

Лабораторная работа №1

Выполнил Поляков Арсений НФИбд-03-19

Работа с git

Задание

Ознакомиться с основными возможностями и командами git.

1.1 Подготовка

Для начала мне нужно подготовить окружение. Для этого я устанавливаю имя и e-mail пользователя, от которого будет вестись работа. Также выставляем параметры для корректной работы с текстовыми файлами (в зависимости от типа ОС).

```
Математическое моделирование\lab01>git config --global user.name "arseniy.pol"
D:\Учёба\Математическое моделирование\lab01>git config --global user.email "arseniy.pol@yandex.ru"
D:\Учёба\Математическое моделирование\lab01>git config --global core.autocrlf true
D:\Учёба\Математическое моделирование\lab01>git config --global core.safecrlf true
D:\Учёба\Математическое моделирование\lab01>git config --global core.quotepath off
```

1.2 Создание проекта

Создаю проект hello, в котором создаю файл hello.html. Записываю в этот файл какой-либо текст. После этого делаю данный проект репозиторием, добавляю изменения в индекс и делаю коммит. Проверяю текущее состояние репозитория.

```
D:\Учёба\Математическое моделирование\lab01\hello>git init
Initialized empty Git repository in D:/Учёба/Математическое моделирование/lab01/hello/.git/

D:\Учёба\Математическое моделирование\lab01\hello>git add hello.html

D:\Учёба\Математическое моделирование\lab01\hello>git commit -m "Initial Commit"
[master (root-commit) 25b41de] Initial Commit
 1 file changed, 1 insertion(+)
 create mode 100644 hello.html

D:\Учёба\Математическое моделирование\lab01\hello>git status
On branch master
nothing to commit, working tree clean
```

1.3 Внесение изменений

Изменим наш текст. Проверим состояние репозитория - изменения не добавлены в

```
D:\Учёба\Математическое моделирование\lab01\hello>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")
```

индекс.

1.4 Индексация изменений

Добавляем и смотрим снова. После этого совершаем коммит, только в этот раз без параметра -m, т.е. пишем комментарий в текстовом редакторе. Смотрим состояние - нет необработанных изменений.

```
D:\Учёба\Математическое моделирование\lab01\hello>git commit
[master cf4fba5] Added h1 tag
 1 file changed, 1 insertion(+), 1 deletion(-)
D:\Учёба\Математическое моделирование\lab01\hello>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

Изменим наш текст еще раз, добавим изменения в индекс. После этого снова изменяем текст и смотрим состояние - наш файл отечем дважды - одно изменение в индексе и готово к коммиту,

```
D:\Учёба\Математическое моделирование\lab01\hello>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
```

второе - нет.

Делаем коммит и смотрим статус - есть новые изменения. Добавляем оставшиеся изменения в индекс и проверяем статус. После совершаем коммит.

```
D:\Учёба\Математическое моделирование\lab01\hello>git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'
hint: Turn this message off by running
hint: "git config advice.addEmptyPathsSpec false"

D:\Учёба\Математическое моделирование\lab01\hello>git add .
D:\Учёба\Математическое моделирование\lab01\hello>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

Просматриваем совершенные изменения. Выберем какой-либо предыдущий коммит, запомним его хэш-код и с помощью команды `checkout` и хэша скопируем его в рабочий репозиторий. Проверим содержимое нашего файла на данном коммите - оно отличается от текущего.

```
D:\Учёба\Математическое моделирование\lab01\hello>git log
commit 33aff77559ef21d2fd8e6e410bf3f05395d01ea5 (HEAD -> master)
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date:   Wed Feb 9 22:57:30 2022 +0400

    Added HTML header

commit 815cefc33e8d1f51b597bc4d105970a38ff4b55e
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date:   Wed Feb 9 22:56:37 2022 +0400

    Added standard HTML page tags

commit cf4fba5dae380b6c203b3aaf7c0a214b114b5f3e
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date:   Wed Feb 9 22:52:31 2022 +0400

    Added h1 tag

commit 25b41deca71e60928197307a6b79c48a3239d1ac
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date:   Wed Feb 9 22:51:14 2022 +0400

    Initial Commit

D:\Учёба\Математическое моделирование\lab01\hello>git checkout 33aff77
Note: switching to '33aff77'.

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 33aff77 Added HTML header
```

= Вернемся

к последней (текущей) версии при помощи `git checkout master` и проверим содержимое файла для достоверности.

```
D:\Учёба\Математическое моделирование\lab01\hello>git checkout master
Previous HEAD position was 25b41de Initial Commit
Switched to branch 'master'

D:\Учёба\Математическое моделирование\lab01\hello>type hello.html
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Создадим тег для

текущей версии страницы hello - назовем его v1. Перейдем на предыдущую версию изменений и проверим содержимое файла. Установим тег и на эту версию - v1-beta. Просматриваем все теги разными способами.

```
D:\Учёба\Математическое моделирование\lab01\hello>git tag v1
D:\Учёба\Математическое моделирование\lab01\hello>git checkout v1~1
Note: switching to 'v1~1'.

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 815cefc Added standard HTML page tags
D:\Учёба\Математическое моделирование\lab01\hello>type hello.html
<html>
    <body>
        <h1>Hello, World!</h1>
    </body>
</html>
```

1.5 Отмена локальных изменений (до индексации)

Перейдем в последнюю версию изменений, и изменим наш файл. Мы понимаем, что это изменение нам не нужно. Смотрим состояние - файл изменен.

```
D:\Учёба\Математическое моделирование\lab01\hello>git checkout master
Previous HEAD position was 815cefc Added standard HTML page tags
Switched to branch 'master'

D:\Учёба\Математическое моделирование\lab01\hello>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")
```

Для отмены

изменений воспользуемся `git checkout hello.html`, смотрим - изменений нет.

```
D:\Учёба\Математическое моделирование\lab01\hello>git checkout hello.html
Updated 1 path from the index

D:\Учёба\Математическое моделирование\lab01\hello>git status
On branch master
nothing to commit, working tree clean
```

1.6 Отмена проиндексированных изменений (перед коммитом)

Воссоздаем предыдущую ситуацию, только в этот раз добавим изменения в индекс.

```
D:\Учёба\Математическое моделирование\lab01\hello>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.html
```

Чтобы отменить нежелательные изменения в этом случае, воспользуемся командой `git reset HEAD hello.html`, а затем тем же алгоритмом, что и на предыдущем шаге.

```
D:\Учёба\Математическое моделирование\lab01\hello>git reset HEAD hello.html
Unstaged changes after reset:
M       hello.html

D:\Учёба\Математическое моделирование\lab01\hello>git checkout hello.html
Updated 1 path from the index

D:\Учёба\Математическое моделирование\lab01\hello>git status
On branch master
nothing to commit, working tree clean
```

1.7 Отмена коммитов

Теперь изменим файл и сделаем его нежелательный коммит. Для его отмены сделаем еще один коммит, отменяющий предыдущий. Проверяем логи.

```
D:\Учёба\Математическое моделирование\lab01\hello>git add hello.html

D:\Учёба\Математическое моделирование\lab01\hello>git commit -m "Oops? we didn't want this commit"
[master 27917a7] Oops? we didn't want this commit
 1 file changed, 1 insertion(+)

D:\Учёба\Математическое моделирование\lab01\hello>git revert HEAD
```

1.8 Удаление коммитов из ветки

Мы видим, что хоть мы и отменили коммит, в логах они остались - удалим их при помощи сборщика мусора. Отметим эту ветку тегом `oops`. Сбросим коммит к предыдущему тегу `v1` и проверим логи.

```
D:\Учёба\Математическое моделирование\lab01\hello>git tag oops

D:\Учёба\Математическое моделирование\lab01\hello>git reset --hard v1
HEAD is now at 33aff77 Added HTML header

D:\Учёба\Математическое моделирование\lab01\hello>git log
```

Даже после этого удаленные коммиты можно увидеть в расширенных логах.

1.9 Удаление тега oops

Удалим тег oops и проверим историю коммитов - сейчас там нет нежелательных

```
D:\Учёба\Математическое моделирование\lab01\hello>git tag -d oops
Deleted tag 'oops' (was 8d51839)

D:\Учёба\Математическое моделирование\lab01\hello>git log --all
commit 33aff77559ef21d2fd8e6e410bf3f05395d01ea5 (HEAD -> master, tag: v1)
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date:   Wed Feb 9 22:57:30 2022 +0400

    Added HTML header

commit 815cefc33e8d1f51b597bc4d105970a38ff4b55e (tag: v1-beta)
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date:   Wed Feb 9 22:56:37 2022 +0400

    Added standard HTML page tags

commit cf4fba5dae380b6c203b3aaf7c0a214b114b5f3e
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date:   Wed Feb 9 22:52:31 2022 +0400
```

КОММИТОВ.

1.10 Внесение изменений в коммиты

Добавил в страницу комментарии автора

```
<!-- Author: Polyakov Arseniy -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

И закоммитил Далее, решил добавить email И чтобы не создавать отдельный коммит, я изменил предыдущий

```
D:\Учёба\Математическое моделирование\lab01\hello>git commit -m "Add an author2 comment"
[master 98de125] Add an author2 comment
1 file changed, 1 insertion(+), 1 deletion(-)

D:\Учёба\Математическое моделирование\lab01\hello>git commit --amend -m "Add an author/email comment"
```

И в

истории можно увидеть, что коммит только один

```
\hello>git log
06d9 (HEAD -> master)
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date:   Wed Feb 9 23:35:10 2022 +0400

    Add an author/email2 comment
```

1.11 Перемещение файлов

Перемещаю файл hello.html в каталог lib

```
D:\Учеба\Математическое моделирование\lab01\hello>git mv hello.html lib

D:\Учеба\Математическое моделирование\lab01\hello>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    hello.html -> lib/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:   lib/hello.html
```

1.12 Второй способ перемещения файлов

Посмотрел второй вариант перемещения файлов, который идентичен предыдущему пункту, а также сделал коммит этого перемещения.

1.13 Подробнее о структуре

Добавил файл index.html и сделал коммит



```
index.html - Блокнот
Файл  Правка  Формат  Вид  Справка
<html>
    <body>
        <iframe src="lib/hello.html" width="200" height="200" />
    </body>
</html>
```

1.14 Git внутри: Каталог .git

Просмотрел каталог .git, базу данных объектов, состоящую из каталогов с именем из двух символов, и углубился в эти каталоги.

```

D:\Учёба\Математическое моделирование\lab01\hello>dir -C .git
Том в устройстве D имеет метку DATA
Серийный номер тома: 08C7-5A45

Содержимое папки D:\Учёба\Математическое моделирование\lab01\hello

Файл не найден

Содержимое папки D:\Учёба\Математическое моделирование\lab01\hello\.git

10.02.2022  00:36          31 COMMIT_EDITMSG
10.02.2022  00:33        213 config
09.02.2022  22:49          73 description
10.02.2022  00:35          23 HEAD
09.02.2022  22:49    <DIR>      hooks
10.02.2022  00:36        413 index
09.02.2022  22:49    <DIR>      info
09.02.2022  22:51    <DIR>      logs
10.02.2022  00:36    <DIR>      objects
10.02.2022  00:19          41 ORIG_HEAD
09.02.2022  23:31          46 packed-refs
10.02.2022  00:36    <DIR>      refs
              7 файлов          840 байт
              5 папок  898 447 065 088 байт свободно

```

Также

просмотрел файл конфигурации, ветки и теги, и файл HEAD.

```

D:\Учёба\Математическое моделирование\lab01\hello>dir -C .git\objects\03
Том в устройстве D имеет метку DATA
Серийный номер тома: 08C7-5A45

Содержимое папки D:\Учёба\Математическое моделирование\lab01\hello

Файл не найден

Содержимое папки D:\Учёба\Математическое моделирование\lab01\hello\.git\objects

Файл не найден

D:\Учёба\Математическое моделирование\lab01\hello>dir -C .git\refs
Том в устройстве D имеет метку DATA
Серийный номер тома: 08C7-5A45

Содержимое папки D:\Учёба\Математическое моделирование\lab01\hello

Файл не найден

Содержимое папки D:\Учёба\Математическое моделирование\lab01\hello\.git\refs

10.02.2022  00:36    <DIR>      .
10.02.2022  00:36    <DIR>      ..
10.02.2022  00:36    <DIR>      heads
10.02.2022  00:36    <DIR>      remotes
09.02.2022  23:31    <DIR>      tags
              0 файлов          0 байт
              5 папок  898 447 065 088 байт свободно

```

1.15 Работа непосредственно с объектами git

Вывел на экран последний коммит в репозитории командой `git log --max-count=1`, вывел последний коммит с помощью SHA1 хэша, затем вывел дерево каталогов, ссылку

на который взял в коммите. После этого вывел каталог lib, файл hello.html

```
· моделирование\lab01\hello\lib>git log --max-count=1
cc96375d48518d42bec2693b (HEAD -> master)
seniy.pol@yandex.ru>
1:57 2022 +0400

D:\Учёба\Математическое моделирование\lab01\hello\lib>git cat-file -t c82023aebf5d5643cc96375d48518d42bec2693b
commit

D:\Учёба\Математическое моделирование\lab01\hello\lib>git cat-file -p c82023aebf5d5643cc96375d48518d42bec2693b
tree 2f21d789d3a1c98b4877b68330690ecf67ad8185
parent d2946c583565a6c236a8c2a4f7b16e897bb93b37
author arseniy.pol <arseniy.pol@yandex.ru> 1644435657 +0400
committer arseniy.pol <arseniy.pol@yandex.ru> 1644435657 +0400

Added index.html

D:\Учёба\Математическое моделирование\lab01\hello\lib>git cat-file -p 2f21d789d3a1c98b4877b68330690ecf67ad8185
040000 tree d4b743582faff4a0e037949241f47cce69fa841d lib
```

1.16 Создание ветки

Чтобы сделать hello world более выразительным, создал новую ветку style, добавил файл стилей style.css, изменил основную страницу hello.html, чтобы использовать стили, обновил файл index.html, чтобы он тоже использовал style.css.

```
D:\Учёба\Математическое моделирование\lab01\hello\lib>git add style.css

D:\Учёба\Математическое моделирование\lab01\hello\lib>git commit -m "Added css stylesheet"
[style d286724] Added css stylesheet
1 file changed, 3 insertions(+)
create mode 100644 lib/style.css

D:\Учёба\Математическое моделирование\lab01\hello\lib>git add hello.html

D:\Учёба\Математическое моделирование\lab01\hello\lib>git commit -m "Hello uses style.css"
[style 00af254] Hello uses style.css
1 file changed, 2 insertions(+), 1 deletion(-)

D:\Учёба\Математическое моделирование\lab01\hello\lib>git add index.html

D:\Учёба\Математическое моделирование\lab01\hello\lib>git commit -m "Updated index.html"
[style a995921] Updated index.html
1 file changed, 3 insertions(+)
```

1.17 Навигация по веткам

Так как у нас теперь есть две ветки, для начала я переключился на ветку master, просмотрел файл hello.html без стилей, затем вернулся к ветке style и просмотрел файл

hello.html, в котором есть стили.

```
Д:\Математическое моделирование\lab01\hello\lib>git checkout master
Switched to branch 'master'

Д:\Математическое моделирование\lab01\hello\lib>type hello.html
<!-- Author: Polyakov Arseniy (arseniy.pol@yandex.ru) -->
<html>
  <head>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

Д:\Математическое моделирование\lab01\hello\lib>git checkout style
Switched to branch 'style'

Д:\Математическое моделирование\lab01\hello\lib>type hello.html
<!-- Author: Polyakov Arseniy (arseniy.pol@yandex.ru) -->
<html>
  <head>
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

1.18 Изменения в ветке master

Создал файл README.md и записал туда строку

```
Д:\Математическое моделирование\lab01\hello\lib>git add README.md

Д:\Математическое моделирование\lab01\hello\lib>git commit -m "Added README"
[master b6e1d7d] Added README
 1 file changed, 1 insertion(+)
 create mode 100644 lib/README.md
```

1.19 Коммит изменений README.md в ветку master

Сделал коммит изменений README.md, просмотрел текущие ветки и их отличия лог-командой git log —graph —all. Команда вызывает построение дерева коммитов с

помощью простых ASCII символов, а -all означает, что мы видим все ветки.

```
D:\Учёба\Математическое моделирование\lab01\hello\lib>git log --graph --all
* commit b6e1d7df72b99265e2b9d80a92318db80a70b81b (HEAD -> master)
  Author: arseniy.pol <arseniy.pol@yandex.ru>
  Date: Thu Feb 10 00:08:51 2022 +0400

    Added README

* commit a9959219f36c314d06544e5cf0cd67ac896497f9 (style)
  Author: arseniy.pol <arseniy.pol@yandex.ru>
  Date: Thu Feb 10 00:05:39 2022 +0400

    Updated index.html

* commit 00af2546ae2b25944dc092ad35223dea9f821cd9
  Author: arseniy.pol <arseniy.pol@yandex.ru>
  Date: Thu Feb 10 00:04:00 2022 +0400

    Hello uses style.css

* commit d286724de822ffbf5a5d8a159d834d6db7ce4eaa2
  Author: arseniy.pol <arseniy.pol@yandex.ru>
```

1.20 Слияние

Вернулся к ветке style и выполнил слияние master с style. Таким образом перенес изменения из двух веток в одну

```
D:\Учёба\Математическое моделирование\lab01\hello\lib>git checkout style
Switched to branch 'style'

D:\Учёба\Математическое моделирование\lab01\hello\lib>git merge master
Merge made by the 'ort' strategy.
 lib/README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 lib/README.md
```

1.21 Создание конфликта

Вернулся в ветку master и внес изменения в файл hello.html, сделал коммит. Затем просмотрел логи веток, последнее изменение в master конфликтует с некоторыми изменениями в style.

```

D:\Учёба\Математическое моделирование\lab01\hello\lib>git checkout master
Switched to branch 'master'

D:\Учёба\Математическое моделирование\lab01\hello\lib>git add hello.html

D:\Учёба\Математическое моделирование\lab01\hello\lib>git commit -m "Life is great"
[master d5bc9c8] Life is great
1 file changed, 1 insertion(+)

D:\Учёба\Математическое моделирование\lab01\hello\lib>git log --graph --all
* commit 012ae4c9f18c83bb01c7cb7813 (HEAD -> master)
  Author: arseniy.pol@yandex.ru
  Date: Thu Feb 10 00:11:25 2022 +0400
    Life is great

* commit e9f50b56f6379f9740056e959b0e85c9affe7494 (style)
  Merge: a995921 b6e1d7d
  Author: arseniy.pol <arseniy.pol@yandex.ru>
  Date: Thu Feb 10 00:09:50 2022 +0400
    Merge branch 'master' into style

* commit b6e1d7df72b99265e2b9d80a92318db80a70b81b
  Author: arseniy.pol <arseniy.pol@yandex.ru>
  Date: Thu Feb 10 00:08:51 2022 +0400
    Added README

* commit a9959219f36c314d06544e5cf0cd67ac896497f9
  Author: arseniy.pol <arseniy.pol@yandex.ru>
  Date: Thu Feb 10 00:05:39 2022 +0400
    Updated index.html

```

1.22 Разрешение конфликтов

Вернулся к ветке style и попытался объединить ее с новой веткой master, возник конфликт. Открыл файл hello.html и вручную разрешил конфликт. Затем сделал коммит решения конфликта.

```

<!-- Author: Polyakov Arseniy (arseniy.pol@yandex.ru) -->
<html>
  <head>
<<<<<< HEAD
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
=====
    <!-- no style -->
>>>>>> master
    </head>
    <body>
      <h1>Hello, World!</h1>
    </body>
  </html>

```

1.23 Сброс ветки style

Вернулся к ветке style в точку перед слиянием с master, нашел последний коммит перед слиянием и сбросил ветку style к этому коммиту. Затем проверил ветку, в логах нет

КОММИТОВ СЛИЯНИЙ.

```
D:\Учёба\Математическое моделирование\lab01\hello\lib>git reset --hard a9959219f36c314d06544
HEAD is now at a995921 Updated index.html
```

1.24 Сброс ветки master

Вернулся к ветке master перед внесением конфликтующих изменений. Нашел коммит “Added README” и сбросил ветку к этому коммиту.

```
D:\Учёба\Математическое моделирование\lab01\hello\lib>git reset --hard b6e1d7df72b99265e2b9d80a92318db80a70b81b
HEAD is now at b6e1d7d Added README
```

1.25 Перебазирование

В данном пункте я использую команду rebase, вместо команды merge. И хочу перенести изменения из ветки master в ветку style.

```
D:\Учёба\Математическое моделирование\lab01\hello\lib>git checkout style
Switched to branch 'style'

D:\Учёба\Математическое моделирование\lab01\hello\lib>git rebase master
Successfully rebased and updated refs/heads/style.
```

Конечный результат перебазирования очень похож на результат слияния. Однако, дерево коммитов значительно отличается. Дерево коммитов ветки style было переписано таким образом, что ветка master является частью истории коммитов. Это делает цепь коммитов линейной и гораздо более читабельной. Рекомендуется использовать git rebase для кратковременных, локальных веток, а слияние для веток в публичном репозитории.

1.26 Слияние в ветку master

Я поддерживал соответствие ветки style с веткой master, теперь солью изменения style

```
...: моделирование\lab01\hello\lib>git checkout style
...le'

...: моделирование\lab01\hello\lib>git rebase master
Successfully rebased and updated refs/heads/style.

D:\Учёба\Математическое моделирование\lab01\hello\lib>git checkout master
Switched to branch 'master'

D:\Учёба\Математическое моделирование\lab01\hello\lib>git merge style
Updating b6e1d7d..dba92ff
Fast-forward
 lib/hello.html | 3 ++-
 lib/index.html | 3 +++
 lib/style.css  | 3 +++
 3 files changed, 8 insertions(+), 1 deletion(-)
 create mode 100644 lib/style.css
```

в ветку master.

Теперь ветки style и master идентичны.

```
D:\Учёба\Математическое моделирование\lab01\hello\lib>git log
commit dba92ff0646234299cbf05f34b3c95d3bfec69bf (HEAD -> master, style)
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date: Thu Feb 10 00:05:39 2022 +0400

    Updated index.html
```

1.27 Клонирование репозитория

Для клонирования репозитория нужно перейти в рабочий каталог и с помощью команды clone создать clone репозитория

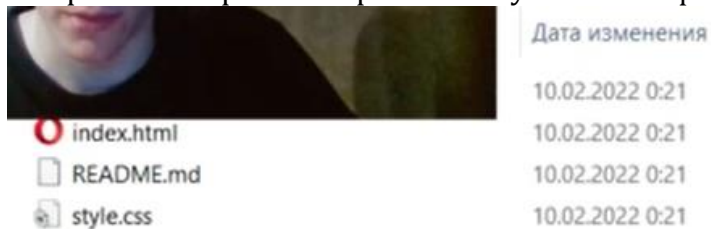
```
D:\Учёба\Математическое моделирование\lab01\hello\lib>cd ..

D:\Учёба\Математическое моделирование\lab01\hello>cd ..

D:\Учёба\Математическое моделирование\lab01>git clone hello cloned_hello
Cloning into 'cloned_hello'...
done.
```

1.28 Просмотр клонированного репозитория

Для этого перейдём в клонированный репозиторий. Я вижу все те же файлы, что и в



исходном репозитории.

Далее

просмотрим историю клонированного репозитория. Она совпадает с исходной

```
D:\Учёба\Математическое моделирование\lab01\cloned_hello\lib>git log --all
commit dba92ff0646234299cbf05f34b3c95d3bfec69bf (HEAD -> master, origin/style, origin/master, origin/HEAD)
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date: Thu Feb 10 00:05:39 2022 +0400

    Updated index.html

commit 72cdddc8b88c43c43e427dfad1ce8d2a94b13690
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date: Thu Feb 10 00:04:00 2022 +0400

    Hello uses style.css

commit c3770e599ac46a84847cbf4121d0b126a666d7db
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date: Thu Feb 10 00:02:30 2022 +0400

    Added css stylesheet

commit b6e1d7df72b99265e2b9d80a92318db80a70b81b
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date: Thu Feb 10 00:08:51 2022 +0400

    Added README

commit c82023aebf5d5643cc96375d48518d42bec2693b
Author: arseniy.pol <arseniy.pol@yandex.ru>
```

1.29 Что такое origin?

Выполним команду `git remote`.

```
D:\Учёба\Математическое моделирование\lab01\cloned_hello\lib>git remote
origin
```

Мы видим, что

клонированный репозиторий знает об имени по умолчанию удаленного репозитория.

Выполним команду `git remote show origin`.

```
D:\Учёба\Математическое моделирование\lab01\cloned_hello\lib>git remote show origin
* remote origin
  Fetch URL: D:/Учёба/Математическое моделирование/lab01/hello
  Push  URL: D:/Учёба/Математическое моделирование/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)
```

Удалённые

репозитории могут с тем же успехом указывать на репозиторий на той же машине. Нет ничего особенного в имени «origin», однако существует традиция использовать «origin» в качестве имени первичного централизованного репозитория (если таковой имеется).

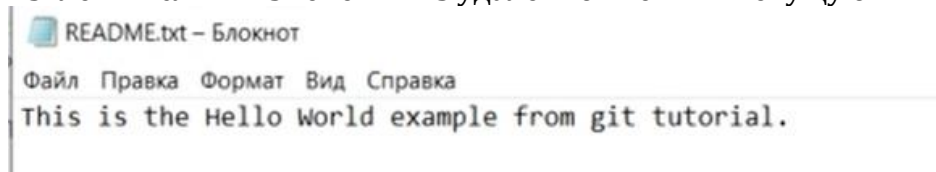
1.30 Удаленные ветки

Для того, чтобы увидеть все ветки, в том числе удалённые, я выполнил следующую команду: `git branch -a`

```
D:\Учёба\Математическое моделирование\lab01\cloned_hello\lib>git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/style
```

1.31 Изменение оригинального репозитория

Я внес некоторые изменения в оригинальный репозиторий, чтобы затем попытаться извлечь и слить изменения из удаленной ветки в текущую



Для того, чтобы

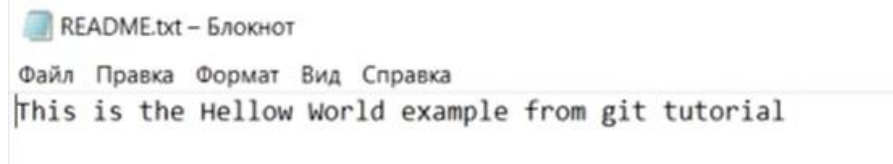
извлечь изменения я воспользовался командой `git fetch`

```
D:\Учёба\Математическое моделирование\lab01\cloned_hello>git fetch
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 462 bytes | 0 bytes/s, done.
From D:/Учёба/Математическое моделирование/lab01/hello
 dba92ff..b737e5f  master    -> origin/master
```

На данный

момент в клонированном репозитории есть все коммиты из оригинального репозитория, но они не интегрированы в локальные ветки клонированного

репозитория. Команда `git fetch` будет извлекать новые коммиты из удаленного репозитория, но не будет сливать их с вашими наработками в локальных ветках. Можно посмотреть файл `README.md` и увидеть, что он не изменился



1.32 Слияние извлеченных изменений

Для начала солью извлеченные изменения в локальную ветку `master` командой `merge`

```
D:\Учёба\Математическое моделирование\lab01\cloned_hello>git merge origin/master
Updating dba92ff..b737e5f
Fast-forward
 lib/README.md | 2 +-
 lib/index.html | 2 +-
 2 files changed, 2 insertions(+), 2 deletions(-)
```

Проверяю

файл `README` и вижу, что он изменился

```
D:\Учёба\Математическое моделирование\lab01\cloned_hello\lib>type README.md
This is the Hello World example from git tutorial.
```

Объединение `fetch` и `merge` в одну команду: `pull`

1.33 Добавление ветки наблюдения

Добавляю локальную ветку, которая отслеживает удаленную ветку. Для этого:

```
D:\Учёба\Математическое моделирование\lab01\cloned_hello\lib>git branch --track style origin/style
branch 'style' set up to track 'origin/style'.

D:\Учёба\Математическое моделирование\lab01\cloned_hello\lib>git branch -a
* master
  style
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/style

D:\Учёба\Математическое моделирование\lab01\cloned_hello\lib>git log --max-count=2
commit d6a669f9c185524b71f4fcd186364ed41c4cba25 (HEAD -> master, origin/master, origin/HEAD)
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date: Thu Feb 10 00:30:56 2022 +0400

    123

commit b737e5fe71ec9600b18ebb130581b7c9bf00ecdb
Author: arseniy.pol <arseniy.pol@yandex.ru>
Date: Thu Feb 10 00:26:07 2022 +0400

    Changed README od original repo
```

1.34 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, а все что им действительно нужно — это дельты изменений и другие бинарные данные репозитория. Вот это и есть «чистый репозиторий».

1.35 Создайте чистый репозиторий

Для создания чистого репозитория воспользуюсь командой

```
D:\Учёба\Математическое моделирование\lab01>git clone --bare hello hello.git
Cloning into bare repository 'hello.git'...
done.
```

Как

правило, репозитории, оканчивающиеся на .git являются чистыми репозиториями

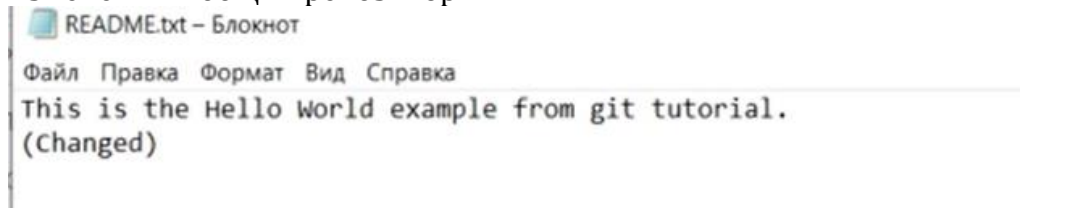
1.36 Добавление удаленного репозитория

Добавляем репозиторий hello.git к нашему оригинальному репозиторию.

```
D:\Учёба\Математическое моделирование\lab01\hello>git remote add shared ../hello.git
```

1.37 Отправка изменений

Для этого я отредактировал файл README.md и сделал коммит. Затем отправил изменения в общий репозиторий



```
D:\Учёба\Математическое моделирование\lab01\hello\lib>git push shared master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 448 bytes | 448.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To ../hello.git
d6a669f..927fe15 master -> master
```

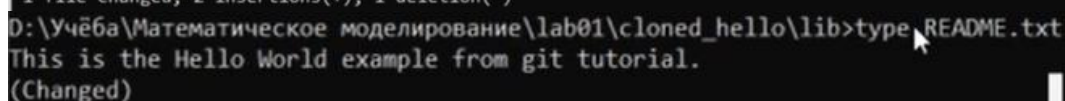
1.38 Извлечение общих изменений

Для этого я выполнил ряд команд

```
D:\Учёба\Математическое моделирование\lab01\cloned_hello>git remote add shared ../hello.git

D:\Учёба\Математическое моделирование\lab01\cloned_hello>git branch --track shared master
fatal: a branch named 'shared' already exists

D:\Учёба\Математическое моделирование\lab01\cloned_hello>git pull shared master
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 428 bytes | 2.00 KiB/s, done.
From ../hello
 * branch                master       -> FETCH_HEAD
 * [new branch]          master       -> shared/master
Updating d6a669f..927fe15
Fast-forward
 lib/README.md | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```



Вывод

В ходе лабораторной работы я ознакомился с основными возможностями и командами git.