

# Отчет по лабораторной работе №1

Использование git

Исаханян Эдуард Тигранович

2022 Feb 11th

# Содержание

Цель работы	5
Задание	6
Теоретическое введение	7
Выполнение лабораторной работы	8
Выводы	46
Список литературы	47

# Список таблиц

# Список иллюстраций

## Цель работы

Цель данной лабораторной работы является ознакомление с git.

## Задание

В ходе работы мы должны научится создавать проекты и изменять их. Также научится индексировать и удалять изменения, добавлять, удалять и изменять коммиты, перемещать файлы, создавать новые ветки и перемещаться по веткам, сливать, решать конфликты при слиянии и перебазировать ветки, клонировать репозитории, создавать чистые репозитории.

# Теоретическое введение

Описание основных команд git.

---

Имя команды	описание команды
git config	Позволяет настроить значение конфигурации
git init	Создает новый проект
git status	Показывает состояние репозитория
git reset	Убирает индексацию
git add	Добавляет индексацию
git commit	Фиксирует в коммите проиндексированные изменения
git log	Показывает список коммитов
git tag	Позволяет дать тег текущему состоянию на коммите
cat	Показывает содержимое файла
git revert	Отменяет новый коммит
git merge	Сливает текущую ветку с указанным
git clone	Копирует репозиторий
git branch	Создает новую ветку
git pull	Загружает данные с внешнего репозитория

---

# Выполнение лабораторной работы

## 1. Подготовка

### 1.1. Установка имени и электронной почты

Первое что нужно сделать это установить имя и электронную почту, для того чтобы git узнавал нас.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1
$ git config --global user.name "iesdai"

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1
$ git config --global user.email "i.edik2001@gmail.com"
```

### 1.2. Установка окончаний строк

Настраиваем core.autocrlf с параметрами true, чтобы сделать все переводы строк текстовых файлов в главном репозитории одинаковы. Устанавливаем core.safecrlf в warn.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1
$ git config --global core.autocrlf true

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1
$ git config --global core.safecrlf warn
```

### 1.3. Установка отображения unicode

Git по умолчанию не печатает ASCII, и чтобы избежать нечитаемых строк, установим отображение unicode.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1
$ git config --global core.quotepath off
```

## 2. Создание проекта

### \*2.1. Создание страницу “Hello,World!”\*\*

Для начала создаем пустой каталог с именем hello, затем заходим в него и создаем

```
там файл с именем hello.html.  
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1  
$ mkdir hello  
  
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1  
$ cd hello  
  
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello  
$ touch hello.thml  
  
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)  
$ git add hello.html  
  
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)  
$ git commit -m "Initial Commit"  
On branch master  
nothing to commit, working tree clean
```

## 2.2. Создание репозитория

Чтобы создать git репозиторий из этого каталога, нужно выполнить команду git init.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello  
$ git init  
Initialized empty Git repository in C:/Users/iedik/OneDrive/Документы/пары/МатМод/лаб1/hello/.git/
```

## 2.3. Добавление файла в репозиторий

Добавим файл в репозиторий.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)  
$ git add hello.html  
  
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)  
$ git commit -m "Initial Commit"  
On branch master  
nothing to commit, working tree clean
```

## 2.4. Проверка состояния репозитория

Используем команду git status, чтобы проверить текущее состояние репозитория.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)  
$ git status  
On branch master  
nothing to commit, working tree clean
```

И мы видим, что коммитить нечего. Это означает, что в репозитории хранится текущее состояние рабочего каталога, и нет никаких изменений, ожидающих записи.

## 3. Внесение изменений

### 3.1 Изменение страницы «Hello, World»

Добавим кое-какие HTML-теги к нашему приветствию.

\*hello.html - Notepad

```
File Edit Format View Help  
<h1> Hello, World!</h1>
```

#### 4. Индексация изменений

##### 4.1. Индексация

Индексируем изменения и проверим статус.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)  
$ git add hello.html  
  
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)  
$ git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    modified:   hello.html  
  
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)  
$ git commit  
[master 6bc7ba7] "Added h1 tag"  
 1 file changed, 1 insertion( ), 1 deletion(-)
```

Мы видим, что git знает об изменениях, но изменение пока не записано в репозиторий.

##### 4.2. Коммит

Сделаем коммит без -m, откроется редактор и там напишем «Added h1 tag».

```
'Added h1 tag'  
Please enter the commit message for your changes. Lines starting  
with '#' will be ignored, and an empty message aborts the commit.  
  
# On branch master  
# Changes to be committed:  
#   modified:   hello.html
```

##### 4.3. Добавить стандартные теги страницы

Изменим страницу «Hello, World», чтобы она содержала стандартные теги <html> и <body>.

```
hello.html - Notepad
File Edit Format View Help
<html>
<body>
<h1> Hello, World!</h1>
</body>
</html>
```

Теперь добавим это изменение в индекс git.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git add hello.html
```

Добавим заголовки HTML (секцию < head>) к странице «Hello, World».

```
*hello.html - Notepad
File Edit Format View Help
<html>
<head>
</head>
<body>
<h1> Hello, World!</h1>
</body>
</html>
```

Проверим текущий статус.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/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
```

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

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

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git commit -m "Added standart HTML page"
[master 772e6bc] Added standart HTML page
 1 file changed, 4 insertions(+)

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/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")
```

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

Теперь добавим второе изменение в индекс, а затем сделаем коммит.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git add .

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git commit -m "Added HTML header"
[master fe5e404] Added HTML header
 1 file changed, 2 insertions(+)
```

В качестве файла для добавления, мы использовали текущий каталог ().. Это краткий и удобный путь для добавления всех изменений в файлы текущего каталога и его подкаталоги.

#### 4.4. История

Получим список произведенных изменений.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git log
commit fe5e404a1efa6b13b4ed8435aa2242f615f34ee8 (HEAD -> master)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:51 2022 +0300

    Added HTML header

commit 772e6bc0084cb601223e8b19668cdb32548764fd
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:04 2022 +0300

    Added standart HTML page

commit 6bc7ba73d03eaeb24ed76330c8c66ac9eb806002
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:15:01 2022 +0300

    "Added h1 tag"

commit a51c8e8c7e3dfecbdb1dc47acf0e49b4a2cc9b06
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:12:45 2022 +0300

    Initial Commit
```

4.5. Получение старых версий Изучим данные лога и найдем хэш для первого коммита. Он должен быть в последней строке данных. Используем этот хэш-код, чтобы вернуть состояние к начальному.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git checkout a51c8e8c7e
Note: switching to 'a51c8e8c7e'.

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 a51c8e8 Initial Commit
```

Проверим, что мы вернулись назад.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello ((a51c8e8...))
$ cat hello.html
Hello, World!
```

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git checkout master
Previous HEAD position was a51c8e8 Initial commit
Switched to branch 'master'
```

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ cat hello.html
<html>
<head>
</head>
<body>
<h1> Hello, World!</h1>
</body>
</html>
```

Вернемся к последней версии через master и проверим это.

#### 4.6. Создание тегов версий

Назовем текущую версию страницы hello v1.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git tag v1
```

Теперь текущая версия страницы называется v1.

Создадим тег для версии, которая идет перед текущей версией и назовем его v1-beta.

В первую очередь нам надо переключиться на предыдущую версию. Вместо поиска до хэш, мы будем использовать ^, обозначающее «родитель v1».

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/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 772e6bc Added standart HTML page

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello ((772e6bc...))
$ cat hello.html
<html>
<body>
<h1> Hello, World!</h1>
</body>
</html>
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello ((772e6bc...))
$ git tag v1-beta
```

#### 4.7 Просмотр тегов

Мы можем посмотреть теги в логе.

```

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello ((v1))
$ git log master --all
commit fe5e404a1efa6b13b4ed8435aa2242f615f34ee8 (HEAD, tag: v1, master)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:51 2022 +0300

    Added HTML header

commit 772e6bc0084cb601223e8b19668cdb32548764fd (tag: v1-beta)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:04 2022 +0300

    Added standart HTML page

commit 6bc7ba73d03eaeb24ed76330c8c66ac9eb806002
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:15:01 2022 +0300

    "Added h1 tag"

commit a51c8e8c7e3dfecdb1dc47acf0e49b4a2cc9b06
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:12:45 2022 +0300

    Initial Commit

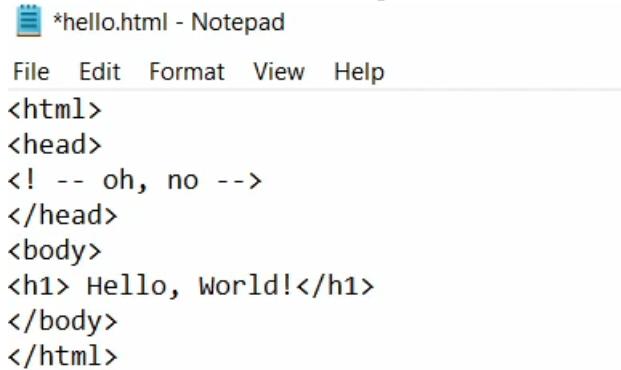
```

Мы видим теги (v1 и v1-beta) в логе вместе с именем ветки (master). Кроме того, HEAD показывает коммит, на который мы переключились.

## 5. Отмена локальных изменений

### 5.1. Изменим hello.html

Иногда случается, что мы изменили файл в рабочем каталоге, и хотим отменить последние коммиты. С этим справится команда git checkout. Добавим в hello.html нежелательный комментарий.



```

*hello.html - Notepad
File Edit Format View Help
<html>
<head>
<! -- oh, no -->
</head>
<body>
<h1> Hello, World!</h1>
</body>
</html>

```

### 5.2. Отмена изменений в рабочем каталоге

Используем команду git checkout для переключения версии файла hello.html в

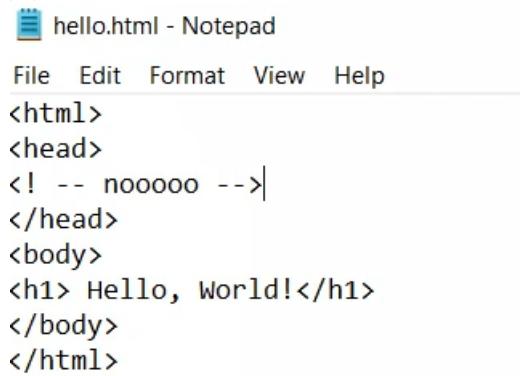
репозитории.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git checkout hello.html
Updated 1 path from the index

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ cat hello.html
<html>
<head>
</head>
<body>
<h1> Hello, World!</h1>
</body>
</html>
```

## 6. Отмена проиндексированных изменений

Добавим в hello.html нежелательный комментарий.



hello.html - Notepad

File Edit Format View Help

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

Проиндексируем это изменение и отменим индексацию.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git add hello.html

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git reset HEAD hello.html
Unstaged changes after reset:
M      hello.html
```

Команда git reset сбрасывает буферную зону к HEAD. Это очищает буферную зону от изменений, которые мы только что проиндексировали. Команда git reset (по умолчанию) не изменяет рабочий каталог. Поэтому рабочий каталог все еще содержит нежелательный комментарий. Мы можем использовать команду git checkout, чтобы удалить нежелательные изменения в рабочем каталоге.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git checkout hello.html
Updated 1 path from the index

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git status
On branch master
nothing to commit, working tree clean

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ cat hello.html
<html>    ^
<head>
</head>
<body>
<h1> Hello, World!</h1>
</body>
</html>
```

Наш рабочий каталог опять чист.

## 7. Отмена коммитов

 hello.html - Notepad

```
File Edit Format View Help
<html>
<head>
<! -- oops -->
</head>
<body>
<h1> Hello, World!</h1>
</body>
</html>
```

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git add hello.html

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git commit -m "Oops"
[master 5038c3a] Oops
 1 file changed, 1 insertion(+)
```

Чтобы отменить коммит, нам необходимо сделать коммит, который удаляет изменения, сохраненные нежелательным коммитом.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git revert HEAD
[master ca49e65] Revert "Oops"
 1 file changed, 1 deletion(-)
```

Перейдем в редактор, где мы можем отредактировать коммит-сообщение по умол-

чанию или оставить все как есть.

```
Revert "Oops"
```

```
This reverts commit 5038c3a1c3c8c8bb0bd142fcfed03eccd2589019.
```

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#       modified:   hello.html
#
```

Посмотрим, что у нас в логе.

```
commit ca49e653d61d21a3c06d748d6fe5bca8d719f008 (HEAD -> master)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:22:24 2022 +0300
```

```
Revert "Oops"
```

```
This reverts commit 5038c3a1c3c8c8bb0bd142fcfed03eccd2589019.
```

```
commit 5038c3a1c3c8c8bb0bd142fcfed03eccd2589019
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:22:10 2022 +0300
```

```
Oops
```

```
commit fe5e404a1efa6b13b4ed8435aa2242f615f34ee8 (tag: v1)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:51 2022 +0300
```

```
Added HTML header
```

```
commit 772e6bc0084cb601223e8b19668cdb32548764fd (tag: v1-beta)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:04 2022 +0300
```

```
Added standart HTML page
```

```
commit 6bc7ba73d03eaeb24ed76330c8c66ac9eb806002
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:15:01 2022 +0300
```

Проверка лога показывает нежелательные и отмененные коммиты в наш репозиторий.

## 8. Удаление коммитов из ветки

Отметим тегом это состояние и сбросим коммит к предшествующим коммиту Oops.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git tag oops

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git reset --hard v1
HEAD is now at fe5e404 Added HTML header
```

Проверим удалили ли мы этот коммит.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git log
commit fe5e404a1efa6b13b4ed8435aa2242f615f34ee8 (HEAD -> master, tag: v1)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:51 2022 +0300

    Added HTML header

commit 772e6bc0084cb601223e8b19668cdb32548764fd (tag: v1-beta)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:04 2022 +0300

    Added standart HTML page

commit 6bc7ba73d03eaeb24ed76330c8c66ac9eb806002
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:15:01 2022 +0300

    "Added h1 tag"

commit a51c8e8c7e3dfebcd1dc47acf0e49b4a2cc9b06
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:12:45 2022 +0300

    Inital Commit
```

Но если мы посмотрим на все коммиты, то увидим, что ошибочный коммит не исчез.

```
commit ca49e653d61d21a3c06d748d6fe5bca8d719f008 (tag: oops)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:22:24 2022 +0300

    Revert "Oops"

    This reverts commit 5038c3a1c3c8c8bb0bd142fcfed03eccd2589019.

commit 5038c3a1c3c8c8bb0bd142fcfed03eccd2589019
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:22:10 2022 +0300

    Oops

commit fe5e404a1efa6b13b4ed8435aa2242f615f34ee8 (HEAD -> master, tag: v1)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:51 2022 +0300

    Added HTML header

commit 772e6bc0084cb601223e8b19668cdb32548764fd (tag: v1-beta)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:04 2022 +0300

    Added standart HTML page

commit 6bc7ba73d03eaeb24ed76330c8c66ac9eb806002
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:15:01 2022 +0300
```

Давайте удалим этот коммит с помощью тега.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git tag -d oops
Deleted tag 'oops' (was ca49e65)
```

Проверим исчезли ли коммиты.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git log --all
commit fe5e404a1efa6b13b4ed8435aa2242f615f34ee8 (HEAD -> master, tag: v1)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:51 2022 +0300

    Added HTML header

commit 772e6bc0084cb601223e8b19668cdb32548764fd (tag: v1-beta)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:04 2022 +0300

    Added standart HTML page

commit 6bc7ba73d03eaeb24ed76330c8c66ac9eb806002
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:15:01 2022 +0300

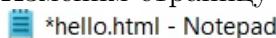
    "Added h1 tag"

commit a51c8e8c7e3dfecdb1dc47acf0e49b4a2cc9b06
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:12:45 2022 +0300

    Initial Commit
```

## 9. Внесение изменений в коммиты

Изменим страницу добавим имя и фамилию.



```
File Edit Format View Help
<! -- Author: Edik Isakhanyan -->
<html>
<head>
</head>
<body>
<h1> Hello, World!</h1>
</body>
</html>
```

Проиндексируем и сделаем коммит.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git add hello.html

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git commit -m "Added an author comment"
[master 362091d] Added an author comment
 1 file changed, 1 insertion(+)
```

Теперь добавим электронную почту.

\*hello.html - Notepad

```
File Edit Format View Help
<! -- Author: Edik Isakhanyan(i.edik2001@gamil.com)| -->
<html>
<head>
</head>
<body>
<h1> Hello, World!</h1>
</body>
</html>
```

Мы не хотим создавать отдельный коммит только ради электронной почты. Поэтому изменим предыдущий коммит, включив в него адрес электронной почты.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git add hello.html
```

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git commit --amend -m "Added an author/email comment"
[master 04e47c0] Added an author/email comment
  Date: Fri Feb 11 14:25:27 2022 +0300
  1 file changed, 1 insertion(+)
```

Посмотрим изменился ли наш коммит.

```

commit 04e47c00d6f4e52678b6fbb4c10bf26fd3cdbe5 (HEAD -> master)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:25:27 2022 +0300

    Added an author/email comment

commit fe5e404a1efa6b13b4ed8435aa2242f615f34ee8 (tag: v1)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:51 2022 +0300

    Added HTML header

commit 772e6bc0084cb601223e8b19668cdb32548764fd (tag: v1-beta)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:04 2022 +0300

    Added standart HTML page

commit 6bc7ba7103eaeb24ed76330c8c66ac9eb806002
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:15:01 2022 +0300

    "Added h1 tag"

commit a51c8e8c7e3dfecdb1dc47acf0e49b4a2cc9b06
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:12:45 2022 +0300

Initial Commit

```

## 10. Перемещение файлов

Давайте создадим новый каталог и перенесем страницу в каталог lib. Затем сделаем коммит этого перемещения.

```

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ mkdir lib

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git mv hello.html lib

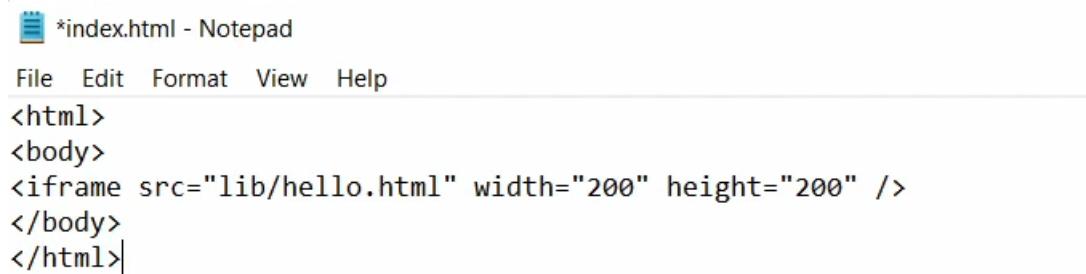
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:    hello.html -> lib/hello.html

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git commit -m "move"
[master 13402c6] move
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename hello.html => lib/hello.html (100%)

```

И добавим новый файл под именем index.html.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ touch index.html
```



```
*index.html - Notepad
File Edit Format View Help
<html>
<body>
<iframe src="lib/hello.html" width="200" height="200" />
</body>
</html>
```

## 11. Каталог .git

Посмотрим содержимое каталога .git и посмотрим базу данных объектов, мы увидим имена которых состоят из 2 символов. Это имена каталогов являются первыми двумя буквами хэша.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ ls -C .git
COMMIT_EDITMSG config description HEAD hooks/ index info/ logs/ objects/ ORIG_HEAD packed-refs refs/
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ ls -C .git/objects
04/ 1a/ 36/ 50/ 5a/ 70/ 74/ 85/ 8a/ b6/ c4/ ca/ da/ fe/ pack/
13/ 27/ 42/ 52/ 6b/ 71/ 77/ 87/ a5/ bf/ c7/ d2/ dd/ info/
```

Также посмотрим на какой-нибудь хэш в данных объектов и настройки конфигурации.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ ls -C .git/objects/fe
5e404a1efa6b13b4ed8435aa2242f615f34ee8

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
```

ПОСМОТРИМ на содержимое веток и тегов.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ ls .git/refs
heads/ tags/

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ ls .git/refs/heads
master

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ ls .git/refs/tags
v1 v1-beta

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ cat .git/refs/tags/v1
fe5e404a1efa6b13b4ed8435aa2242f615f34ee8
```

Также посмотрим на содержимое HEAD.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ cat .git/HEAD
ref: refs/heads/master
```

Мы видим, что содержимое HEAD это ссылка на текущую ветку, в данный момент это ветка master.

## 12. Работа непосредственно с объектами git

Посмотрим на наш последний коммит.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git log --max-count=1
commit da6a219195ecce83315b768d2a4734109b1ef167 (HEAD -> master)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:28:52 2022 +0300
```

index

Посмотрим на данные последнего коммита.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git cat-file -t da6a219
commit

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git cat-file -p da6a219
tree 27d9343b41bd7892698ae2aa847aff3c290764dd
parent 13402c665cda95efea08750284d83bba94ff0bb41
author iesdai <i.edik2001@gmail.com> 1644578932 +0300
committer iesdai <i.edik2001@gmail.com> 1644578932 +0300

index
```

посмотрим на данные дерево.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git cat-file -p 27d9343b41b
100644 blob 4259d9fadfd92fdfbdfebfbe1d0dbfe3ac19f1a91    index.html
040000 tree c4a2af1f2892a476d35e5f3b61e7e9514a08e846    lib
```

Затем посмотрим на данные каталога lib.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git cat-file -p c4a2af1f2
100644 blob b6765ff16cd121596e6258d76d03c87f065423c2    hello.html
```

И посмотрим на что хранится в hello.html.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git cat-file -p b6765ff16cd1
<! -- Author: Edik Isakhanyan(i.edik2001@gamil.com) -->
<html>
<head>
</head>
<body>
<h1> Hello, World!</h1>
</body>
</html>
```

### 13. Создание ветки

Давайте создадим новую ветку и назовём «style».

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git checkout -b style
Switched to a new branch 'style'
```

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (style)
$ git status
On branch style
nothing to commit, working tree clean
```

Добавим файл стилей style.css.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (style)
$ touch lib/style.css

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (style)
$ echo "h1 { color: red; }" > lib/style.css
```

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (style)
$ git add lib/style.css

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (style)
$ git commit -m "Added css"
[style 4b66de2] Added css
 1 file changed, 1 insertion(+)
 create mode 100644 lib/style.css
```

Обновим файл hello.html, чтобы использовать стили style.css.

 hello.html - Notepad

```
File Edit Format View Help
<! -- Author: Edik Isakhanyan(i.edik2001@gmail.com) -->
<html>
<head>
<link type="text/css" rel="stylesheet" media="all" href="style.css" />
</head>
<body>
<h1> Hello, World!</h1>
</body>
</html>
```

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (style)
$ git add lib/hello.html
```

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (style)
$ git commit -m "Hello uses css"
[style be135e3] Hello uses css
 1 file changed, 1 insertion(+)
```

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

 \*index.html - Notepad

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

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (style)
$ git add index.html

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (style)
$ git commit --amend -m "Updated index"
[style e5cd166] Updated index
Date: Fri Feb 11 14:34:49 2022 +0300
2 files changed, 4 insertions(+)
```

#### 14. Навигация по веткам

Посмотрим на наши ветки.

```
commit e5cd166c1fbf42244554d51be76b12550e799fbc (HEAD -> style)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:34:49 2022 +0300

    Updated index

commit 4b66de23a7f7e1dfdbda4130231e3f23bda45352
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:33:43 2022 +0300

    Added css

commit da6a219195ecce83315b768d2a4734109b1ef167 (master)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:28:52 2022 +0300

    index

commit 13402c665cda95efa08750284d83bba94ff0bb41
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:27:03 2022 +0300

    move

commit 04e47c00d6f4e52678b6fbb4c10bf26fd3cdbef5
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:25:27 2022 +0300

    Added an author/email comment
```

Вернемся на ветку master и проверим, что мы на ветке master.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (style)
$ git checkout master
Switched to branch 'master'

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ cat lib/hello.html
<! -- Author: Edik Isakhanyan(i.edik2001@gamil.com) -->
<html>
<head>
</head>
<body>
<h1> Hello, World!</h1>
</body>
</html>
```

Вернемся на ветку style и проверим это.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git checkout style
Switched to branch 'style'
```

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (style)
$ cat lib/hello.html
<! -- Author: Edik Isakhanyan(i.edik2001@gamil.com) -->
<html>
<head>
<link type="text/css" rel="stylesheet" media="all" href="style.css" />
</head>
<body>
<h1> Hello, World!</h1>
</body>
</html>
```

## 15. Изменения в ветке master

Создадим файл README в ветке master.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ touch README.md

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ echo "This is the Hello World example from the git tutorial" > README.md

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git add README.md

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git commit -m "Added README"
[master af9c287] Added README
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

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

```
* commit af9c2876c1eae2a827d7d9216b4860e2c707e89a (HEAD -> master)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:39:18 2022 +0300

    Added README

* commit 6d0d371c605310ef15aa80db4cc7a18e6eaf4ec1 (style)
| Author: iesdai <i.edik2001@gmail.com>
| Date:   Fri Feb 11 14:34:49 2022 +0300
|
|     Updated hello uses css
|
* commit 4b66de23a7f7e1dfdbda4130231e3f23bda45352
/ Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:33:43 2022 +0300

    Added css

* commit da6a219195ecce83315b768d2a4734109b1ef167
| Author: iesdai <i.edik2001@gmail.com>
| Date:   Fri Feb 11 14:28:52 2022 +0300
|
|     index
|
* commit 13402c665cda95efa08750284d83bba94ff0bb41
| Author: iesdai <i.edik2001@gmail.com>
| Date:   Fri Feb 11 14:27:03 2022 +0300
|
|     move
```

## 16. Слияние

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

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git checkout style
Switched to branch 'style'

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (style)
$ git merge
fatal: No remote for the current branch.

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (style)
$ git merge master
Merge made by the 'ort' strategy.
 README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

```

* commit f5e0cfbcffc5fe5ee99daf7316fd6a9f83347961 (HEAD -> style)
| Merge: 6d0d371 af9c287
| Author: iesdai <i.edik2001@gmail.com>
| Date:   Fri Feb 11 14:39:51 2022 +0300

|       Merge branch 'master' into style

* commit af9c2876c1eae2a827d7d9216b4860e2c707e89a (master)
| Author: iesdai <i.edik2001@gmail.com>
| Date:   Fri Feb 11 14:39:18 2022 +0300

|       Added README

* commit 6d0d371c605310ef15aa80db4cc7a18e6eaf4ec1
| Author: iesdai <i.edik2001@gmail.com>
| Date:   Fri Feb 11 14:34:49 2022 +0300

|       Updated hello uses css

* commit 4b66de23a7f7e1dfdbda4130231e3f23bda45352
| Author: iesdai <i.edik2001@gmail.com>
| Date:   Fri Feb 11 14:33:43 2022 +0300

|       Added css

* commit da6a219195ecce83315b768d2a4734109b1ef167
| Author: iesdai <i.edik2001@gmail.com>
| Date:   Fri Feb 11 14:28:52 2022 +0300

```

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

Теперь попробуем создать конфликт. Для этого в ветке master зайдем в файл hello.html и немного изменим его.

 \*hello.html - Notepad

```

File Edit Format View Help
<! -- Edik Isakhanyan (i.edik2001@gmail.com) -->
<html>
<head>
<! -- no style -->
</head>
<body>
<h1> Hello, World! Life is great! </h1>
</body>
</html>

```

```

iedik@MSI MINGW64 ~/hello (master)
$ git add lib/hello.html

iedik@MSI MINGW64 ~/hello (master)
$ git commit -m "Life is great"
[master 4890859] Life is great
 1 file changed, 2 insertions(+), 1 deletion(-)

* commit 4890859a4a4b1749f3b8d59ad1a3edd0426557a1 (HEAD -> master)
Author: edik.isa <i.edik2001@gmail.com>
Date:   Fri Feb 11 13:39:16 2022 +0300

  Life is great

* commit eb9f0778492247b36ac31079c5c02bc3e89f3459 (style)
| Merge: 47b8d07 1f21b11      ↗
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:37:36 2022 +0300

  Merge branch 'master' into style

* commit 1f21b11e86648eeceba63b45e8f6520a3a0d7b94
Author: edik.isa <i.edik2001@gmail.com>
Date:   Fri Feb 11 13:36:49 2022 +0300

  Added README

* commit 47b8d07dc5aac842801157307daf7f6cf188e774
Author: edik.isa <i.edik2001@gmail.com>
Date:   Fri Feb 11 13:33:53 2022 +0300

  Updated index.html

* commit a29850b98028267bf07f5a14b8be0220d3b6adaf
Author: edik.isa <i.edik2001@gmail.com>
Date:   Fri Feb 11 13:32:47 2022 +0300

```

После попытки слить master с style наш файл hello.html изменился, и чтобы решить этот конфликт нужно вручную изменить файл.

```

iedik@MSI MINGW64 ~/hello (master)
$ git checkout style
Switched to branch 'style'

iedik@MSI MINGW64 ~/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.

```

```

hello.html - Notepad
File Edit Format View Help
<! -- Edik Isakhanyan (i.edik2001@gmail.com) -->
<html>
<head>
<<<<< HEAD
<link type="text/css" rel="stylesheet" media="all" href="style.css" />
=====
<! -- no style -->
>>>>> master
</head>
<body>
<h1> Hello, World! Life is great! </h1>
</body>
</html>

*hello.html - Notepad
File Edit Format View Help
<! -- Edik Isakhanyan (i.edik2001@gmail.com) -->
<html>
<head>
<link type="text/css" rel="stylesheet" media="all" href="style.css" />
</head>
<body>
<h1> Hello, World! Life is great! </h1>
</body>
</html>

```

И сделаем коммит решения конфликта.

```

iedik@MSI MINGW64 ~/hello (style|MERGING)
$ git add lib/hello.html

iedik@MSI MINGW64 ~/hello (style|MERGING)
$ git commit -m "Merged master fixed conflict"
[style 24fdc1e] Merged master fixed conflict

```

## 17. Сброс веток

Перейдем на ветку style к точке перед тем, как мы слили ее с веткой master. Для этого, посмотрим на лог и скопируем хэш до слияния, после сбросим ветку style до этого момента.

```
* commit 24fdc1e3d174d1d12c007e363aa90608cc7ce8fe (HEAD -> style)
|\ Merge: eb9f077 4890859
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:42:54 2022 +0300
|
|     Merged master fixed conflict
|
* commit 4890859a4a4b1749f3b8d59ad1a3edd0426557a1 (master)
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:39:16 2022 +0300
|
|     Life is great
|
* commit eb9f0778492247b36ac31079c5c02bc3e89f3459
|\ Merge: 47b8d07 1f21b11
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:37:36 2022 +0300
|
|     Merge branch 'master' into style
|
* commit 1f21b11e86648eeceba63b45e8f6520a3a0d7b94
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:36:49 2022 +0300
|
|     Added README
|
* commit 47b8d07dc5aac842801157307daf7f6cf188e774
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:33:53 2022 +0300
```

iedik@MSI MINGW64 ~/hello (style)  
\$ git reset --hard 47b8d07dc5aac842801157307daf7f6cf188e774  
HEAD is now at 47b8d07 Updated index.html

```
* commit 4890859a4a4b1749f3b8d59ad1a3edd0426557a1 (master)
  Author: edik.isa <i.edik2001@gmail.com>
  Date:   Fri Feb 11 13:39:16 2022 +0300

    Life is great

* commit 1f21b11e86648eeceba63b45e8f6520a3a0d7b94
  Author: edik.isa <i.edik2001@gmail.com>
  Date:   Fri Feb 11 13:36:49 2022 +0300

    Added README

* commit 47b8d07dc5aac842801157307daf7f6cf188e774 (HEAD -> style)
  Author: edik.isa <i.edik2001@gmail.com>
  Date:   Fri Feb 11 13:33:53 2022 +0300

    Updated index.html

* commit a29850b98028267bf07f5a14b8be0220d3b6adaf
  Author: edik.isa <i.edik2001@gmail.com>
  Date:   Fri Feb 11 13:32:47 2022 +0300

    Hello uses style.css

* commit b30465e204377b3136b2a85cf61a1c335247e8df
  Author: edik.isa <i.edik2001@gmail.com>
  Date:   Fri Feb 11 13:30:42 2022 +0300

    Added css stylesheet
```

Теперь пойдем на ветку master в точку перед внесением конфликтующих изменений.

```
iedik@MSI MINGW64 ~/hello (master)
$ git reset --hard 1f21b11e86648eeceba63b45e8f6520a3a0d7b94
HEAD is now at 1f21b11 Added README
```

```
* commit 1f21b11e86648eeceba63b45e8f6520a3a0d7b94 (HEAD -> master)
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:36:49 2022 +0300

|     Added README

* commit 47b8d07dc5aac842801157307daf7f6cf188e774 (style)
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:33:53 2022 +0300

|     Updated index.html

* commit a29850b98028267bf07f5a14b8be0220d3b6adaf
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:32:47 2022 +0300

|     Hello uses style.css

* commit b30465e204377b3136b2a85cf61a1c335247e8df
/ Author: edik.isa <i.edik2001@gmail.com>
Date:   Fri Feb 11 13:30:42 2022 +0300

|     Added css stylesheet

* commit cb493417766e8772a08e47591959019acb008ff5
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:24:25 2022 +0300

|     Added index.html
```

## 18. Перебазирование

Мы вернулись в точку до первого слияния и хотим перенести изменения из ветки master в нашу ветку style.

```
iedik@MSI MINGW64 ~/hello (master)
$ git checkout style
Switched to branch 'style'

iedik@MSI MINGW64 ~/hello (style)
$ git rebase master
Successfully rebased and updated refs/heads/style.
```

Мы видим, что перебазирование очень похоже на слияние, однако дерево коммитов ветки style было переписано таким образом, что ветка master является частью исто-

```

commit 6c3079e692df4686f3e84bb3f01873bef5bad837 (HEAD -> master, style)
Author: edik.isa <i.edik2001@gmail.com>
Date:   Fri Feb 11 13:33:53 2022 +0300

    Updated index.html

commit 86fc07c8c11022e11f4a352444011b4d348da7af
Author: edik.isa <i.edik2001@gmail.com>
Date:   Fri Feb 11 13:32:47 2022 +0300

    Hello uses style.css

commit d8ca454fd16c5ff0a5fc7ef6871870df4340fe2d
Author: edik.isa <i.edik2001@gmail.com>
Date:   Fri Feb 11 13:30:42 2022 +0300

    Added css stylesheet

commit 1f21b11e86648eeceba63b45e8f6520a3a0d7b94
Author: edik.isa <i.edik2001@gmail.com>
Date:   Fri Feb 11 13:36:49 2022 +0300

    Added README

commit cb493417766e8772a08e47591959019acb008ff5
Author: edik.isa <i.edik2001@gmail.com>
Date:   Fri Feb 11 13:24:25 2022 +0300

```

рии коммитов.      **Added index.html**

## 19. Слияние в ветку master

После преобразования, давайте сольем изменения style в ветку master.

```

iedik@MSI MINGW64 ~/hello (style)
$ git checkout master
Switched to branch 'master'

iedik@MSI MINGW64 ~/hello (master)
$ git merge style
Updating 1f21b11..6c3079e
Fast-forward
 index.html      | 3 +++
 lib/hello.html  | 1 +
 lib/style.css   | 1 +
 3 files changed, 5 insertions(+)
 create mode 100644 lib/style.css

```

Поскольку последний коммит ветки master прямо предшествует последнему коммиту ветки style, git выполняет ускоренное слияние-перемотку.

Теперь ветки style и master идентичны.

```
* commit 6c3079e692df4686f3e84bb3f01873bef5bad837 (HEAD -> style)
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:33:53 2022 +0300
|
|     Updated index.html
|
* commit 86fc07c8c11022e11f4a352444011b4d348da7af
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:32:47 2022 +0300
|
|     Hello uses style.css
|
* commit d8ca454fd16c5ff0a5fc7ef6871870df4340fe2d
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:30:42 2022 +0300
|
|     Added css stylesheet
|
* commit 1f21b11e86648eeceba63b45e8f6520a3a0d7b94 (master)
| Author: edik.isa <i.edik2001@gmail.com>
| Date:   Fri Feb 11 13:36:49 2022 +0300
|
|     Added README
|
* commit cb493417766e8772a08e47591959019acb008ff5
| Author: edik.isa <i.edik2001@gmail.com>
```

## 20. Клонирование репозиториев

Перейдем в рабочий каталог и сделаем клон нашего репозитория hello.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ cd ../

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1
$ pwd
/c/Users/iedik/OneDrive/Документы/пары/МатМод/лаб1

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1
$ ls
hello/ lab1.mov Скринь/]

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1
$ git clone hello cloned_hello
Cloning into 'cloned_hello'...
done.

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1
$ ls
cloned_hello/ hello/ lab1.mov Скринь/]

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1
$ cd cloned_hello

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ ls
index.html lib/ README.md
```

Мы видим, что в клонированном репозитории все файлы из оригинального репозитория.

Теперь посмотрим на историю репозитория.

```
commit af9c2876c1ae2a827d7d9216b4860e2c707e89a (HEAD -> master, origin/style, origin/master, origin/HEAD)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:39:18 2022 +0300

    Added README

commit da6a219195ecce83315b768d2a4734109b1ef167
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:28:52 2022 +0300

    index

commit 13402c665cda95efa08750284d83bba94ff0bb41
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:27:03 2022 +0300

    move

commit 04e47c00d6f4e52678b6fb4c10bf26fd3cdbe5
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:25:27 2022 +0300

    Added an author/email comment

commit fe5e404a1efa6b13b4ed8435aa2242f615f34ee8 (tag: v1)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:17:51 2022 +0300

    Added HTML header
```

Мы видим список всех коммитов в новый репозиторий, и он совпадает с историей коммитов в оригинальном репозитории. Единственная разница в названиях веток.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ git remote show origin
* remote origin
  Fetch URL: C:/Users/iedik/OneDrive/Документы/пары/МатМод/лаб1/hello
  Push  URL: C:/Users/iedik/OneDrive/Документы/пары/МатМод/лаб1/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)
```

Мы видим, что клонированный репозиторий знает об имени по умолчанию удаленного репозитория.

## 21. Удаленные ветки

Давайте посмотрим на ветки, доступные в нашем клонированном репозитории.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ git branch
* master

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style
```

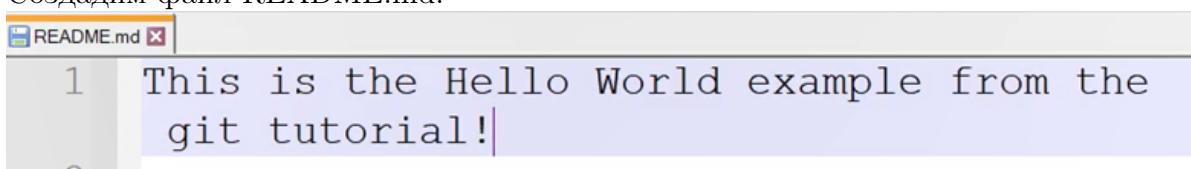
Как мы видим, в списке только ветка master. Команда git branch выводит только список локальных веток по умолчанию. Но используя команду git branch -a мы видим все ветки.

## 22. Изменение оригинального репозитория

Вернемся к оригинальному репозиторию.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ cd ../hello
```

Создадим файл README.md.



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

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git add README.md

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git commit -m "Changed README"
[master 8892088] Changed README
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Извлечем изменения из удаленного репозитория.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ cd ..../cloned_hello/

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/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), 317 bytes | 45.00 KiB/s, done.
From C:/Users/iedik/OneDrive/Документы/пары/МатМод/лаб1/hello
  af9c287..8892088  master -> origin/master
```

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ git log --all
commit 8892088468381779bd02fdb2d578843fd51c645d (origin/master, origin/HEAD)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:49:05 2022 +0300
        ↗
    Changed README

commit af9c2876c1eae2a827d7d9216b4860e2c707e89a (HEAD -> master, origin/style)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:39:18 2022 +0300
        ↗
    Added README

commit da6a219195ecce83315b768d2a4734109b1ef167
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:28:52 2022 +0300
        ↗
    index

commit 13402c665cda95efa08750284d83bba94ff0bb41
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:27:03 2022 +0300
        ↗
```

Проверим что находится в клонированном README, сольем извлеченные изменения

и проверим.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ cat README.md
This is the Hello World example from the git tutorial! ↗

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ git merge origin/master
Updating af9c287..8892088
Fast-forward
 README.md | 2 ++
 1 file changed, 1 insertion(+), 1 deletion(-)

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ cat README.md
This is the Hello World example from the git tutorial! ↗
```

Вместо двух команд git fetch и git merge origin/master мы могли бы просто сделать git pull.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ git pull
Already up to date.

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ cat README.md
This is the Hello World example from the git tutorial! ↗
```

## 23. Добавление ветки наблюдения

Ветки, которые начинаются с remotes/origin являются ветками оригинального репо-

зитория. У нас больше нет ветки под названием style, но система контроля версий знает, что в оригинальном репозитории ветка style была. Поэтому добавим локальную ветку, которая отслеживает удаленную ветку.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ git branch --track style origin/style
branch 'style' set up to track 'origin/style'.

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ git branch -a
* master
  style
    remotes/origin/HEAD -> origin/master
    remotes/origin/master
    remotes/origin/style

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ git log --max-count=2
commit 8892088468381779bd02fdb2d578843fd51c645d (HEAD -> master, origin/master, origin/HEAD)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:49:05 2022 +0300

    Changed README

commit af9c2876c1eae2a827d7d9216b4860e2c707e89a (origin/style, style)
Author: iesdai <i.edik2001@gmail.com>
Date:   Fri Feb 11 14:39:18 2022 +0300

    Added README
```

## 24. Чистые репозитории

Чистые репозитории (без рабочих каталогов) обычно используются для расширения. Обычный git-репозиторий подразумевает, что мы будем использовать его как рабочую директорию, поэтому вместе с файлами проекта в актуальной версии, git хранит все служебные, «чисто-репозиториевые» 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.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ cd ../

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1
$ git clone --bare hello hello.git
Cloning into bare repository 'hello.git'...
done.

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1
$ ls hello.git
config  description  HEAD  hooks/  info/  objects/  packed-refs  refs/
```

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.

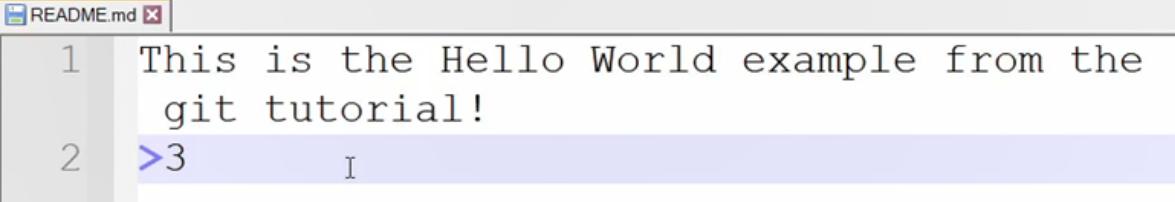
## 25. Добавление удаленного репозитория

Давайте добавим репозиторий hello.git к нашему оригинальному репозиторию.

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1
$ cd hello
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git remote add shared ..../hello.git
```

## 26. Отправка изменений

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



```
1 This is the Hello World example from the
  git tutorial!
2 >3
```

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git add README.md

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ git commit -m "Added shared"
[master b4d7dea] Added shared
 1 file changed, 1 insertion(+)

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/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), 331 bytes | 165.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
To ../hello.git
 8892088..b4d7dea  master -> master
```

## 26. Извлечение общих изменений

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

```
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/hello (master)
$ cd ../cloned_hello/
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ git remote add shared ..../hello.git/
iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ git branch --track shared master
branch 'shared' set up to track 'master'.

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/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 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 311 bytes | 51.00 KiB/s, done.
From ../hello
 * branch           master      -> FETCH_HEAD
 * [new branch]     master      -> shared/master
Updating 8892088..b4d7dea
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)

iedik@MSI MINGW64 ~/OneDrive/Документы/пары/МатМод/лаб1/cloned_hello (master)
$ cat README.md
This is the Hello World example from the git tutorial!
>3
```

## Выводы

В ходе работы, мы научились использовать git, изучили основы работы с git репозиториями.

# Список литературы