-@fig:029])

```
ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git reset HEAD hello.html
Unstaged changes after reset:
M    hello.html
```

{#fig:029 width=70%}

Команда `git reset` (по умолчанию) не изменяет рабочий каталог. Поэтому рабочий каталог все еще содержит нежелательный комментарий. Мы можем использовать команду `git checkout`, чтобы удалить нежелательные изменения в рабочем каталоге. (рис. [-@fig:030])

```
ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git checkout hello.html
Updated 1 path from the index

ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git status
On branch master
nothing to commit, working tree clean
```

{#fig:030 width=70%}

Наш рабочий каталог опять чист.

Отмена коммитов

Иногда мы понимаем, что новые коммиты являются неверными, и хотим их отменить. Есть несколько способов решения этого вопроса, здесь мы будем использовать самый безопасный.

Мы отменим коммит путем создания нового коммита, отменяющего нежелательные изменения.

Изменим файл `hello.html` на:

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

Проиндексируем его и выполним коммит. (рис. [-@fig:031])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit -m "Oops, we didn't want this commit"
[master e14eebd] Oops, we didn't want this commit
1 file changed, 1 insertion(+)
```

{#fig:031 width=70%}

Чтобы отменить коммит, нам необходимо сделать коммит, который удаляет изменения, сохраненные нежелательным коммитом. (рис. [-@fig:032])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git revert HEAD
[master 9074cb5] Revert "Oops, we didn't want this commit, so we deleted it"
1 file changed, 1 deletion(-)
```

{#fig:032 width=70%}

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

Проверка лога показывает нежелательные и отмененные коммиты в наш репозиторий. (рис. [-@fig:033])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log
commit 9074cb57908f28543b4add7ee143adb342299dc0 (HEAD -> master)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:28:56 2022 +0300

    Revert "Oops, we didn't want this commit, so we deleted it"

    This reverts commit e14eebd7bc9a8540ce2fcfb3e8585d9db56c2f48.
```

{#fig:033 width=70%}

Удаление коммитов из ветки

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

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

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

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

Давайте сделаем быструю проверку нашей истории коммитов. (рис. [-@fig:034])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log
commit 9074cb57908f28543b4add7ee143adb342299dc0 (HEAD -> master)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:28:56 2022 +0300

    Revert "Oops, we didn't want this commit, so we deleted it"

    This reverts commit e14eebd7bc9a8540ce2fcfb3e8585d9db56c2f48.

commit e14eebd7bc9a8540ce2fcfb3e8585d9db56c2f48
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:27:24 2022 +0300

    Oops, we didn't want this commit

commit 0a9bfada91fe47814cee1a0787f1d34e508784e0 (tag: v1)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:00:14 2022 +0300

    Added HTML header

commit d3c64b8a36319a50b15822061ef6d996a4239a01 (tag: v1-beta)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:58:42 2022 +0300

    Added standard HTML page tags

commit 0e873e7de95fbb3ba172cfe9a06053d443b07ba1
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:50:49 2022 +0300

    Added h1 tag

commit 7d021d976b4c2a45847dd2b48d90c58191d07eb1
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:45:05 2022 +0300

    Initial Commit

```

{#fig:034 width=70%}

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

Но прежде чем удалить коммиты, давайте отметим последний коммит тегом, чтобы потом можно было его найти. (рис. [-@fig:035])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git tag oops

```

{#fig:035 width=70%}

Глядя на историю лога, мы видим, что коммит с тегом «v1» является коммитом, предшествующим ошибочному коммиту. Давайте сбросим ветку до этой точки. Поскольку ветка имеет тег, мы можем использовать имя тега в команде сброса (если она не имеет тега, мы можем использовать хэш-значение). (рис. [-@fig:036])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git reset --hard v1
HEAD is now at 0a9bfad Added HTML header

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log
commit 0a9bfada91fe47814cee1a0787f1d34e508784e0 (HEAD -> master, tag: v1)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:00:14 2022 +0300

    Added HTML header

commit d3c64b8a36319a50b15822061ef6d996a4239a01 (tag: v1-beta)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:58:42 2022 +0300

    Added standard HTML page tags

commit 0e873e7de95fbb3ba172cfe9a06053d443b07ba1
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:50:49 2022 +0300

    Added h1 tag

commit 7d021d976b4c2a45847dd2b48d90c58191d07eb1
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:45:05 2022 +0300

    Initial Commit

```

{#fig:036 width=70%}

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

Что же случается с ошибочными коммитами? Оказывается, что коммиты все еще находятся в репозитории. На самом деле, мы все еще можем на них ссылаться. Помните, в начале этого урока мы создали для отмененного коммита тег «oops». Давайте посмотрим на все коммиты. (рис. [-@fig:037])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log --all
commit 9074cb57908f28543b4add7ee143adb342299dc0 (tag: oops)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:28:56 2022 +0300

    Revert "Oops, we didn't want this commit, so we deleted it"

    This reverts commit e14eebd7bc9a8540ce2fcfb3e8585d9db56c2f48.

commit e14eebd7bc9a8540ce2fcfb3e8585d9db56c2f48
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:27:24 2022 +0300

    Oops, we didn't want this commit

commit 0a9bfada91fe47814cee1a0787f1d34e508784e0 (HEAD -> master, tag: v1)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:00:14 2022 +0300

    Added HTML header

commit d3c64b8a36319a50b15822061ef6d996a4239a01 (tag: v1-beta)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:58:42 2022 +0300

    Added standard HTML page tags

commit 0e873e7de95fbb3ba172cfe9a06053d443b07ba1
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:50:49 2022 +0300

    Added h1 tag

commit 7d021d976b4c2a45847dd2b48d90c58191d07eb1
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:45:05 2022 +0300

    Initial Commit

```

{#fig:037 width=70%}

Мы видим, что ошибочные коммиты не исчезли. Они все еще находятся в репозитории. Просто они отсутствуют в ветке master. Если бы мы не отметили их тегами, они по-прежнему находились бы в репозитории, но не было бы никакой возможности ссылаться на них, кроме как при помощи их хэш имен. Коммиты, на которые нет ссылок, остаются в репозитории до тех пор, пока не будет запущен сборщик мусора.

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

Удаление тега оops

Тег оops свою функцию выполнил. Теперь удалим его и коммиты, на которые он ссылался, сборщиком мусора. (рис. [-@fig:038])



```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git tag -d oops
Deleted tag 'oops' (was 9074cb5)

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log --all
commit 0a9bfada91fe47814cee1a0787f1d34e508784e0 (HEAD -> master, tag: v1)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:00:14 2022 +0300

    Added HTML header

commit d3c64b8a36319a50b15822061ef6d996a4239a01 (tag: v1-beta)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:58:42 2022 +0300

    Added standard HTML page tags

commit 0e873e7de95fbb3ba172cfe9a06053d443b07ba1
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:50:49 2022 +0300

    Added h1 tag

commit 7d021d976b4c2a45847dd2b48d90c58191d07eb1
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 15:45:05 2022 +0300

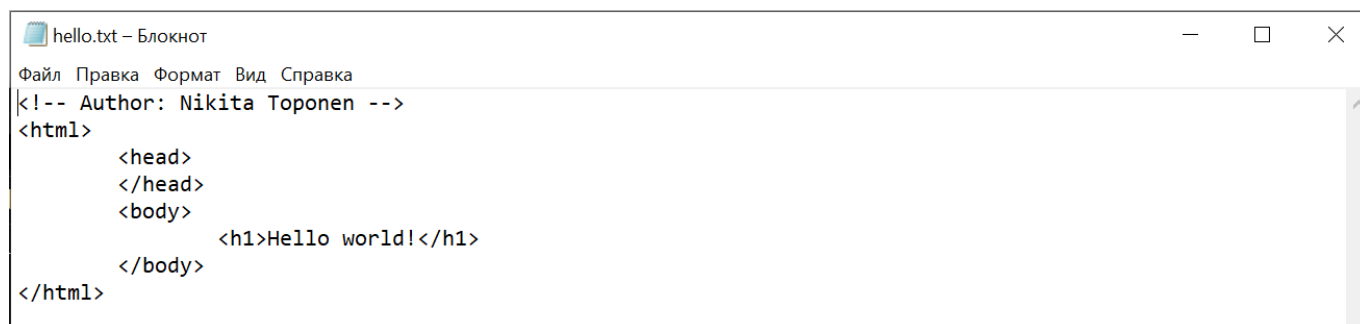
    Initial Commit
```

{#fig:038 width=70%}

Тег «oops» больше не будет отображаться в репозитории.

Внесение изменений в коммиты

Добавим в страницу комментарий автора (свою фамилию) (рис. [-@fig:039])



```
hello.txt - Блокнот
Файл Правка Формат Вид Справка
<!-- Author: Nikita Toponen -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello world!</h1>
  </body>
</html>
```

{#fig:039 width=70%}

Выполним коммит изменений. (рис. [-@fig:040])


```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit -m "Add an author comment"
[master fd695e1] Add an author comment
1 file changed, 1 insertion(+)
```

{#fig:040 width=70%}

Обновим страницу hello, включив в нее email. (рис. [-@fig:041])

```
<!-- Author: Nikita Toponen (ntoponen@gmail.com) -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello world!</h1>
  </body>
</html>
```

{#fig:041 width=70%}

Мы действительно не хотим создавать отдельный коммит только ради электронной почты. Давайте изменим предыдущий коммит, включив в него адрес электронной почты. (рис. [-@fig:042])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit --amend -m "Add an author/email comment"
[master 60e25ae] Add an author/email comment
Date: Tue Feb 8 16:42:20 2022 +0300
1 file changed, 1 insertion(+)
```

{#fig:042 width=70%}

Просмотрим историю изменений. (рис. [-@fig:043])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log
commit 60e25ae709cb81fa1d5c21ad9ba1a4667cdea0fa (HEAD -> master)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:42:20 2022 +0300

    Add an author/email comment

commit 0a9bfada91fe47814cee1a0787f1d34e508784e0 (tag: v1)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:00:14 2022 +0300

    Added HTML header
```

{#fig:043 width=70%}

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

Перемещение файлов

Сейчас мы собираемся создать структуру нашего репозитория. Давайте создадим каталог lib и перенесем файл hello.html в него. (рис. [-@fig:044])

```
ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git mv hello.html lib

ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    hello.html -> lib/hello.html
```

{#fig:044 width=70%}

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

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

Второй способ перемещения файлов

Положительной чертой `git` является то, что вы можете забыть о версионном контроле до того момента, когда вы готовы приступить к коммиту кода. Что бы случилось, если бы мы использовали командную строку операционной системы для перемещения файлов вместо команды `git`?

Следующий набор команд идентичен нашим последним действиям. Работы здесь побольше, но результат тот же. Мы могли бы выполнить:

```
mkdir lib
mv hello.html lib
git add lib/hello.html
git rm hello.html
```

Давайте сделаем коммит этого перемещения: (рис. [-@fig:045])

```
ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit -m "Moved hello.html to lib"
[master a186b61] Moved hello.html to lib
1 file changed, 0 insertions(+), 0 deletions(-)
rename hello.html => lib/hello.html (100%)
```

{#fig:045 width=70%}

Подробнее о структуре

Добавим файл `index.html` в наш репозиторий. (рис. [-@fig:046])

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

{#fig:046 width=70%}

Проиндексируем и закоммитим изменения. (рис. [-@fig:047])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add lib/index.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit -m "Added index.html"
[master 8527ec7] Added index.html
1 file changed, 5 insertions(+)
create mode 100644 lib/index.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log
commit 8527ec7a6258c4e898d827c1f369e9eb0365579b (HEAD -> master)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:53:45 2022 +0300

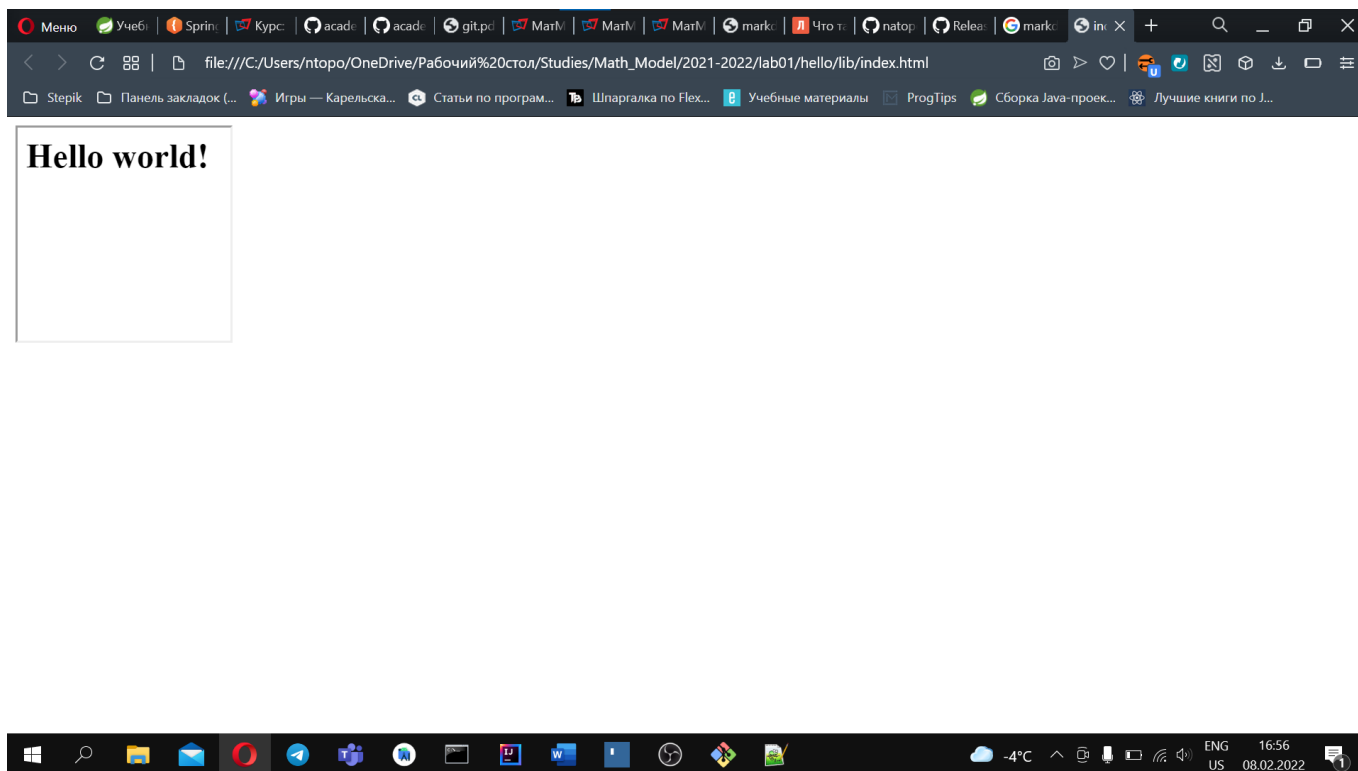
    Added index.html

commit a186b61ab59d0bae7ef10c6b8c9e53806008a862
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:50:43 2022 +0300

    Moved hello.html to lib
```

{#fig:047 width=70%}

Теперь при открытии index.html, мы видим кусок страницы hello в маленьком окошке. (рис. [-@fig:048])



{#fig:048 width=70%}

Git внутри: Каталог .git

Выполним

```
ls -C .git
```

Это каталог, в котором хранится вся информация git. (рис. [-@fig:049])

```
ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ ls -C .git
COMMIT_EDITMSG  ORIG_HEAD  description  index  logs/  packed-refs
HEAD           config     hooks/       info/  objects/ refs/
```

{#fig:049 width=70%}

Выполним

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

Мы видим набор каталогов, имена которых состоят из 2 символов. Имена каталогов являются первыми двумя буквами хэша sha1 объекта, хранящегося в git. (рис. [-@fig:050])

```
ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ ls -C .git/objects
0a/ 11/ 1d/ 41/ 57/ 67/ 7d/ 85/ 90/ a6/ ca/ d3/ e8/ ef/ fd/ pack/
0e/ 1c/ 23/ 42/ 60/ 74/ 84/ 86/ a1/ ad/ cf/ e1/ ec/ f1/ info/
```

{#fig:050 width=70%}

Выполним

```
ls -C .git/objects/0a
```

Смотрим в один из каталогов с именем из 2 букв. Мы видим файлы с именами из 38 символов. Это файлы, содержащие объекты, хранящиеся в git. Они сжаты и закодированы, поэтому просмотр их содержимого нам мало чем поможет. (рис. [-@fig:051])


```
ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ ls -C .git/objects/0a
4828708da3a970d3108ba59607562b997bc120  9bfada91fe47814cee1a0787f1d34e508784e0
```

{#fig:051 width=70%}

Выполним

```
cat .git/config
```

Это файл конфигурации, создающийся для каждого конкретного проекта. Записи в этом файле будут перезаписывать записи в файле .gitconfig вашего главного каталога, по крайней мере в рамках этого проекта. (рис. [-@fig:052])



```
MINGW64:/c/Users/ntopo/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello
core
  repositoryformatversion = 0
  filemode = false
  bare = false
  logallrefupdates = true
  symlinks = false
  ignorecase = true

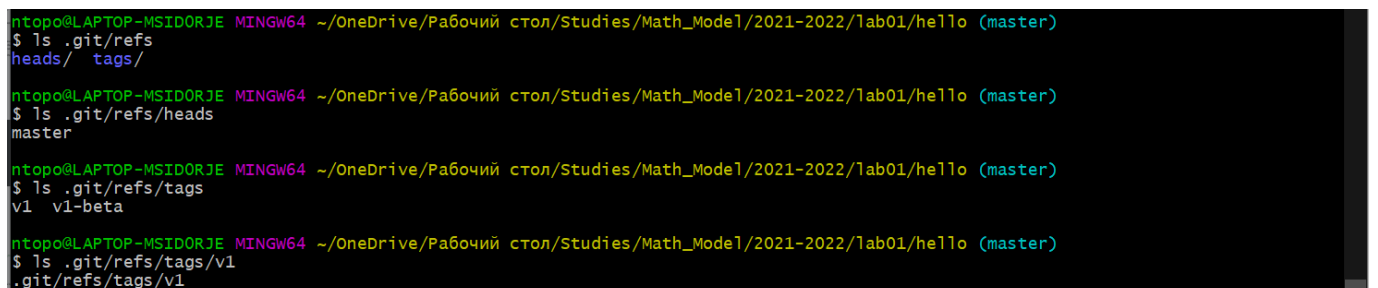
git/config [unix] (15:43 08/02/2022) 1,1 All
```

{#fig:052 width=70%}

Выполним

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

Вы должны узнавать файлы в подкаталоге тегов. Каждый файл соответствует тегу, ранее созданному с помощью команды `git tag`. Его содержание — это всего лишь хэш коммита, привязанный к тегу. Каталог `heads` практически аналогичен, но используется для веток, а не тегов. На данный момент у нас есть только одна ветка, так что все, что вы увидите в этом каталоге – это ветка `master`. (рис. [-@fig:053])



```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ ls .git/refs
heads/  tags/

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ ls .git/refs/heads
master

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ ls .git/refs/tags
v1  v1-beta

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ ls .git/refs/tags/v1
.git/refs/tags/v1
```

{#fig:053 width=70%}

Выполним

```
cat .git/HEAD
```

Файл HEAD содержит ссылку на текущую ветку, в данный момент это должна быть ветка master. (рис. [-@fig:054])

A terminal window with a black background and a Windows logo icon in the top left corner. The title bar shows the path: MINGW64:/c/Users/ntopo/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello. The command 'cat .git/HEAD' has been executed, and the output is 'ref: refs/heads/master'. The terminal shows several lines of tilde (~) characters, indicating that the output has been truncated.

.git/HEAD [unix] (16:15 08/02/2022)

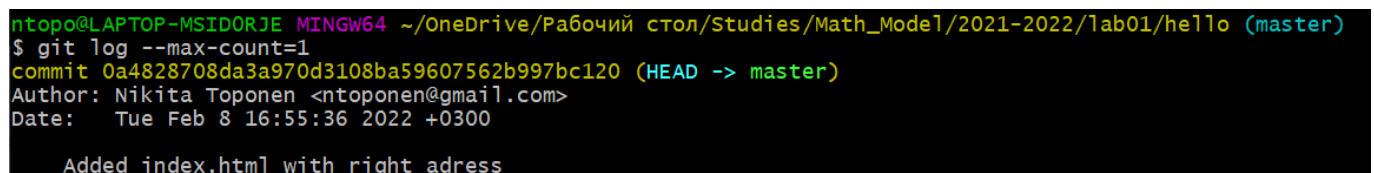
{#fig:054 width=70%}

Работа непосредственно с объектами git

Выполним

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

Эта команда показывает последний коммит в репозиторий. (рис. [-@fig:055])

A terminal window with a black background and a Windows logo icon in the top left corner. The title bar shows the path: ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master). The command '\$ git log --max-count=1' has been executed. The output shows a single commit: 'commit 0a4828708da3a970d3108ba59607562b997bc120 (HEAD -> master)' by 'Author: Nikita Toponen <ntoponen@gmail.com>' on 'Date: Tue Feb 8 16:55:36 2022 +0300'. The commit message is 'Added index.html with right adress'.

{#fig:055 width=70%}

Выведем последний коммит с помощью SHA1 хэша с помощью команд (рис. [-@fig:056]):

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

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -t 0a48
commit

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -p 0a48
tree ad820bbe9bcedcf7e4c93bdb59cad9e0315ae4b
parent 8527ec7a6258c4e898d827c1f369e9eb0365579b
author Nikita Toponen <ntoponen@gmail.com> 1644328536 +0300
committer Nikita Toponen <ntoponen@gmail.com> 1644328536 +0300

Added index.html with right adress
```

{#fig:056 width=70%}

Мы можем вывести дерево каталогов, ссылка на который идет в коммите. Это должно быть описание файлов (верхнего уровня) в нашем проекте (для конкретного коммита). Используем SHA1 хэш(ad820...) из строки «дерева», из списка выше. Для этого выполним (рис. [-@fig:057])

```
git cat-file -p ad82
```

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -p ad82
040000 tree 579f55d0d89e50d8bee7e31429b063d3b339e155    lib
```

{#fig:057 width=70%}:

Выполним то же самое, только с хэшем каталога lib(579f5...) (рис. [-@fig:058])

```
git cat-file -p 579f
```

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -p 579f
100644 blob 1ccf3b72558ad9464fe211a52ac772fd3f40d367    hello.html
100644 blob e802ef143a47e81fe724da0a816edb2b59a7e29d    index.html
```

{#fig:058 width=70%}:

Выполним то же самое, только с хэшем файла hello(1ccf3...) (рис. [-@fig:059])

```
git cat-file -p 1ccf
```

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -p 1ccf
<!-- Author: Nikita Toponen (ntoponen@gmail.com) -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello world!</h1>
  </body>
</html>
```

{#fig:059 width=70%}:

Дойдем до оригинального файла hello.html с самого первого коммита вручную по ссылкам SHA1 хэша. Для этого будем переходить по хэшу parent, пока не дойдем до начального коммита, откуда и посмотрим начальный файл. (рис. [-@fig:060] рис. [-@fig:061])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -p 8527
tree ca8805019da5a8ea5fb5cc067df4623121193c69
parent a186b61ab59d0bae7ef10c6b8c9e53806008a862
author Nikita Toponen <ntoponen@gmail.com> 1644328425 +0300
committer Nikita Toponen <ntoponen@gmail.com> 1644328425 +0300

Added index.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -p a186
tree a6e442ce647dfd6ad790d358a0ccee65b57566
parent 60e25ae709cb81fa1d5c21ad9bala4667cdea0fa
author Nikita Toponen <ntoponen@gmail.com> 1644328243 +0300
committer Nikita Toponen <ntoponen@gmail.com> 1644328243 +0300

Moved hello.html to lib

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -p 60e2
tree ece20ba7538b0e1fe35c9476db46d48ba1957927
parent 0a9bfada91fe47814cee1a0787f1d34e508784e0
author Nikita Toponen <ntoponen@gmail.com> 1644327740 +0300
committer Nikita Toponen <ntoponen@gmail.com> 1644327989 +0300

Add an author/email comment

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -p 0a9b
tree f10f99603143a4313a31417e552f4247af3a0b76
parent d3c64b8a36319a50b15822061ef6d996a4239a01
author Nikita Toponen <ntoponen@gmail.com> 1644325214 +0300
committer Nikita Toponen <ntoponen@gmail.com> 1644325214 +0300

Added HTML header

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -p d3c6
tree ef987ce51858688c5990cf2cccc438cf3efcf43d
parent 0e873e7de95fbb3ba172cfe9a06053d443b07ba1
author Nikita Toponen <ntoponen@gmail.com> 1644325122 +0300
committer Nikita Toponen <ntoponen@gmail.com> 1644325122 +0300

Added standard HTML page tags

```

{#fig:060 width=70%}

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -p 0e87
tree 86871b8aa9ea58467e72d15d5ec84218587d40ba
parent 7d021d976b4c2a45847dd2b48d90c58191d07eb1
author Nikita Toponen <ntoponen@gmail.com> 1644324649 +0300
committer Nikita Toponen <ntoponen@gmail.com> 1644324649 +0300

Added h1 tag

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -p 7d02
tree 424a53d959cb803ba950648e5d0ee6f1bd6e2965
author Nikita Toponen <ntoponen@gmail.com> 1644324305 +0300
committer Nikita Toponen <ntoponen@gmail.com> 1644324305 +0300

Initial Commit

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -p 424a
100644 blob 6769dd60bdf536a83c9353272157893043e9f7d0    hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git cat-file -p 6769
Hello world!

```

{#fig:061 width=70%}

Создание ветки

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

Назовем нашу новую ветку «style». Выполним (рис. [-@fig:070]):


```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git checkout -b style
Switched to a new branch 'style'

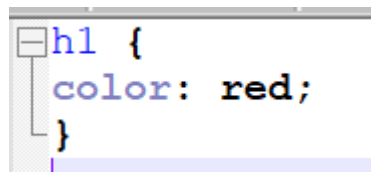
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git status
On branch style
nothing to commit, working tree clean
```

{#fig:070 width=100%}

Команда `git status` сообщает о том, что в данный момент мы находимся в ветке «style».

Добавим файл стилей `style.css`. Выполним (рис. [-@fig:071]):

```
h1 {
  color: red;
}
```



```
h1 {
  color: red;
}
```

{#fig:071 width=100%}

Проиндексируем его и выполним коммит. (рис. [-@fig:072]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git add lib/style.css

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git commit -m "Added css stylesheet"
[style 8e1954a] Added css stylesheet
1 file changed, 3 insertions(+)
create mode 100644 lib/style.css
```

{#fig:072 width=100%}

Теперь изменим основную страницу. Обновим файл `hello.html`, чтобы использовать стили `style.css`.

```
<!-- Author: Arsenij A. Ilinskiy ( ilinskiyar@gmail.com ) -->
<html>
  <head>
    <link type="text/css" rel="stylesheet"
      media="all" href="style.css" />
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Проиндексируем его и выполним коммит. (рис. [-@fig:073]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git add lib/hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git commit -m "Hello uses style.css"
[style 060903e] Hello uses style.css
1 file changed, 1 insertion(+)
```

{#fig:073 width=100%}

Теперь обновим файл index.html, чтобы он тоже использовал style.css.

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

Проиндексируем его и выполним коммит. (рис. [-@fig:074]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git add lib/index.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git commit -m "Updated index.html"
[style 24f1b79] Updated index.html
1 file changed, 3 insertions(+)
```

{#fig:074 width=100%}

Теперь при открытии index.html, мы видим кусок страницы hello в маленьком окошке с красным цветом шрифта. (рис. [-@fig:075])



{#fig:075 width=100%}

Навигация по веткам

Теперь в нашем проекте есть две ветки. Выполним (рис. [-@fig:076]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git log --all
commit 24f1b790c8e376598292d8a64bf122c506716a51 (HEAD -> style)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 17:23:08 2022 +0300

    Updated index.html

commit 060903e1230f79983724a9c80b8d0e8b23b1e355
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 17:21:21 2022 +0300

    Hello uses style.css

commit 8e1954a0853667e256edca7fd824cbf85af717a6
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 17:19:24 2022 +0300

    Added css stylesheet

commit 0a4828708da3a970d3108ba59607562b997bc120 (master)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 16:55:36 2022 +0300
```

{#fig:076 width=100%}

Используя команду git checkout переключимся на ветку master. (рис. [-@fig:077])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git checkout master
Switched to branch 'master'

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ cat lib/hello.html
<!-- Author: Nikita Toponen (ntoponen@gmail.com) -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello world!</h1>
  </body>
</html>
```

{#fig:077 width=100%}

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

Теперь вернемся к ветке style. Выполним (рис. [-@fig:078]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git checkout style
Switched to branch 'style'

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ cat lib/hello.html
<!-- Author: Nikita Toponen (ntoponen@gmail.com) -->
<html>
  <head>
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
  </head>
  <body>
    <h1>Hello world!</h1>
  </body>
</html>
```

{#fig:078 width=100%}

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

Изменения в ветке main

Допустим, пока мы меняли ветку `style`, кто-то решил обновить ветку `main`. Они добавили файл `README.md`.

Создадим файл `README` в ветке `main`. Выполним (рис. [-@fig:079]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git checkout master
Switched to branch 'master'

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ echo "This is the Hello World example from the git tutorial" > README.md
```

{#fig:079 width=100%}

Добавим в него содержимое *This is the Hello World example from the git tutorial*. (рис. [-@fig:080]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git checkout master
Switched to branch 'master'

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ echo "This is the Hello World example from the git tutorial" > README.md
```

{#fig:080 width=100%}

Сделаем коммит изменений `README.md` в ветку `main`

Проиндексируем файл `README.md` и выполним коммит. (рис. [-@fig:081]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git config --global core.autocrlf false

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add README.md

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git config --global core.autocrlf true

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit -m "Added README"
[master e4506b3] Added README
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

{#fig:081 width=100%}

Просмотрим отличающиеся ветки. Теперь у нас в репозитории есть две отличающиеся ветки.

Просмотрим ветки и их отличия, выполнив (рис. [-@fig:082]):

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log --graph --all
* commit e4506b3f2e380e740a35b22ccb7deedcf01dcd8b (HEAD -> master)
| Author: Nikita Toponen <ntoponen@gmail.com>
| Date: Tue Feb 8 17:31:01 2022 +0300
|
| Added README
|
* commit 24f1b790c8e376598292d8a64bf122c506716a51 (style)
| Author: Nikita Toponen <ntoponen@gmail.com>
| Date: Tue Feb 8 17:23:08 2022 +0300
|
| Updated index.html
|
* commit 060903e1230f79983724a9c80b8d0e8b23b1e355
| Author: Nikita Toponen <ntoponen@gmail.com>
| Date: Tue Feb 8 17:21:21 2022 +0300
|
| Hello uses style.css
|
* commit 8e1954a0853667e256edca7fd824cbf85af717a6
| Author: Nikita Toponen <ntoponen@gmail.com>
| Date: Tue Feb 8 17:19:24 2022 +0300
|
| Added css stylesheet
|
* commit 0a4828708da3a970d3108ba59607562b997bc120
| ...skipping...

```

{#fig:082 width=100%}

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

Слияние

Слияние переносит изменения из двух веток в одну. Вернемся к ветке `style` и сольем `main` с `style`, выполнив (рис. [-@fig:083], рис. [-@fig:084]):

```

git checkout style
git merge master
git log --graph --all

```

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git log --graph --all
*   commit 5b04301ca84b09c80fb0b63b3a80022d506711fb (HEAD -> style)
    |   Merge: 24f1b79 e4506b3
    |   Author: Nikita Toponen <ntoponen@gmail.com>
    |   Date:   Tue Feb 8 17:34:24 2022 +0300
    |
    |   Merge branch 'master' into style
    |
*   commit e4506b3f2e380e740a35b22ccb7deedcf01dcd8b (master)
    |   Author: Nikita Toponen <ntoponen@gmail.com>
    |   Date:   Tue Feb 8 17:31:01 2022 +0300
    |
    |   Added README
    |
*   commit 24f1b790c8e376598292d8a64bf122c506716a51
    |   Author: Nikita Toponen <ntoponen@gmail.com>
    |   Date:   Tue Feb 8 17:23:08 2022 +0300
    |
    |   Updated index.html
    |
*   commit 060903e1230f79983724a9c80b8d0e8b23b1e355
    |   Author: Nikita Toponen <ntoponen@gmail.com>
    |   Date:   Tue Feb 8 17:21:21 2022 +0300
    |
    |   Hello uses style.css
    |
*   commit 8e1954a0853667e256edca7fd824cbf85af717a6
    |   Author: Nikita Toponen <ntoponen@gmail.com>
    |   Date:   Tue Feb 8 17:19:24 2022 +0300
    |
    |   Hello uses style.css

```

{#fig:083 width=100%}

Путем периодического слияния ветки main с веткой style мы можем переносить из main любые изменения и поддерживать совместимость изменений style с изменениями в основной ветке.

Создание конфликта

Но что если изменения в ветке main конфликтуют с изменениями в style?

Вернемся в ветку master. (рис. [-@fig:085])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git checkout master
Switched to branch 'master'

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ vim lib/hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ vim lib/hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add lib/hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit -m "Life is great"
[master 31b70fb] Life is great
1 file changed, 2 insertions(+), 1 deletion(-)

```

{#fig:085 width=100%}

И внесем следующие изменения в файл hello.html. (рис. [-@fig:086])

```

<!-- Author: Arsenij A. Ilinskij ( ilinskiyar@gmail.com ) -->
<html>
  <head>
    <!-- no style -->
  </head>
  <body>
    <h1>Hello, World! Life is great!</h1>

```

```
</body>
</html>
```

Проиндексируем его и выполним коммит. (рис. [-@fig:086])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git checkout master
Switched to branch 'master'

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ vim lib/hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ vim lib/hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add lib/hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit -m "Life is great"
[master 31b70fb] Life is great
1 file changed, 2 insertions(+), 1 deletion(-)
```

{#fig:086 width=100%}

Теперь посмотрим ветки, выполнив (рис. [-@fig:087]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log --graph --all
* commit 31b70fb7f41a69ca49dad3e6efac13324caee7aa (HEAD -> master)
  Author: Nikita Toponen <ntoponen@gmail.com>
  Date: Tue Feb 8 17:37:54 2022 +0300
    Life is great

* commit 5b04301ca84b09c80fb0b63b3a80022d506711fb (style)
  Merge: 24f1b79 e4506b3
  Author: Nikita Toponen <ntoponen@gmail.com>
  Date: Tue Feb 8 17:34:24 2022 +0300
    Merge branch 'master' into style

* commit e4506b3f2e380e740a35b22ccb7deedcf01dcd8b
  Author: Nikita Toponen <ntoponen@gmail.com>
  Date: Tue Feb 8 17:31:01 2022 +0300
    Added README
```

{#fig:087 width=100%}

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

Разрешение конфликтов

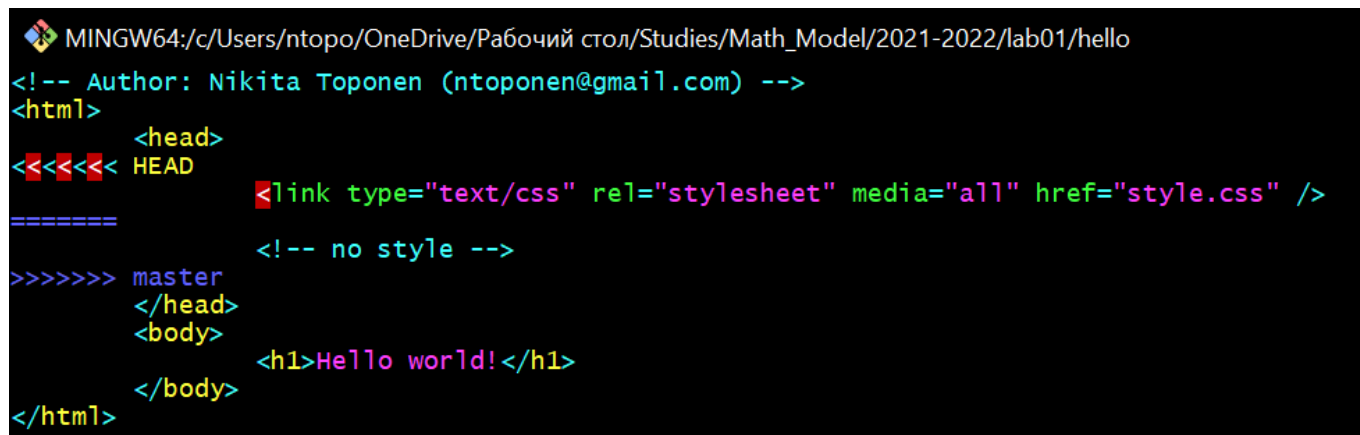
Теперь вернемся к ветке style и попытаемся объединить ее с новой веткой master, выполнив (рис. [-@fig:088]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git checkout style
Switched to branch 'style'

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git merge master
Auto-merging lib/hello.html
CONFLICT (content): Merge conflict in lib/hello.html
Automatic merge failed; fix conflicts and then commit the result.
```

{#fig:088 width=100%}

Теперь, если мы откроем файл lib/hello.html, то мы должны увидеть (рис. [-@fig:089]):



```

MINGW64:/c/Users/ntopo/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello
<!-- Author: Nikita Toponen (ntoponen@gmail.com) -->
<html>
  <head>
<<<<<< HEAD
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
=====
    <!-- no style -->
>>>>>> master
    </head>
    <body>
      <h1>Hello world!</h1>
    </body>
  </html>

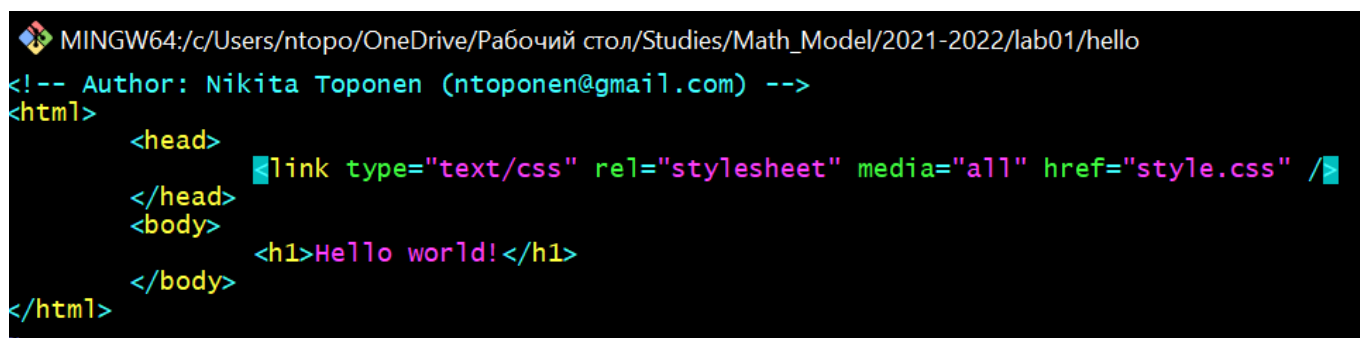
```

{#fig:089 width=100%}

Можно увидеть, что:

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

Для разрешения конфликта, нужно вручную решить проблему, а именно внести изменения в lib/hello.html. (рис. [-@fig:090])



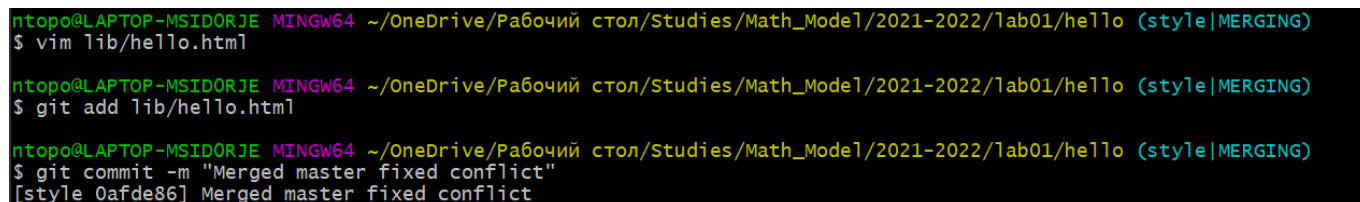
```

MINGW64:/c/Users/ntopo/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello
<!-- Author: Nikita Toponen (ntoponen@gmail.com) -->
<html>
  <head>
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
  </head>
  <body>
    <h1>Hello world!</h1>
  </body>
</html>

```

{#fig:090 width=100%}

Сделаем коммит решения конфликта. (рис. [-@fig:091])



```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style|MERGING)
$ vim lib/hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style|MERGING)
$ git add lib/hello.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style|MERGING)
$ git commit -m "Merged master fixed conflict"
[style 0afde86] Merged master fixed conflict

```

{#fig:091 width=100%}

Сброс ветки style

Вернемся на ветке style к точке перед тем, как мы слили ее с веткой main. Мы можем сбросить ветку к любому коммиту. По сути, это изменение указателя ветки на любую точку дерева коммитов. В этом случае мы хотим вернуться в ветке style в точку перед слиянием с main. Нам необходимо найти последний коммит перед слиянием, для этого выполним (рис. [-@fig:092]):


```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git checkout style
Already on 'style'

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git log --graph
*   commit 0afde869212243b032ee085a73181850dea2221a (HEAD -> style)
|  \ Merge: 5b04301 31b70fb
|   Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:43:17 2022 +0300
|
|       Merged master fixed conflict
|
| *   commit 31b70fb7f41a69ca49dad3e6efac13324caee7aa (master)
|  \ Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:37:54 2022 +0300
|
|       Life is great
|
| *   commit 5b04301ca84b09c80fb0b63b3a80022d506711fb
|  \ Merge: 24f1b79 e4506b3
|   Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:34:24 2022 +0300
|
|       Merge branch 'master' into style
|
| *   commit e4506b3f2e380e740a35b22ccb7deedcf01dcd8b
|  \ Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:31:01 2022 +0300
|
|       Added README
|
| *   commit 24f1b790c8e376598292d8a64bf122c506716a51
|  \ Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:23:08 2022 +0300
|
|       Updated index.html
|
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git reset --hard 24f1
HEAD is now at 24f1b79 Updated index.html

```

{#fig:092 width=100%}

Мы видим, что коммит «Updated index.html» (хэш(d9ac061...)) был последним на ветке style перед слиянием, поэтому сбросим ветку style к этому коммиту. (рис. [-@fig:093])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git checkout style
Already on 'style'

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git log --graph
*   commit 0afde869212243b032ee085a73181850dea2221a (HEAD -> style)
|  \ Merge: 5b04301 31b70fb
|   Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:43:17 2022 +0300
|
|   Merged master fixed conflict
|
| *   commit 31b70fb7f41a69ca49dad3e6efac13324caee7aa (master)
|  \ Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:37:54 2022 +0300
|
|   Life is great
|
| *   commit 5b04301ca84b09c80fb0b63b3a80022d506711fb
|  \ Merge: 24f1b79 e4506b3
|   Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:34:24 2022 +0300
|
|   Merge branch 'master' into style
|
| *   commit e4506b3f2e380e740a35b22ccb7deedcf01dcd8b
|  \ Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:31:01 2022 +0300
|
|   Added README
|
| *   commit 24f1b790c8e376598292d8a64bf122c506716a51
|  \ Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:23:08 2022 +0300
|
|   Updated index.html

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git reset --hard 24f1
HEAD is now at 24f1b79 Updated index.html

```

{#fig:093 width=100%}

Теперь проверим ветку. Найдем лог ветки style. (рис. [-@fig:094])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git log --graph --all
*   commit 31b70fb7f41a69ca49dad3e6efac13324caee7aa (master)
|  \ Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:37:54 2022 +0300
|
|   Life is great
|
| *   commit e4506b3f2e380e740a35b22ccb7deedcf01dcd8b
|  \ Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:31:01 2022 +0300
|
|   Added README
|
| *   commit 24f1b790c8e376598292d8a64bf122c506716a51 (HEAD -> style)
|  \ Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:23:08 2022 +0300
|
|   Updated index.html
|
| *   commit 060903e1230f79983724a9c80b8d0e8b23b1e355
|  \ Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:21:21 2022 +0300
|
|   Hello uses style.css
|
| *   commit 8e1954a0853667e256edca7fd824cbf85af717a6
|  \ ...skipping...
| *   commit 31b70fb7f41a69ca49dad3e6efac13324caee7aa (master)
|  \ Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:37:54 2022 +0300
|
|   Life is great
|
| *   commit e4506b3f2e380e740a35b22ccb7deedcf01dcd8b
|  \ Author: Nikita Toponen <ntoponen@gmail.com>
|   Date:   Tue Feb 8 17:31:01 2022 +0300

```

{#fig:094 width=100%}

Можно заметить, что у нас в истории больше нет коммитов слияний.

Сброс ветки main

Добавив интерактивный режим в ветку main, мы внесли изменения, конфликтующие с изменениями в ветке style. Давайте вернемся в ветку main в точку перед внесением конфликтующих изменений. Для этого выполним (рис. [-@fig:095]):

```
ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git reset --hard e450
HEAD is now at e4506b3 Added README

ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log --graph --all
* commit e4506b3f2e380e740a35b22ccb7deedcf01dcd8b (HEAD -> master)
| Author: Nikita Toponen <ntoponen@gmail.com>
| Date: Tue Feb 8 17:31:01 2022 +0300
|
| Added README
|
| * commit 24f1b790c8e376598292d8a64bf122c506716a51 (style)
| | Author: Nikita Toponen <ntoponen@gmail.com>
| | Date: Tue Feb 8 17:23:08 2022 +0300
| |
| | Updated index.html
| |
| | * commit 060903e1230f79983724a9c80b8d0e8b23b1e355
| | | Author: Nikita Toponen <ntoponen@gmail.com>
| | | Date: Tue Feb 8 17:21:21 2022 +0300
| | |
| | | Hello uses style.css
| |
| | * commit 8e1954a0853667e256edca7fd824cbf85af717a6
| | / Author: Nikita Toponen <ntoponen@gmail.com>
| | / Date: Tue Feb 8 17:19:24 2022 +0300
```

{#fig:095 width=100%}

Коммит «Added README» идет непосредственно перед коммитом конфликтующего интерактивного режима, поэтому мы сбросим ветку master к коммиту «Added README» (хэш(81e271b..)). (рис. [-@fig:096])

```
ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git reset --hard e450
HEAD is now at e4506b3 Added README

ntopo@LAPTOP-MSIDORJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log --graph --all
* commit e4506b3f2e380e740a35b22ccb7deedcf01dcd8b (HEAD -> master)
| Author: Nikita Toponen <ntoponen@gmail.com>
| Date: Tue Feb 8 17:31:01 2022 +0300
|
| Added README
|
| * commit 24f1b790c8e376598292d8a64bf122c506716a51 (style)
| | Author: Nikita Toponen <ntoponen@gmail.com>
| | Date: Tue Feb 8 17:23:08 2022 +0300
| |
| | Updated index.html
| |
| | * commit 060903e1230f79983724a9c80b8d0e8b23b1e355
| | | Author: Nikita Toponen <ntoponen@gmail.com>
| | | Date: Tue Feb 8 17:21:21 2022 +0300
| | |
| | | Hello uses style.css
| |
| | * commit 8e1954a0853667e256edca7fd824cbf85af717a6
| | / Author: Nikita Toponen <ntoponen@gmail.com>
| | / Date: Tue Feb 8 17:19:24 2022 +0300
```

{#fig:096 width=100%}

Посмотрев лог, можно прийти к выводу что, как будто репозиторий был перемотан назад во времени к точке до какого-либо слияния.

Перебазирование

Рассмотрим различия между слиянием и перебазированием. Для того, чтобы это сделать, нам нужно вернуться в репозиторий в момент до первого слияния (что мы и сделали в предыдущих 2ух пунктах), а затем повторить те же действия, но с использованием перебазирования вместо слияния.

Используем команду rebase вместо команды merge. Мы вернулись в точку до первого слияния и хотим перенести изменения из ветки main в нашу ветку style. На этот раз для переноса изменений из ветки main мы будем использовать команду git rebase вместо слияния. (рис. [-@fig:097])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git checkout style
Switched to branch 'style'

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git rebase master
Successfully rebased and updated refs/heads/style.

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git log --graph
* commit 9cde43282a5bcac288c405d3b2820ad826b53e5c (HEAD -> style)
  Author: Nikita Toponen <ntoponen@gmail.com>
  Date: Tue Feb 8 17:23:08 2022 +0300

    Updated index.html

* commit 09814350b2ce6cd69c33ac15c60d43b1626870ec
  Author: Nikita Toponen <ntoponen@gmail.com>
  Date: Tue Feb 8 17:21:21 2022 +0300

    Hello uses style.css

* commit b81fbe580dde385bb6c609e5f03f5a8a31029c89
  Author: Nikita Toponen <ntoponen@gmail.com>
  Date: Tue Feb 8 17:19:24 2022 +0300

    Added css stylesheet

* commit e4506b3f2e380e740a35b22ccb7deedcf01dcd8b (master)
  Author: Nikita Toponen <ntoponen@gmail.com>
  Date: Tue Feb 8 17:31:01 2022 +0300

    Added README

* commit 0a4828708da3a970d3108ba59607562b997bc120
```

{#fig:097 width=100%}

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

Слияние в ветку main

Мы поддерживали соответствие ветки style с веткой main (с помощью rebase), теперь давайте сольем изменения style в ветку main, выполнив (рис. [-@fig:098]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (style)
$ git checkout master
Switched to branch 'master'

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git merge style
Updating e4506b3..9cde432
Fast-forward
 lib/hello.html | 1 +
 lib/index.html | 3 +++
 lib/style.css   | 3 +++
 3 files changed, 7 insertions(+)
 create mode 100644 lib/style.css
```

{#fig:098 width=100%}

Поскольку последний коммит ветки main прямо предшествует последнему коммиту ветки style, git может выполнить ускоренное слияние-перемотку. При быстрой перемотке конфликтов быть не может.

Посмотрим логи, выполнив (рис. [-@fig:099]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git log
commit 9cde43282a5bcac288c405d3b2820ad826b53e5c (HEAD -> master, style)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 17:23:08 2022 +0300

    Updated index.html
```

{#fig:099 width=100%}

Теперь ветки style и main идентичны.

Клонирование репозиториев

Перейдем в рабочий каталог и сделаем клон нашего репозитория hello.

Для этого перейдем в «рабочий» каталог. (рис. [-@fig:100])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ cd ..

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01
$ pwd
/c/Users/ntopo/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01
$ ls
hello/
```

{#fig:100 width=100%}

Теперь создадим клон репозитория. (рис. [-@fig:101])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01
$ git clone hello cloned_hello
Cloning into 'cloned_hello'...
done.
```

{#fig:101 width=100%}

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

Просмотр клонированного репозитория

Давайте взглянем на клонированный репозиторий, выполнив (рис. [-@fig:103]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01
$ cd cloned_hello

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ ls
README.md  lib/
```

{#fig:103 width=100%}

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

Давайте посмотрим историю репозитория, выполнив (рис. [-@fig:104]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git log --all
commit 9cde43282a5bcac288c405d3b2820ad826b53e5c (HEAD -> master, origin/style, origin/master, origin/HEAD)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 17:23:08 2022 +0300

    Updated index.html
```

{#fig:104 width=100%}

Мы видим список всех коммитов в новой репозитории, и более / менее совпадает с историей коммитов в оригинальном репозитории.

Однако ветки main (HEAD) мы можем увидеть ветки со странными именами (origin/master, origin/style и origin/HEAD).

Что такое origin?

Выполним (рис. [-@fig:104]):

```
git remote
```

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git remote
origin

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git remote show origin
* remote origin
Fetch URL: C:/Users/ntopo/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello
Push URL: C:/Users/ntopo/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello
HEAD branch: master
Remote branches:
  master tracked
  style tracked
Local branch configured for 'git pull':
  master merges with remote master
Local ref configured for 'git push':
  master pushes to master (up to date)
```

{#fig:105 width=100%}

Мы видим, что клонированный репозиторий знает об имени по умолчанию удаленного репозитория. Давайте посмотрим, можем ли мы получить более подробную информацию об имени по умолчанию, выполнив (рис. [-@fig:105]):

```
git remote show origin
```

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git remote
origin

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git remote show origin
* remote origin
  Fetch URL: C:/Users/ntopo/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello
  Push URL: C:/Users/ntopo/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello
  HEAD branch: master
  Remote branches:
    master tracked
    style tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (up to date)
```

{#fig:106 width=100%}

Удаленные репозитории обычно размещаются на отдельной машине, возможно, централизованном сервере. Однако, как мы видим здесь, они могут с тем же успехом указывать на репозиторий на той же машине.

Удаленные ветки

Давайте посмотрим на ветки, доступные в нашем клонированном репозитории. (рис. [-@fig:107])

```
git branch
```

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git branch
* master
```

{#fig:107 width=100%}

Как мы видим, в списке только ветка master. Где ветка style? Команда git branch выводит только список локальных веток по умолчанию.

Для того, чтобы увидеть все ветки, попробуем следующую команду (рис. [-@fig:108]):

```
git branch -a
```

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/style
```

{#fig:108 width=100%}

Git выводит все коммиты в оригинальный репозиторий, но ветки в удаленном репозитории не рассматриваются как локальные.

Изменение оригинального репозитория

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

Для этого перейдем в оригинальный репозиторий hello. (рис. [-@fig:109])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ vim README.md

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add README
fatal: pathspec 'README' did not match any files

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add README.md

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ vim README

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit -m "Changed README in original repo"
[master 758644c] Changed README in original repo
1 file changed, 1 insertion(+), 1 deletion(-)
```

{#fig:109 width=100%}

Внесем следующие изменения в файла README.md.

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

Теперь добавим это изменение и сделаем коммит. (рис. [-@fig:111])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ vim README.md

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add README
fatal: pathspec 'README' did not match any files

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add README.md

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ vim README

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit -m "Changed README in original repo"
[master 758644c] Changed README in original repo
1 file changed, 1 insertion(+), 1 deletion(-)
```

{#fig:111 width=100%}

Таким образом, в оригинальном репозитории есть более поздние изменения, которых нет в клонированной версии.

Теперь научимся извлекать эти изменения из удаленного репозитория, выполнив (рис. [-@fig:112]):


```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ cd ../cloned_hello

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 294 bytes | 15.00 KiB/s, done.
From C:/Users/ntopo/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello
 9cde432..758644c master -> origin/master

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git log --all
commit 758644ce99018a6a5bf92e1d1b7671ac5123454f (origin/master, origin/HEAD)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 18:02:31 2022 +0300

    Changed README in original repo

commit 9cde43282a5bcac288c405d3b2820ad826b53e5c (HEAD -> master, origin/style)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 17:23:08 2022 +0300

    Updated index.html

```

{#fig:112 width=100%}

Сейчас мы находимся в репозитории cloned_hello. На данный момент в репозитории есть все коммиты из оригинального репозитория, но они не интегрированы в локальные ветки клонированного репозитория.

В истории выше найдем коммит «Changed README in original repo». Обратим внимание, что коммит включает в себя коммиты «origin/master» и «origin/HEAD».

Теперь давайте посмотрим на коммит «Updated index.html». Можно увидеть, что локальная ветка main указывает на этот коммит, а не на новый коммит, который мы только что извлекли.

Теперь мы можем продемонстрировать, что клонированный файл README.md не изменился, выполнив (рис. [-@fig:114]):

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ cat README.md
This is the Hello World example from the git tutorial

```

{#fig:114 width=100%}

Слияние извлеченных изменений

Сольем извлеченные изменения в локальную ветку main, выполнив (рис. [-@fig:115]):

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git merge origin/master
Updating 9cde432..758644c
Fast-forward

 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

```

{#fig:115 width=100%}

Сейчас мы должны увидеть изменения, для этого выполним (рис. [-@fig:116]):

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ cat README.md
This is the Hello World example from the git tutorial.

```

{#fig:116 width=100%}

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

Теперь давайте рассмотрим объединение `fetch` и `merge` в одну команду. (рис. [-@fig:117])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git pull
Already up to date.
```

{#fig:117 width=100%}

Это эквивалентно двум следующим шагам:

```
git fetch
git merge origin/master
```

Добавление ветки наблюдения

Ветки, которые начинаются с `remotes/origin` являются ветками оригинального репозитория. Обратим внимание, что у вас больше нет ветки под названием `style`, но система контроля версий знает, что в оригинальном репозитории ветка `style` была.

Добавим локальную ветку, которая отслеживает удаленную ветку, для этого выполним следующие команды (рис. [-@fig:118]):

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git branch --track style origin/style
Branch 'style' set up to track remote branch 'style' from 'origin'.

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git branch -a
* master
  style
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git log --max-count=2
commit 758644ce99018a6a5bf92e1d1b7671ac5123454f (HEAD -> master, origin/master, origin/HEAD)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 18:02:31 2022 +0300

    Changed README in original repo

commit 9cde43282a5bcac288c405d3b2820ad826b53e5c (origin/style, style)
Author: Nikita Toponen <ntoponen@gmail.com>
Date: Tue Feb 8 17:23:08 2022 +0300

    Updated index.html
```

{#fig:118 width=100%}

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

Чистые репозитории

Чистые репозитории (без рабочих каталогов) обычно используются для расшаривания. Обычный `git`-репозиторий подразумевает, что вы будете использовать его как рабочую директорию, поэтому вместе с файлами проекта в актуальной версии, `git` хранит все служебные, «чисто-репозиторийские» файлы в поддиректории `.git`. В удаленных репозиториях нет смысла хранить рабочие файлы на диске (как это

делается в рабочих копиях), а все что им действительно нужно — это дельты изменений и другие бинарные данные репозитория. Вот это и есть «чистый репозиторий».

Создайте чистый репозиторий

Создадим чистый репозиторий. (рис. [-@fig:119])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ cd ..

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01
$ git clone --bare hello hello.git
Cloning into bare repository 'hello.git'...
done.

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01
$ ls hello.git
HEAD  config  description  hooks/  info/  objects/  packed-refs  refs/
```

{#fig:119 width=100%}

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

Добавление удаленного репозитория

Давайте добавим репозиторий hello.git к нашему оригинальному репозиторию. (рис. [-@fig:120])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01
$ cd hello

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git remote add shared ../hello.git
```

{#fig:120 width=100%}

Отправка изменений

Так как чистые репозитории, как правило, расшариваются на каком-нибудь сетевом сервере, нам необходимо отправить наши изменения в другие репозитории. Начнем с создания изменения для отправки.

Отредактируем файл README.md.

```
This is the Hello World example from the git tutorial.
(Changed in the original and pushed to shared)
```

Теперь добавим это изменение и сделаем коммит. (рис. [-@fig:121])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git checkout master
Already on 'master'

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ vim README.md

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git add README.md

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git commit -m "Added shared comment to readme"
[master fb61d6a] Added shared comment to readme
1 file changed, 1 insertion(+)

```

{#fig:121 width=100%}

Теперь отправим эти изменения в общий репозиторий. (рис. [-@fig:122])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ git push shared master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 372 bytes | 372.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To ../hello.git
758644c..fb61d6a master -> master

```

{#fig:122 width=100%}

Общим называется репозиторий, получающий отправленные нами изменения.

Извлечение общих изменений

Теперь научимся извлекать изменения из общего репозитория.

Для этого переключимся в клонированный репозиторий. (рис. [-@fig:123])

```

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ cd ../cloned_hello

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git remote add shared ../hello.git

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git branch --track shared master
Branch 'shared' set up to track local branch 'master'.

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git pull shared master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 352 bytes | 18.00 KiB/s, done.
From ../hello
* branch          master      -> FETCH_HEAD
* [new branch]    master      -> shared/master
Updating 758644c..fb61d6a
Fast-forward
 README.md | 1 +
1 file changed, 1 insertion(+)

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ cat README.md
This is the Hello world example from the git tutorial.
(Changed in the original and pushed to shared)

```

{#fig:123 width=100%}

Далее извлечем изменения, только что отправленные в общий репозиторий. (рис. [-@fig:124])

```
ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/hello (master)
$ cd ../cloned_hello

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git remote add shared ../hello.git

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git branch --track shared master
Branch 'shared' set up to track local branch 'master'.

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ git pull shared master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 352 bytes | 18.00 KiB/s, done.
From ../hello
* branch          master      -> FETCH_HEAD
* [new branch]     master      -> shared/master
Updating 758644c..fb61d6a
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)

ntopo@LAPTOP-MSID0RJE MINGW64 ~/OneDrive/Рабочий стол/Studies/Math_Model/2021-2022/lab01/cloned_hello (master)
$ cat README.md
This is the Hello world example from the git tutorial.
(Changed in the original and pushed to shared)
```

{#fig:124 width=100%}

Как мы можем заметить, изменения были успешно извлечены.

Выводы

Научился создавать репозитории в git, использовать Git Bash, загружать файлы и папки на GitHub с помощью Git Bash.

Список литературы

- Кулябов Д. С. *Лабораторная работа №1**: git.pdf*
- Кулябов Д. С. *Лабораторная работа №1**: markdown.pdf*