

Лабораторная работа 0. Система контроля версий git

Справочный материал:

Официальная документация – книга с подробными пояснениями (русский язык) <https://git-scm.com/book/ru/v2>

Официальная документация – ссылки на статьи по всем командам <https://git-scm.com/docs>

Короткие учебные курсы:

<https://githowto.com/ru>

<https://smartiqa.ru/courses/git/answer-key>

Youtube

- Git: курс (<https://www.youtube.com/playlist?list=PLDyvV36pndZFHX..>) (50 уроков) – аккуратный, необходимый, слегка «продвинутый» курс по Git
- Уроки по Git (<https://www.youtube.com/playlist?list=PLuY6eeDuleIOMB..>) и GitHub (22 урока)
- GIT (<https://www.youtube.com/watch?v=O00FTZDxD0o>) – полный курс Git и GitHub для начинающих (один четырехчасовой урок с временными метками)
- Git (<https://www.youtube.com/watch?v=SEvR78OhGtw>) – большой практический выпуск (один урок с временными метками)

Задания

1. Установить git

Для Windows

<https://git-scm.com/download/win>

или

<https://gitforwindows.org/>

скачать и запустить инсталлятор

Для Mac

<https://git-scm.com/download/mac>

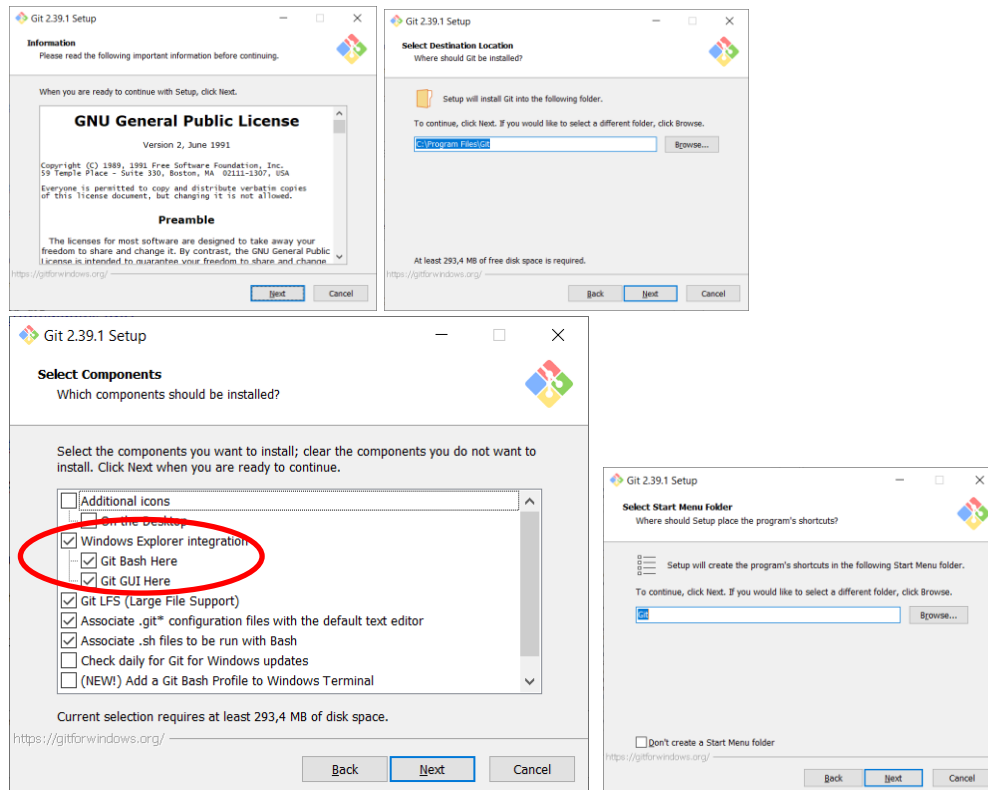
brew install git

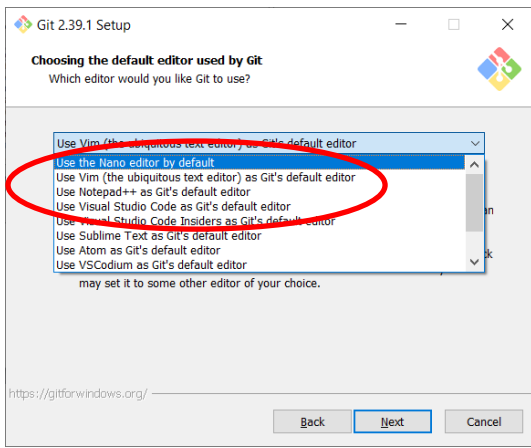
Для Linux

<https://git-scm.com/download/linux>

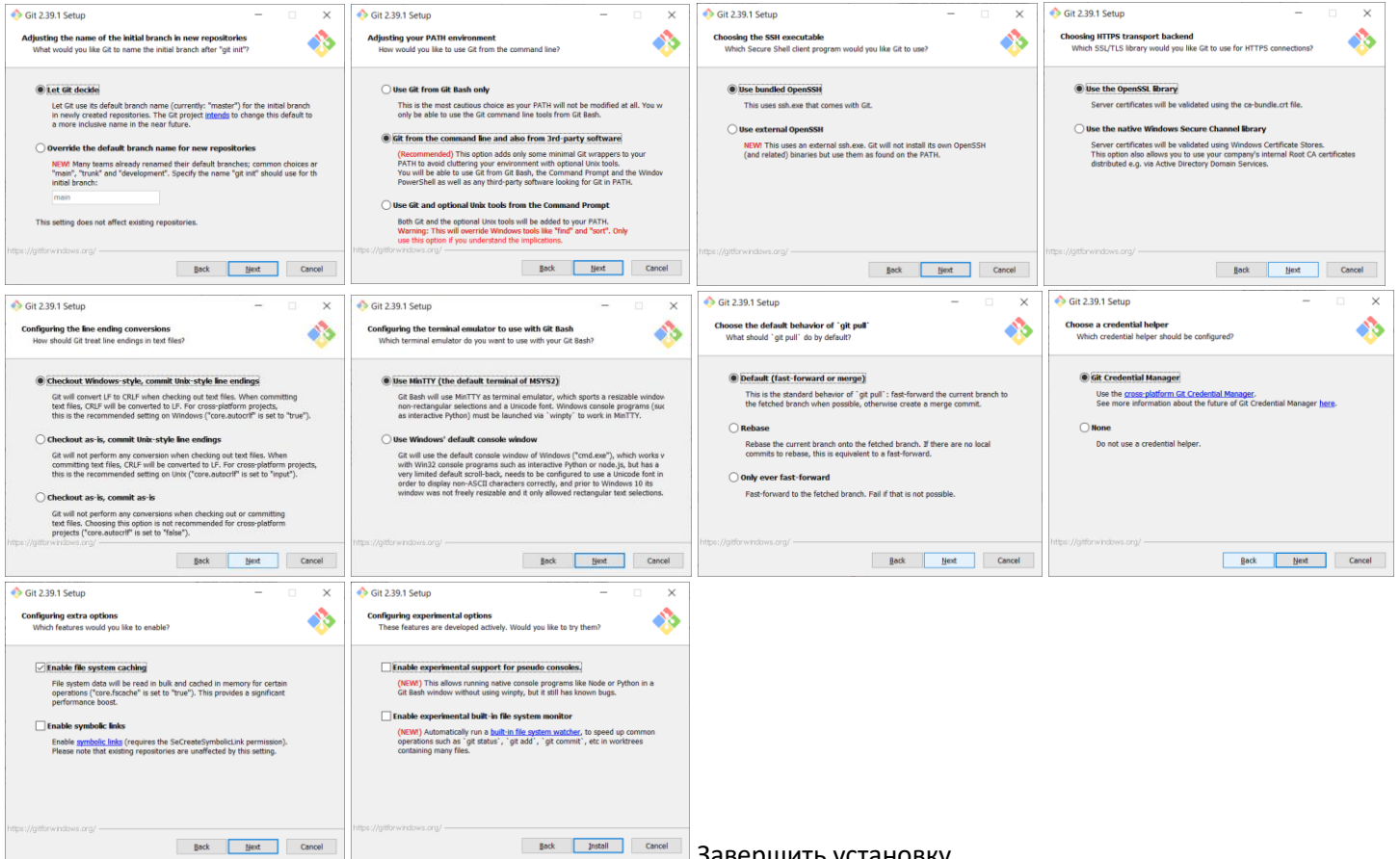
yum install git
apt-get install git
zypper in git

Порядок установки (для Windows)





Выберите удобный редактор текста, рекомендуется VS Code; не рекомендуется Vim (стоит по умолчанию) если Вы не умеете с ним работать.



Завершить установку.

2. Первоначальная настройка git

Параметры, контролирующие все аспекты работы Git и его внешний вид хранятся в конфигурационных файлах (<https://git-scm.com/book/en/v2>) и настраиваются утилитой `git config`

Настроить для своей работы

```
$ git config --global user.name "свое имя"
$ git config --global user.email johndoe@example.com
```

(Указать настоящий email, желательно не из @kpfu.ru; с этим же email нужен аккаунт на github)

3. Создание репозитория, выполнение коммитов, просмотр состояния репозитория и истории коммитов

Создать какую-нибудь папку для выполнения упражнений этого задания:

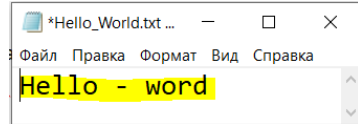
SE_lab01_<группа>_<Фамилия>, например SE_lab01_09251_Ivanov

Упражнение 3.1. Проект «Hello»

1. Создать рабочую папку (будущий репозиторий)

В папке SE_lab01_<группа>_<Фамилия> создать папку с именем Hello

Добавить в нее текстовый файл с содержанием:



2. Открыть **GitBash** в папке Hello через контекстное меню или запустить GitBash и перейти в эту папку командой `cd`

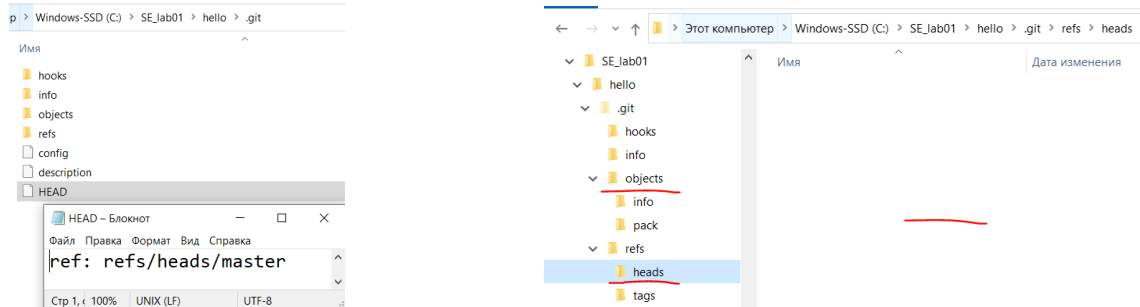
3. Создать репозиторий командой

`git init`

Результат - создана скрытая папка `.git` для хранения локальной базы git-репозитория и его настроек

В ней, в файле `HEAD` текущий указатель на ветку `master`

Но сама ветка пока пустая, нет файлов в папке `.git/refs/`, нет файлов коммитов в папке `.git/objects/`



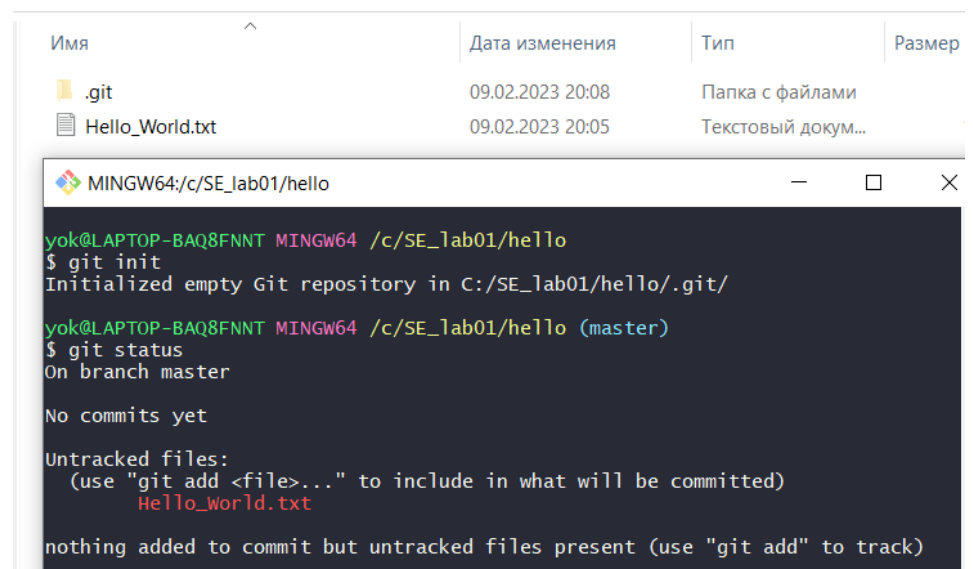
4. Сделать первый коммит:

Проверить состояние репозитория, командой

`git status`

Файл `Hello_World.txt` сейчас не отслеживаемый (Untracked) :

тер > Windows-SSD (C:) > SE_lab01 > hello >



Проиндексировать файл `Hello_World.txt`, сделав его отслеживаемым (staged). Выполнить одну из команд:

`git add Hello_World.txt`

или `git add -A` или `git add *`

Проверить состояние репозитория: `git status`

Теперь файл отслеживается и готов к записи в коммит:

```
MINGW64:/c/SE_lab01/hello

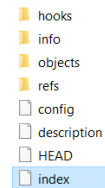
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git add Hello_World.txt

yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Hello_World.txt

yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ |
```



В папке `.git` создан файл `index`, а в `.git/objects` – папка и объект для нулевого (фиктивного пока) коммита

Выполнить коммит командой (в кавычках – краткое описание, это любая строка, описывающая содержание коммита)

`git commit -m "L01-added HelloWorld.txt"`

```
MINGW64:/c/SE_lab01/hello

yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git commit -m "L01-added HelloWorld.txt"
[master (root-commit) aff8a0c] L01-added HelloWorld.txt
1 file changed, 1 insertion(+)
create mode 100644 Hello_World.txt
```

`git status` теперь показывает отсутствие измененных и подготовленных файлов, все закоммичено

```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git status
On branch master
nothing to commit, working tree clean
```

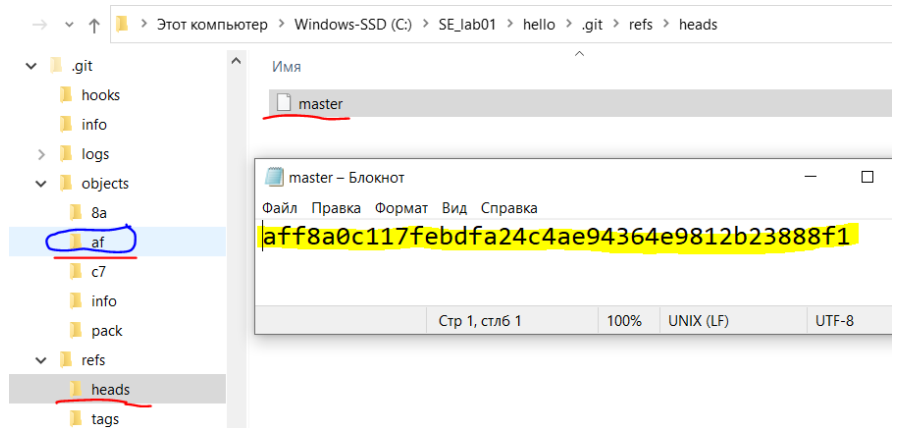
`git show` покажет подробную информацию о последнем коммите, его хеш-время-автора-message и разницу (diff) с предшествующим коммитом

```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git show
commit aff8a0c117febdfa24c4ae94364e9812b23888f1 (HEAD -> master)
Author: omm <olgame2027@gmail.com>
Date: Thu Feb 9 22:15:55 2023 +0300

    L01-added HelloWorld.txt

diff --git a/Hello_world.txt b/Hello_world.txt
new file mode 100644
index 0000000..8ad515f
--- /dev/null
+++ b/Hello_world.txt
@@ -0,0 +1 @@
+Hello, word!!!
\ No newline at end of file
```

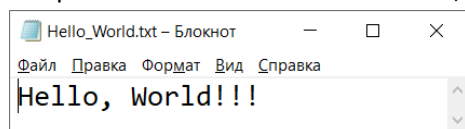
В папке `.git` теперь есть ссылка на этот коммит в файле `refs\heads\master` и сам коммит (в бинарном формате) в отдельной папке `objects\af\`



5. Внести изменения в файл, сделать второй коммит

Изменить файл `HelloWorld.txt`

Исправить слово “word” на “World”, добавить знаки препинания



Проверить состояние:

```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_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_World.txt

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

Проиндексировать файл и снова проверить состояние

```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git add *

yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Hello_World.txt
```

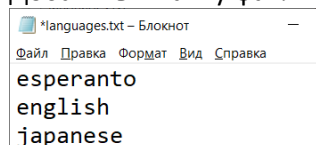
Выполнить второй коммит

`git commit -m "L01 - fixed mistake in the word"`

```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git commit -m "L01 - fixed mistake in the word"
[master 77dc40e] L01 - fixed mistake in the word
1 file changed, 1 insertion(+), 1 deletion(-)
```

6. Внести изменения в файл, добавить еще один файл сделать третий коммит

Добавить в папку файл `languages.txt` с перечислением нескольких языков:



Изменить файл `Hello_World.txt`, добавить в него приветственные надписи на выбранных языках:

```
*Hello_World.txt – Блокнот
Файл  Правка  Формат  Вид  Справка
Saluton, Mondo!!!
Hello, World!!!
こんにちは、世界!!!
```

Проиндексировать файлы `languages.txt` и `Hello_World.txt`
Проконтролировать состояние репозитория:

```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git add *

yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   Hello_world.txt
        new file:   languages.txt
```

Внести дополнительные изменения в файл `languages.txt`

```
*languages.txt – Блокнот
Файл  Правка  Формат  Вид  Справка
esperanto
english
japanese
русский/татарский/родной...|
```

Проконтролировать состояние репозитория – теперь `languages.txt` находится одновременно в двух состояниях: `modified` и `staged`

```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   Hello_world.txt
        new file:   languages.txt

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:   languages.txt
```

Чтобы последние изменения были зафиксированы надо еще раз его проиндексировать:

```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git add languages.txt

yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   Hello_world.txt
        new file:   languages.txt
```

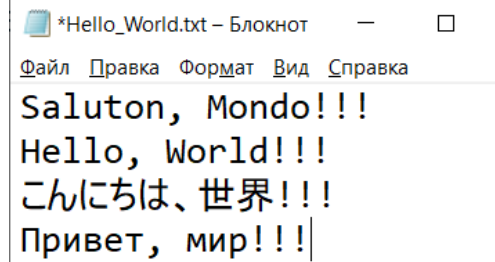
Сделать коммит

`git commit -m "L01 - added languages"`

```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git commit -m "L01 - added languages"
[master 3cd4c8f] L01 - added languages
2 files changed, 7 insertions(+), 1 deletion(-)
create mode 100644 languages.txt
```

7. Внести изменения в файл, сделать четвертый коммит

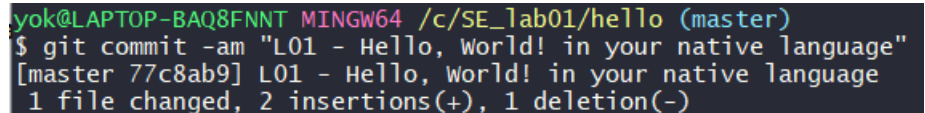
Добавить в файл `Hello_World.txt` надпись на Вашем родном языке



```
*Hello_World.txt - Блокнот
Файл Правка Формат Вид Справка
Saluton, Mondo!!!
Hello, World!!!
こんにちは、世界!!!
Привет, мир!!!
```

Одной командой проиндексировать файл и сделать коммит:

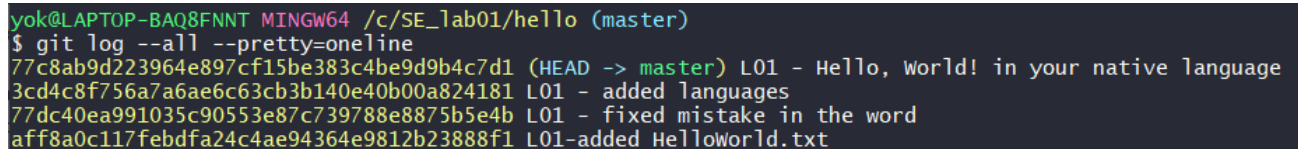
```
git commit -am "L01 - Hello, World! in your native language"
```



```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git commit -am "L01 - Hello, World! in your native language"
[master 77c8ab9] L01 - Hello, World! in your native language
1 file changed, 2 insertions(+), 1 deletion(-)
```

Вывести всю краткую историю изменений – сделано 4 коммита, видны их хеши и описания.

```
git log --all --pretty=oneline
```



```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git log --all --pretty=oneline
77c8ab9d223964e897cf15be383c4be9d9b4c7d1 (HEAD -> master) L01 - Hello, World! in your native language
3cd4c8f756a7a6ae6c63cb3b140e40b00a824181 L01 - added languages
77dc40ea991035c90553e87c739788e8875b5e4b L01 - fixed mistake in the word
aff8a0c117febdfa24c4ae94364e9812b23888f1 L01-added HelloWorld.txt
```

8. Сделать откат к предыдущему коммиту и исправить (переделать) его

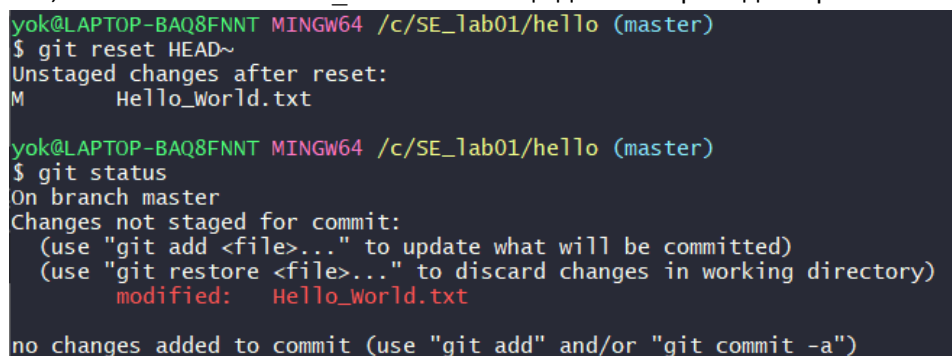
Два последних коммита были сделаны нерационально.

Сделать из них один коммит со всеми последними изменениями в `languages.txt` и `Hello_World.txt` (объединить в один коммит результаты выполнения пунктов 6 и 7)

- 1) Откатить состояние рабочей папки до третьего коммита:

```
git reset HEAD~
```

здесь `languages.txt` уже записан, а изменения в `Hello_World.txt` еще даже не проиндексированы



```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git reset HEAD~
Unstaged changes after reset:
M    Hello_World.txt

yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_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_World.txt

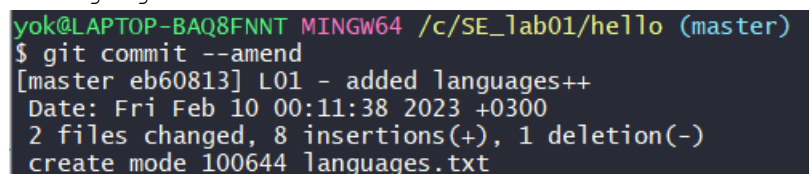
no changes added to commit (use "git add" and/or "git commit -a")
```

- 2) Проиндексировать `Hello_World.txt`

```
git add Hello_World.txt
```

и изменить третий коммит, внеся в него эти добавления (новый - четвертый коммит создаваться не будет)

```
git commit --amend -m "L01 - added languages"
```



```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git commit --amend
[master eb60813] L01 - added languages++
Date: Fri Feb 10 00:11:38 2023 +0300
2 files changed, 8 insertions(+), 1 deletion(-)
create mode 100644 languages.txt
```


- 3) Проверить состояние файлов в рабочей папке (весь текст должен быть на месте), вывести историю изменений и состояние репозитория

```
yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git status
On branch master
nothing to commit, working tree clean

yok@LAPTOP-BAQ8FNNT MINGW64 /c/SE_lab01/hello (master)
$ git log --all --pretty=oneline
eb6081382f369a324de2f4deb42d99f0e751e120 (HEAD -> master) L01 - added languages++
77dc40ea991035c90553e87c739788e8875b5e4b L01 - fixed mistake in the word
aff8a0c117febdfa24c4ae94364e9812b23888f1 L01-added HelloWorld.txt
```

Домашние задания

4. В online-учебном курсе изучить теоретический материал по разделам

Урок 1. Введение. Установка и настройка <https://smartiga.ru/courses/git/lesson-1>

Урок 2. Внутренняя реализация. Индексация. Коммиты. Команды: init, config, status, add, commit
<https://smartiga.ru/courses/git/lesson-2>

5. В online-учебном курсе изучить теоретический материал и выполнить практические задания для уроков:

Урок 3. Ветвление - <https://smartiga.ru/courses/git/lesson-3>

Урок 4. Перемещение курсора и отмена изменений <https://smartiga.ru/courses/git/lesson-4>

Урок 5. Слияние изменений и продвинутая работа с ветками. Команды: merge, cherry-pick, rebase
<https://smartiga.ru/courses/git/lesson-5>

Урок 6. Работа с удаленным репозиторием. Модель ветвления. Команды: clone, fetch, push, pull
<https://smartiga.ru/courses/git/lesson-6>

6. Подготовить аккаунт на ресурсе <https://github.com/>

- При регистрации не использовать почту КФУ

- Login и email, использованный при регистрации установить для рабочей версии git на вашем компьютере:

```
$ git config --global user.name "свое имя"
```

```
$ git config --global user.email johndoe@example.com
```

- Ознакомиться с интерфейсом и основными возможностями ресурса.

- пройти опрос: https://docs.google.com/forms/d/e/1FAIpQLSfqiryqO-bGxJkYG-SLAXK-qwyHZaZZdqeI4QCcagnOupl9iQ/viewform?usp=sf_link