

Лабораторная 3

Для обозначения действий на том или ином шаге будет использоваться эта отметка

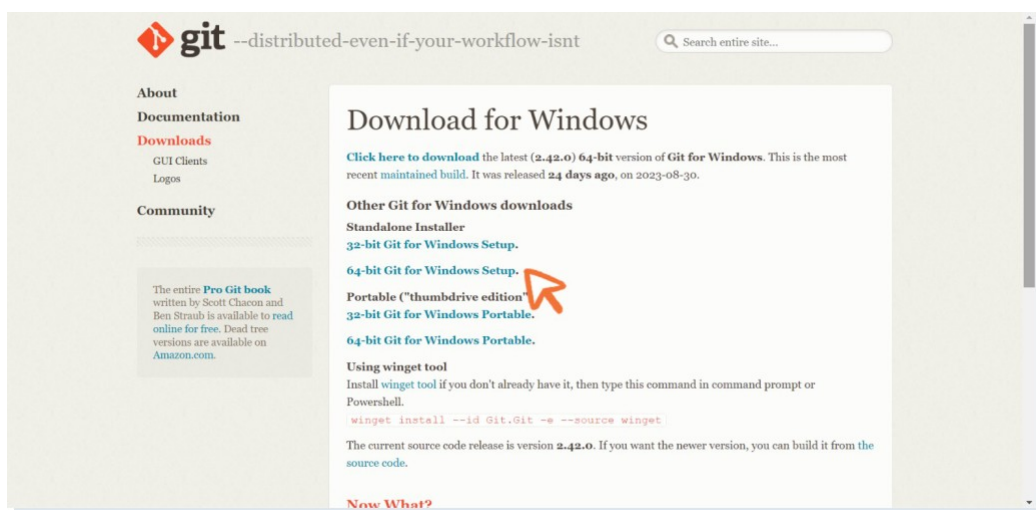


Пункт 1: Установка и проверка Git

1. Установить **Git** можно с официального сайта **git-scm.com**



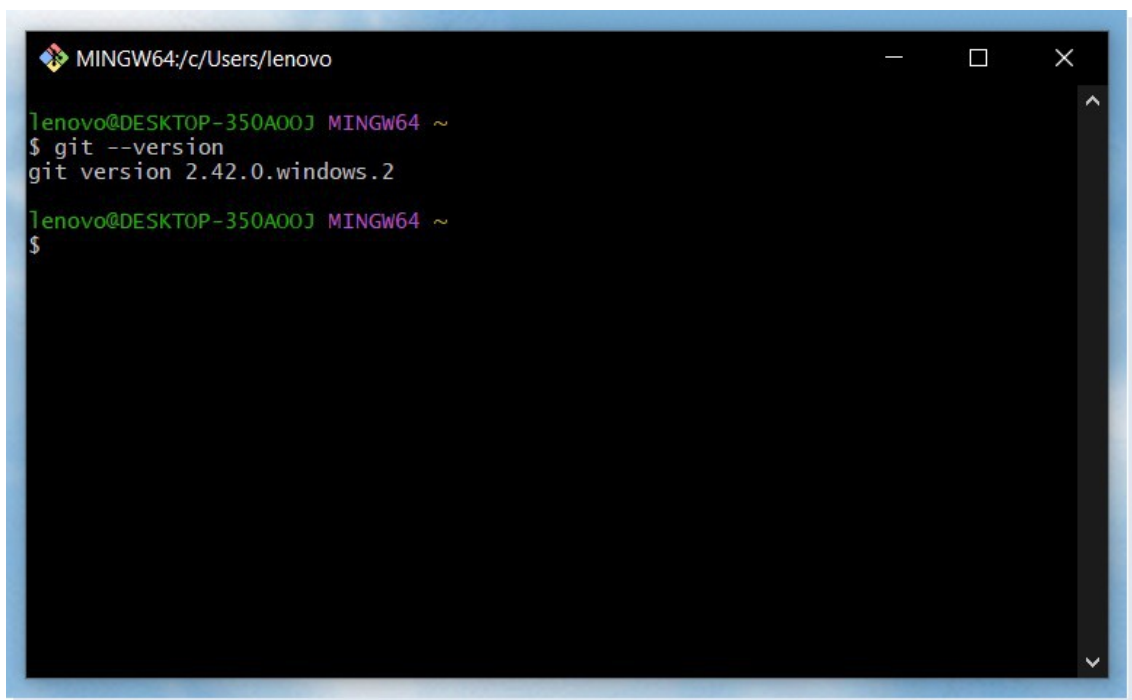
2. Выбираем нужную версию и скачиваем установщик на свой компьютера



3. После прохождения через процесс установки находим и запускаем **Git Bash**

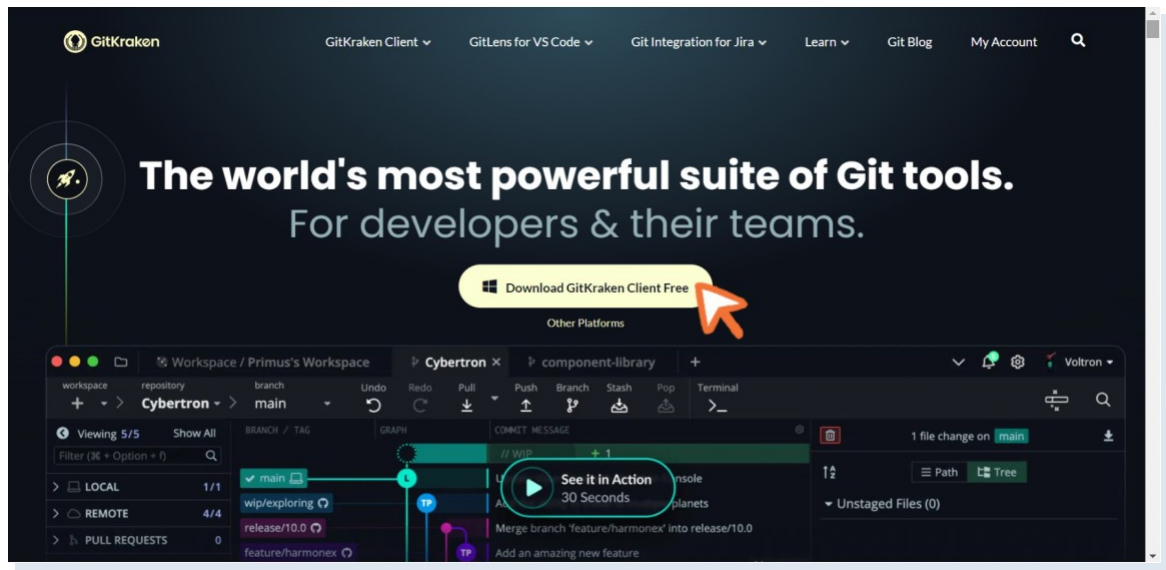


4. Для того, чтобы проверить, что программа установилась корректно, вводим в открывшийся терминал команду **git --version**, которая должна вывести текущую версию ПО

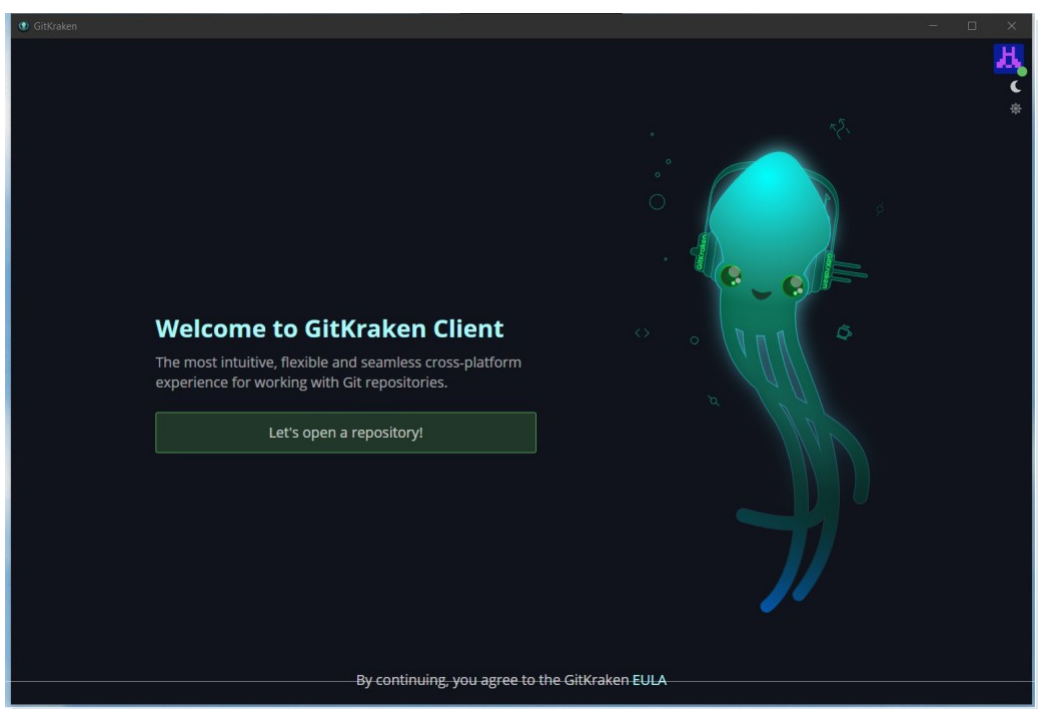
A screenshot of a Windows terminal window titled 'MINGW64: c:/Users/lenovo'. The terminal shows the command 'git --version' being entered and the output 'git version 2.42.0.windows.2'. The prompt is 'lenovo@DESKTOP-350A00J MINGW64 ~'.

Пункт 2: Установка и запуск Git Kraken

5. Теперь можно переходить к установке **Git Kraken** – графического интерфейса для управления git. Для этого переходим на официальный сайт gitkraken.com и скачиваем установщик

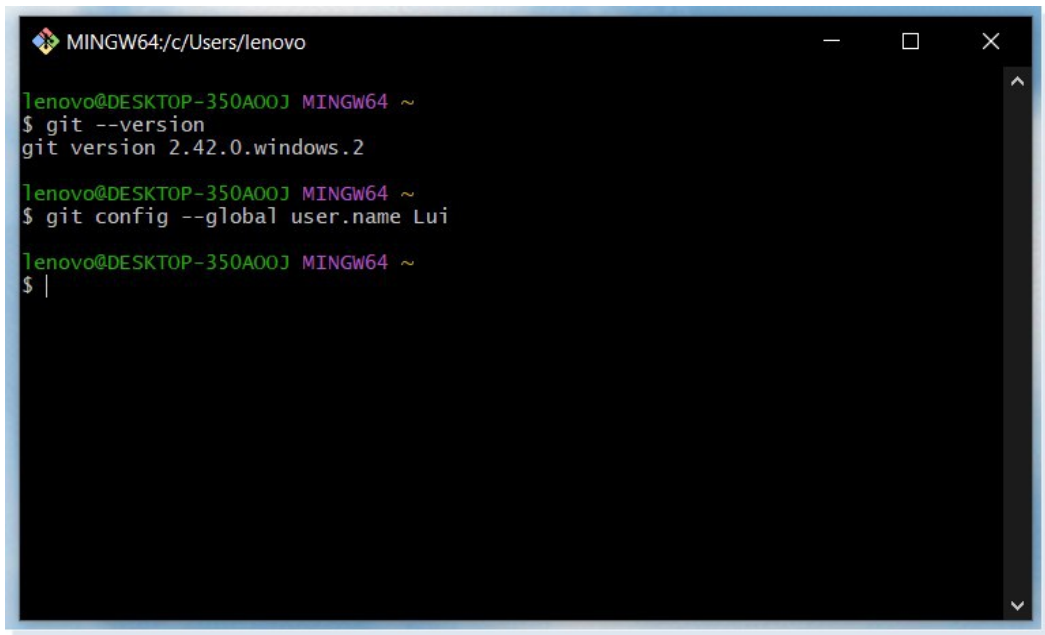


6. Процесс установки пройдет автоматический, после чего программа будет готова к работе



Пункт 2: Настройка пользователя **git**

1. С помощью команды **git config --global user.name [NAME]** можно установить логин, чтобы распознавать пользователя, вносящего изменения в репозиторий



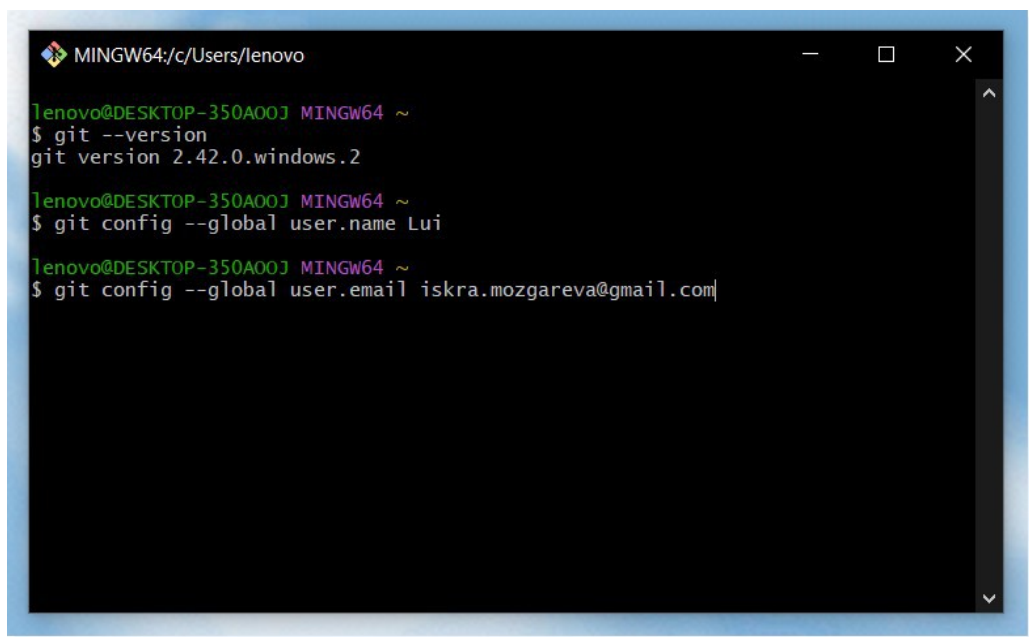
```
MINGW64/c/Users/lenovo

lenovo@DESKTOP-350A00J MINGW64 ~
$ git --version
git version 2.42.0.windows.2

lenovo@DESKTOP-350A00J MINGW64 ~
$ git config --global user.name Lui

lenovo@DESKTOP-350A00J MINGW64 ~
$ |
```

2. С помощью команды **git config --global user.email [EMAIL]** можно установить адрес электронной почты, чтобы с пользователем можно было связаться



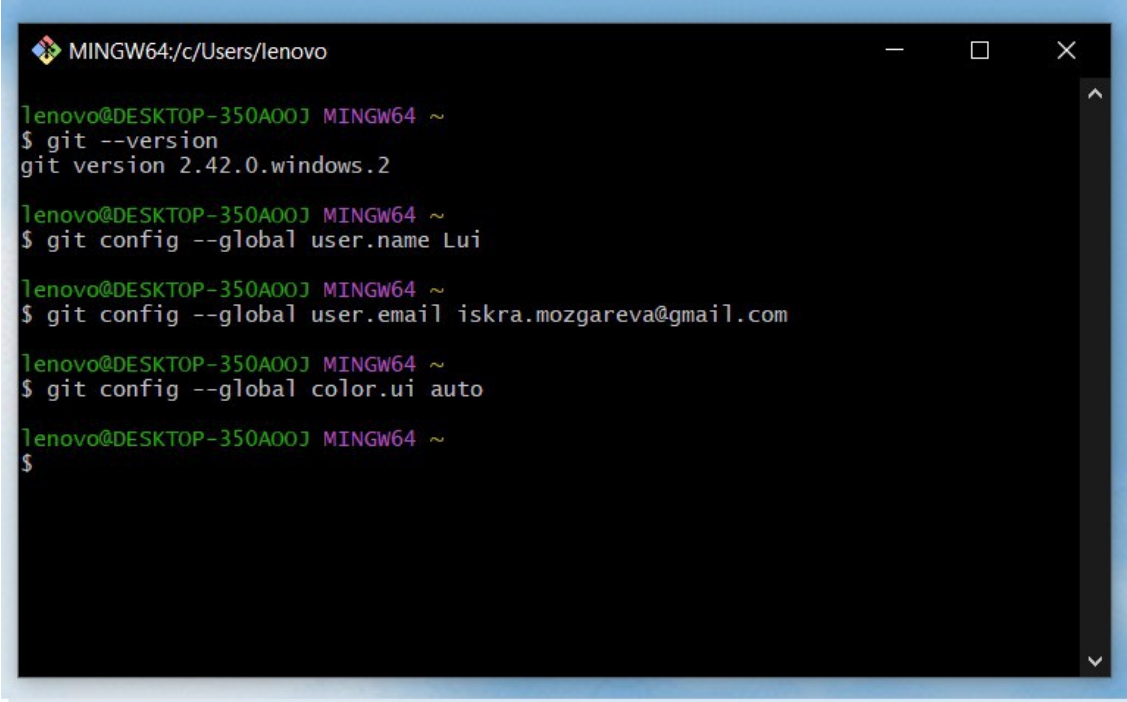
```
MINGW64/c/Users/lenovo

lenovo@DESKTOP-350A00J MINGW64 ~
$ git --version
git version 2.42.0.windows.2

lenovo@DESKTOP-350A00J MINGW64 ~
$ git config --global user.name Lui

lenovo@DESKTOP-350A00J MINGW64 ~
$ git config --global user.email iskra.mozgareva@gmail.com|
```

3. Команда **git config --global color.ui auto** выставляет стандартную цветовую подсветку для текста терминала



```
MINGW64:/c/Users/lenovo

lenovo@DESKTOP-350A00J MINGW64 ~
$ git --version
git version 2.42.0.windows.2

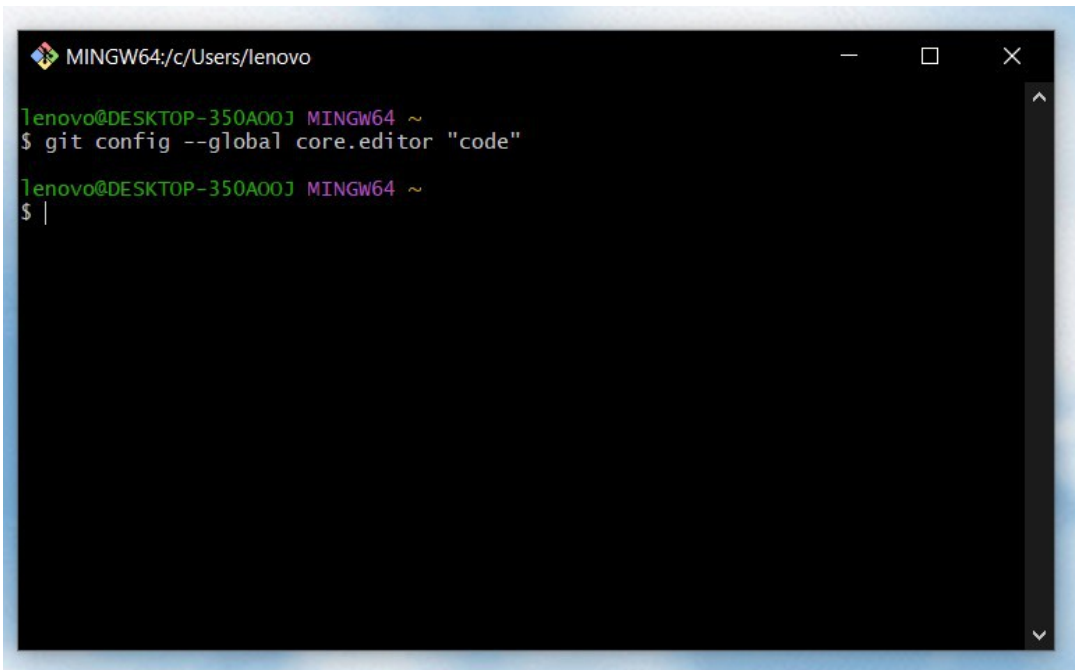
lenovo@DESKTOP-350A00J MINGW64 ~
$ git config --global user.name Lui

lenovo@DESKTOP-350A00J MINGW64 ~
$ git config --global user.email iskra.mozgareva@gmail.com

lenovo@DESKTOP-350A00J MINGW64 ~
$ git config --global color.ui auto

lenovo@DESKTOP-350A00J MINGW64 ~
$
```

4. Команда **git config --global core.editor [EDITOR]** выставляет текстовый редактор для git по умолчанию

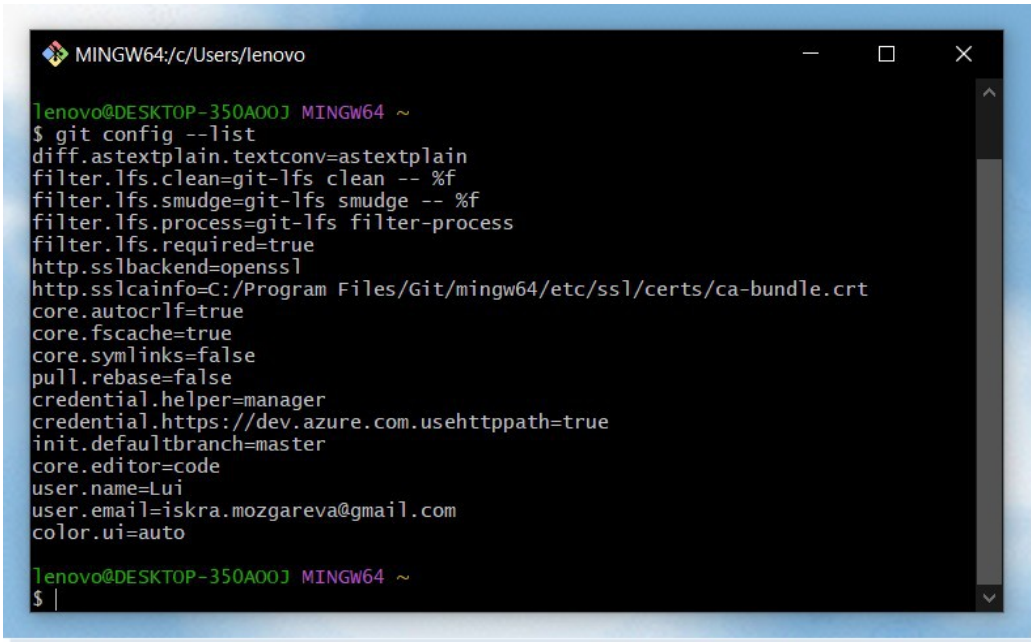


```
MINGW64:/c/Users/lenovo

lenovo@DESKTOP-350A00J MINGW64 ~
$ git config --global core.editor "code"

lenovo@DESKTOP-350A00J MINGW64 ~
$ |
```

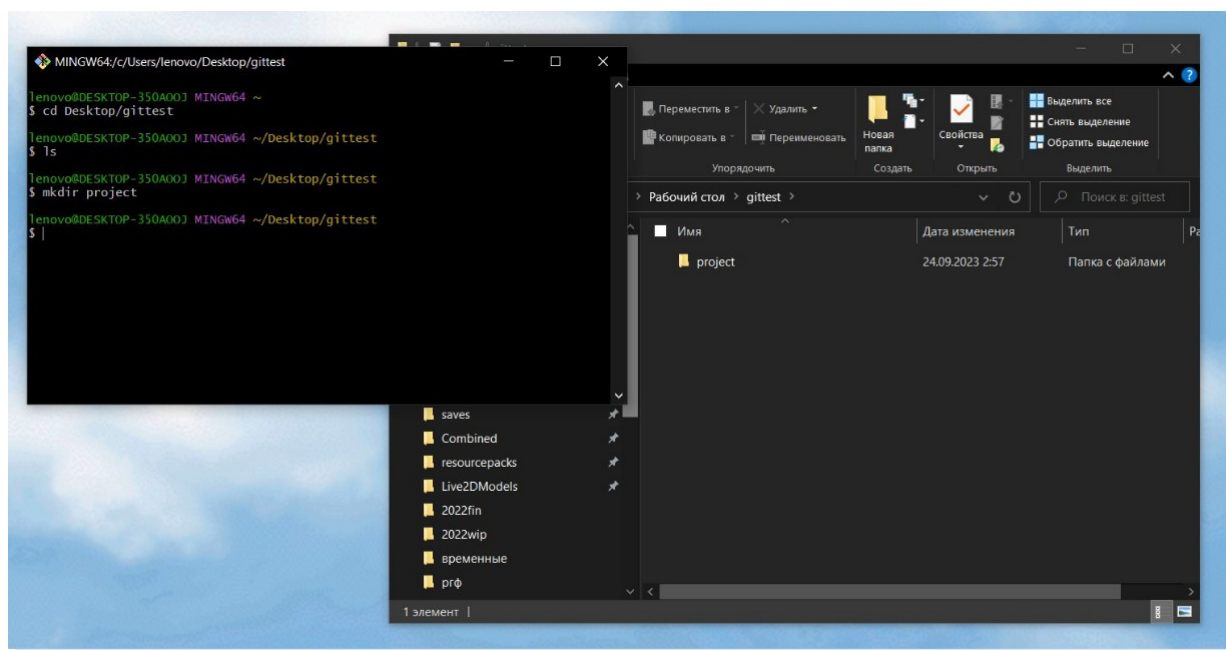
5. Для проверки всех выставленных параметров используется команда **git config --list**



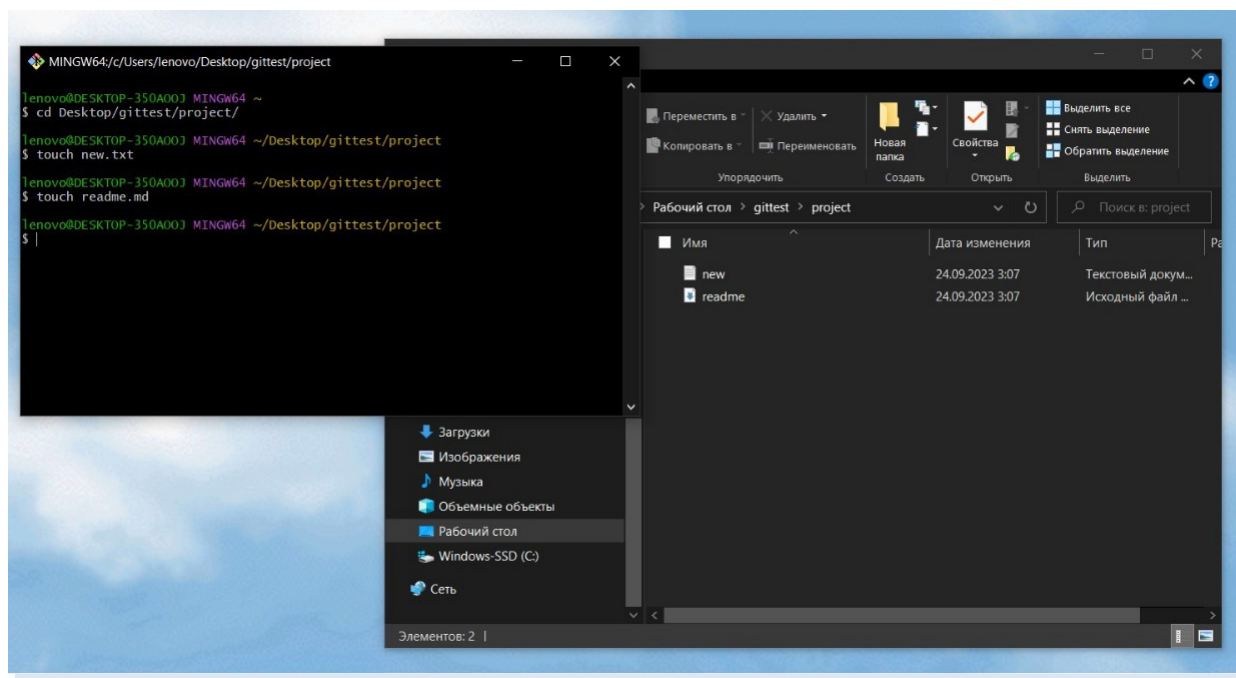
```
lenovo@DESKTOP-350A00J MINGW64 ~  
$ git config --list  
diff.astextplain.textconv=astextplain  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
http.sslbackend=openssl  
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt  
core.autocrlf=true  
core.fscache=true  
core.symlinks=false  
pull.rebase=false  
credential.helper=manager  
credential.https://dev.azure.com.usehttppath=true  
init.defaultbranch=master  
core.editor=code  
user.name=Lui  
user.email=iskra.mozgareva@gmail.com  
color.ui=auto  
  
lenovo@DESKTOP-350A00J MINGW64 ~  
$
```

Пункт 3: Работа с терминалом ос

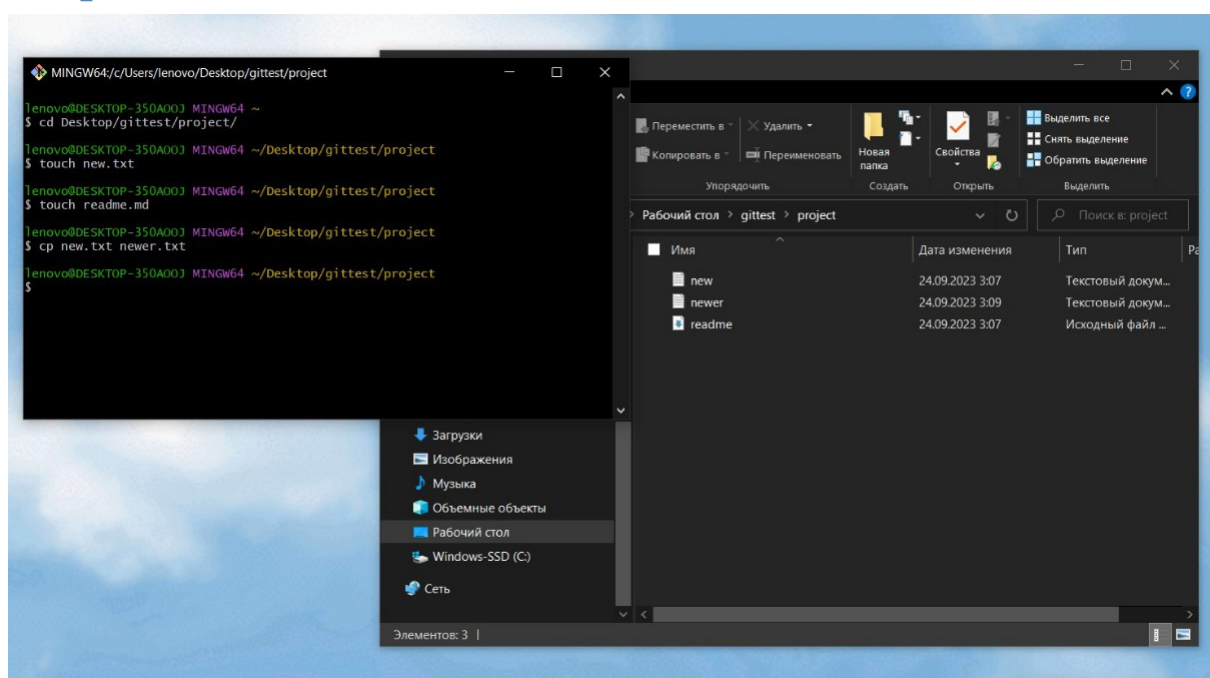
1. Для того, чтобы создать новую директорию, нужно ввести в терминал команду **mkdir [NAME]**



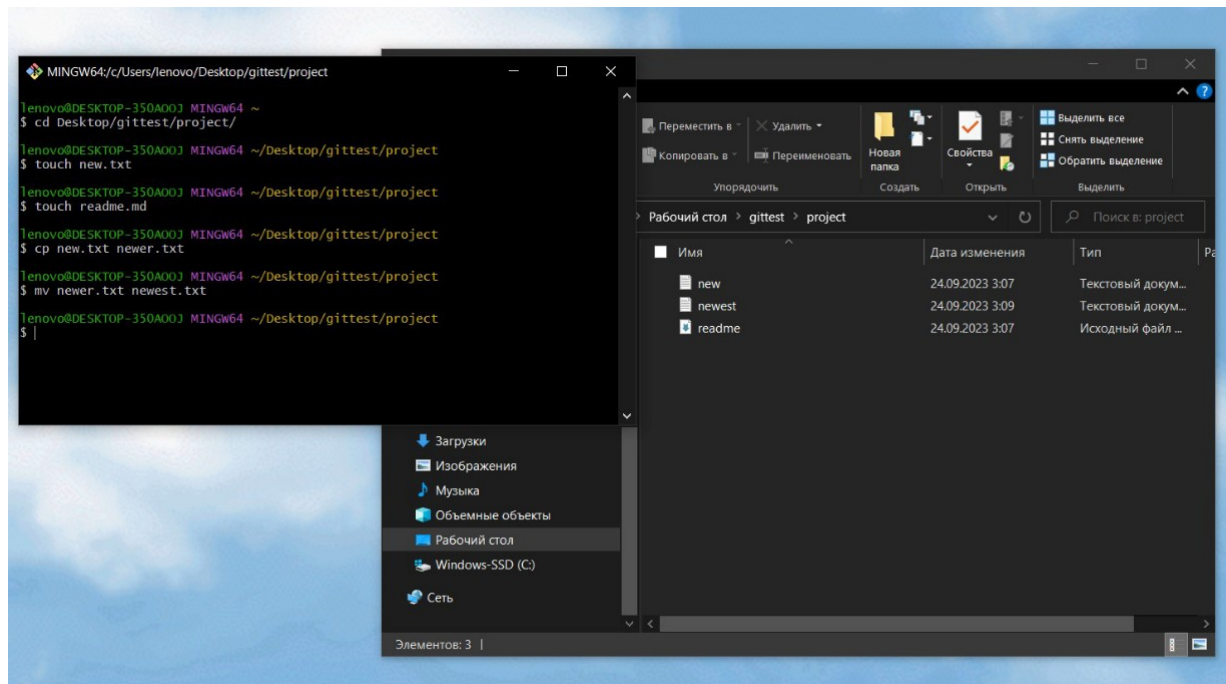
2. Чтобы создать файлы в директории нужно для начала перейти в неё с помощью команды **cd [PATH]**
Затем создать новый файл можно будет с **touch [FILENAME]**



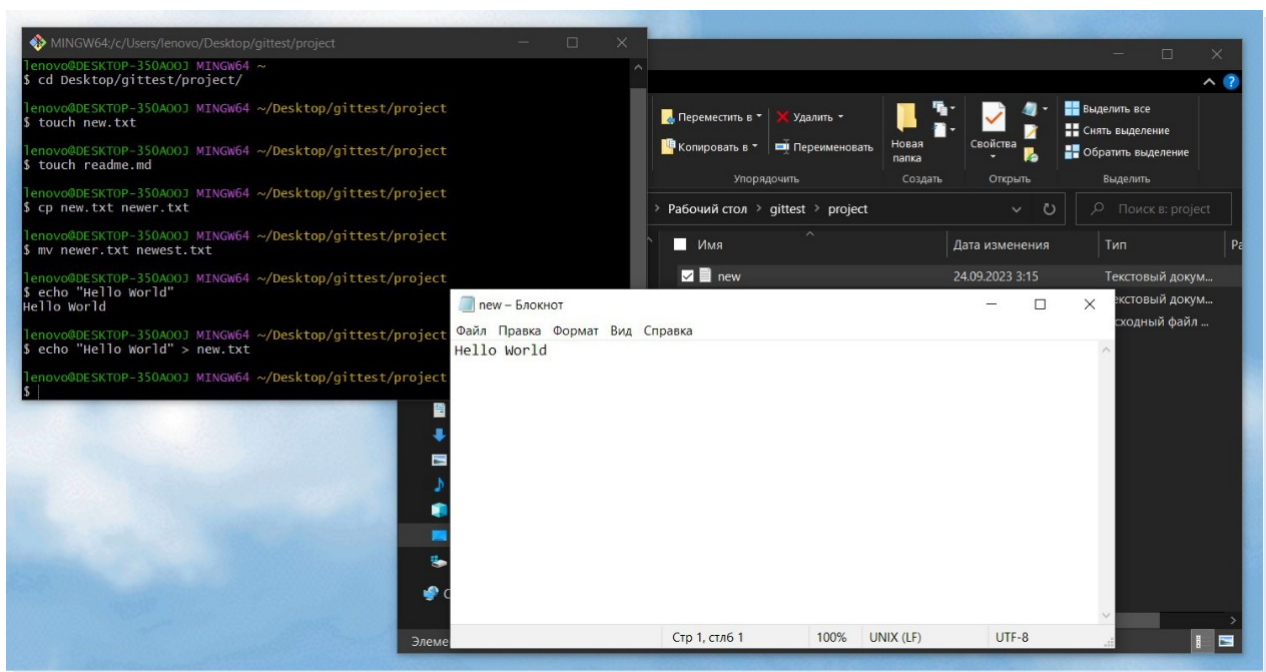
3. Для того, чтобы скопировать файл, нужно использовать команду **cp [COPIED FILE NAME] [NEW FILE NAME]**



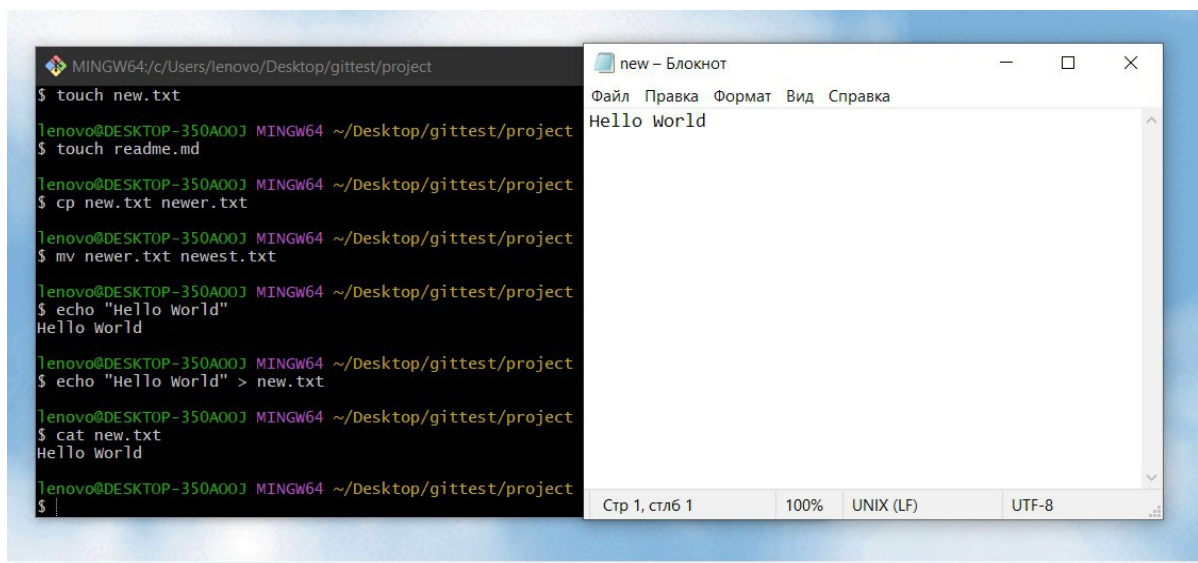
4. Чтобы переименовать файл, используется команда **mv**
[OLD NAME] [NEW NAME]



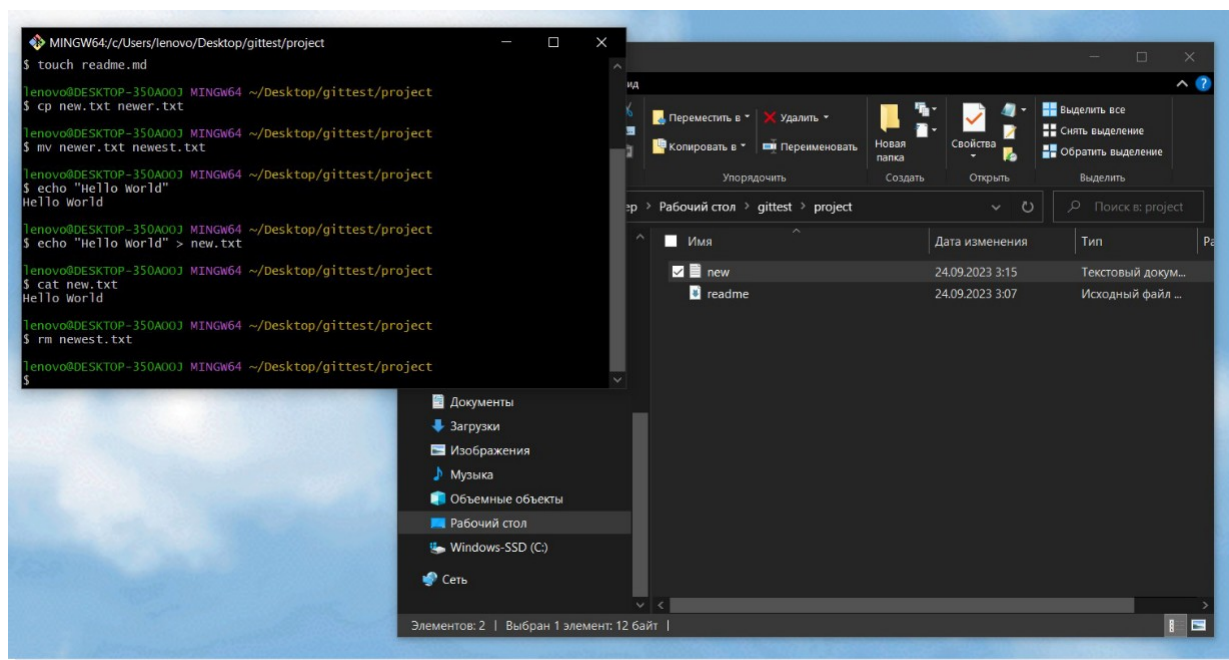
5. Для того, чтобы написать что-то в файле, нужно воспользоваться командой **echo ["TEXT"] > [FILE]**



6. Для того, чтобы просмотреть содержимое файла в терминале, достаточно ввести команду **cat [FILE]**



7. Для того, чтобы удалить файл, есть команда **rm [FILE]**



Пункт 4: Создание и работа с репозиторием на сервисе **GitHub.com**

1. Для того, чтобы создать новый репозиторий на сайте (при учёте наличия аккаунта) нужно зайти в личный кабинет и нажать **New**

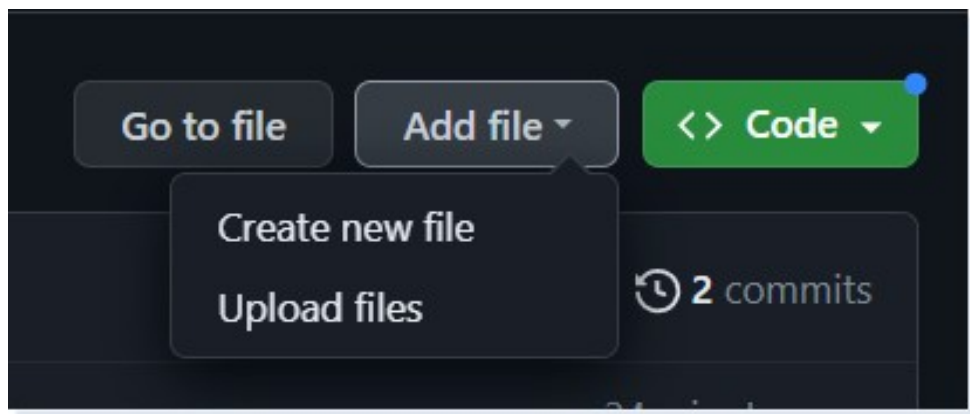


2. При создании обязательно нужно выбрать название и отметить галочку возле **Add a README file**

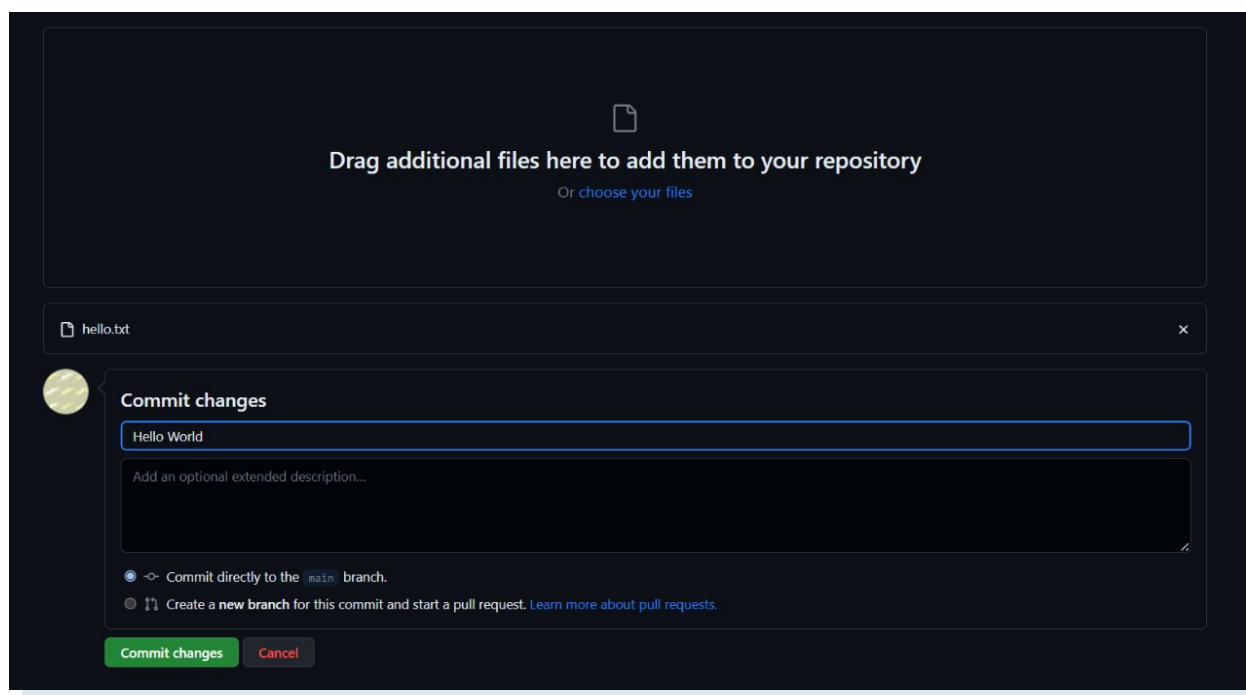
A screenshot of the 'Create a new repository' form on GitHub. The form has a dark background. At the top, it says 'Create a new repository' and provides a brief explanation of what a repository is. Below this, it asks for the 'Owner' (a dropdown menu showing 'Lui-Functional') and the 'Repository name' (a text input field with 'main' entered). A green checkmark indicates 'main is available'. There is a section for 'Description (optional)' with a text input field. Below that, there are two radio buttons for 'Public' (selected) and 'Private'. At the bottom, there is a section 'Initialize this repository with:' with a checked checkbox for 'Add a README file'.

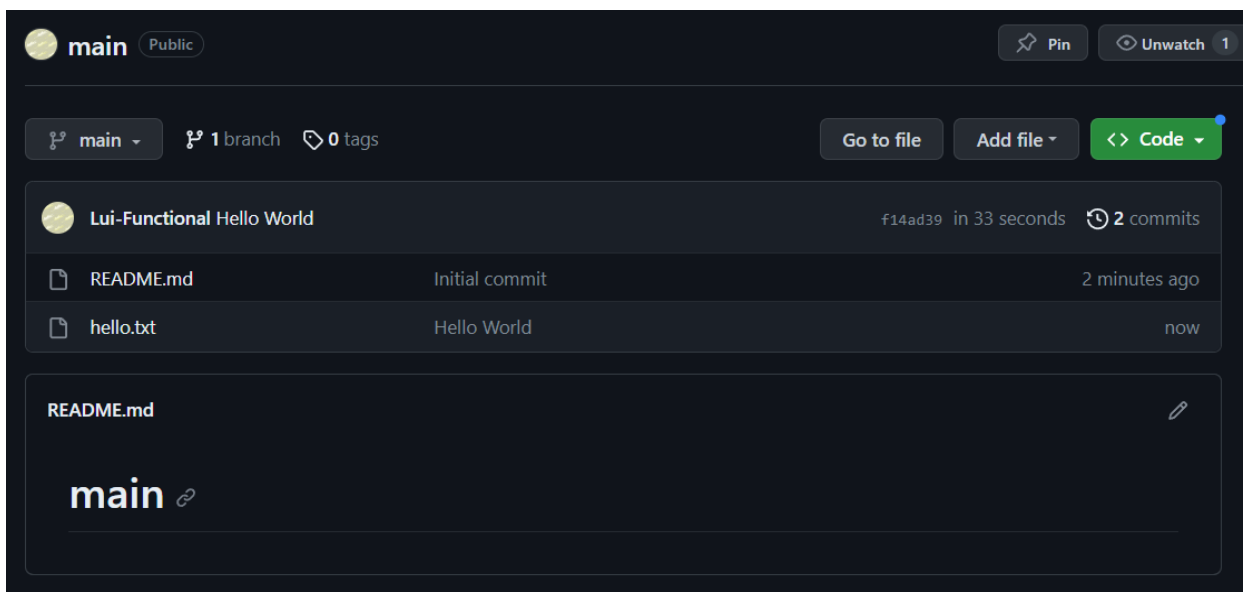
3. После создания общедоступном репозитории, ссылку на неё можно будет использовать для привязки к локальной версии на компьютере пользователя.

4. Чтобы добавить новый файл с компьютера, нужно нажать на **Add file** и выбрать опцию **Upload files**

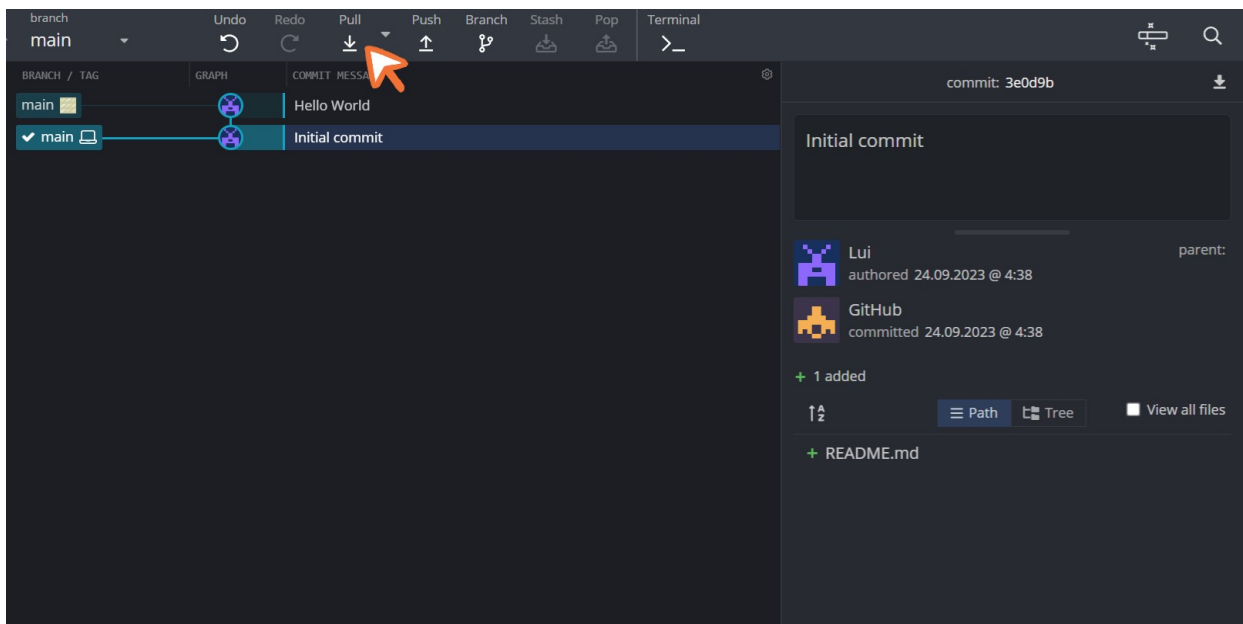


5. Затем можно будет просто перетащить нужный файл в поле загрузки, дать название, опционально описание и закоммитить изменения, после чего новый файл должен отобразиться в репозитории

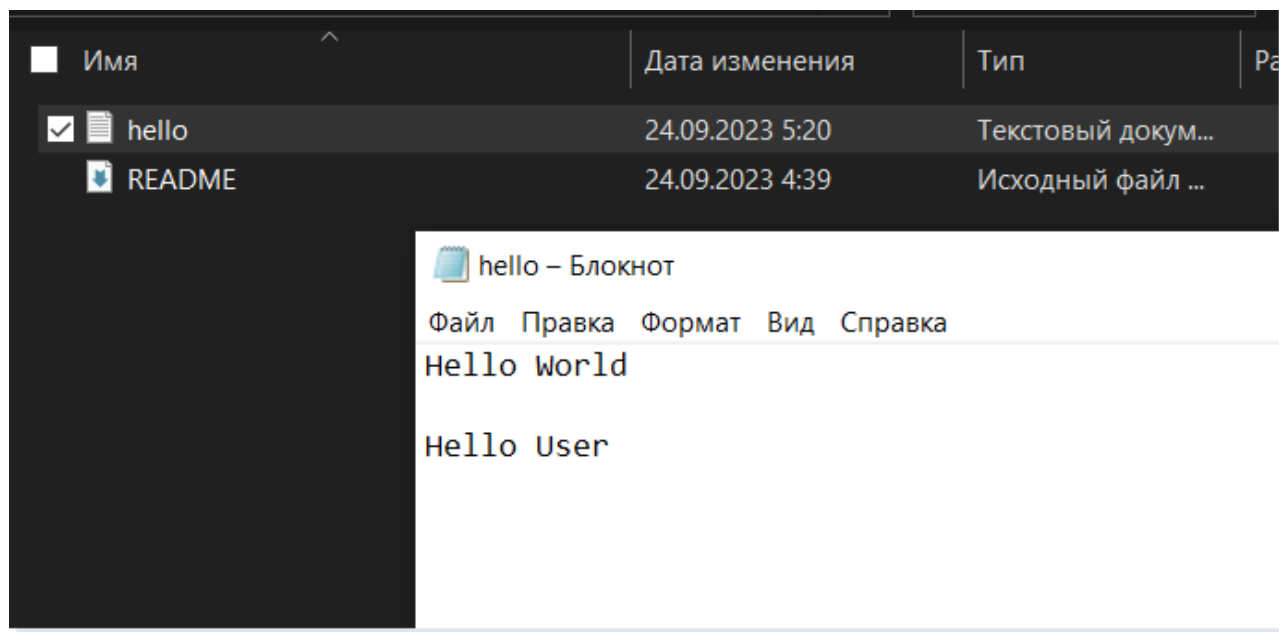




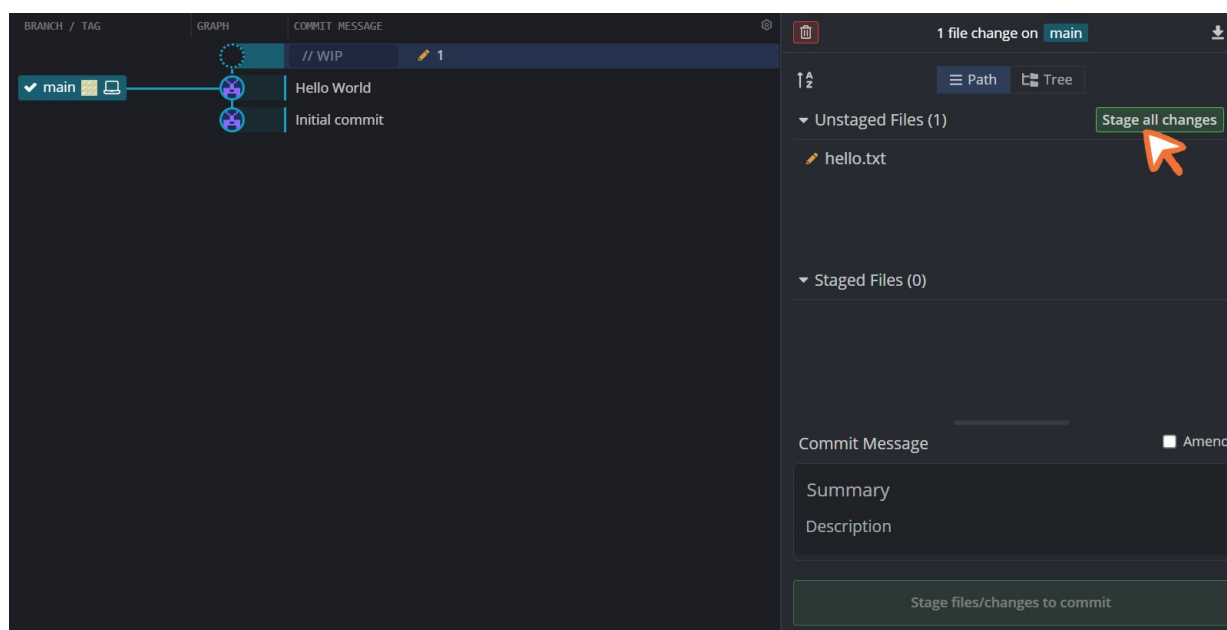
6. Если мы вернёмся в **Git Kraken** и обновим страницу, мы сможем увидеть новый коммит в списке и перенести все изменения в свой локальный репозиторий с помощью **Pull**

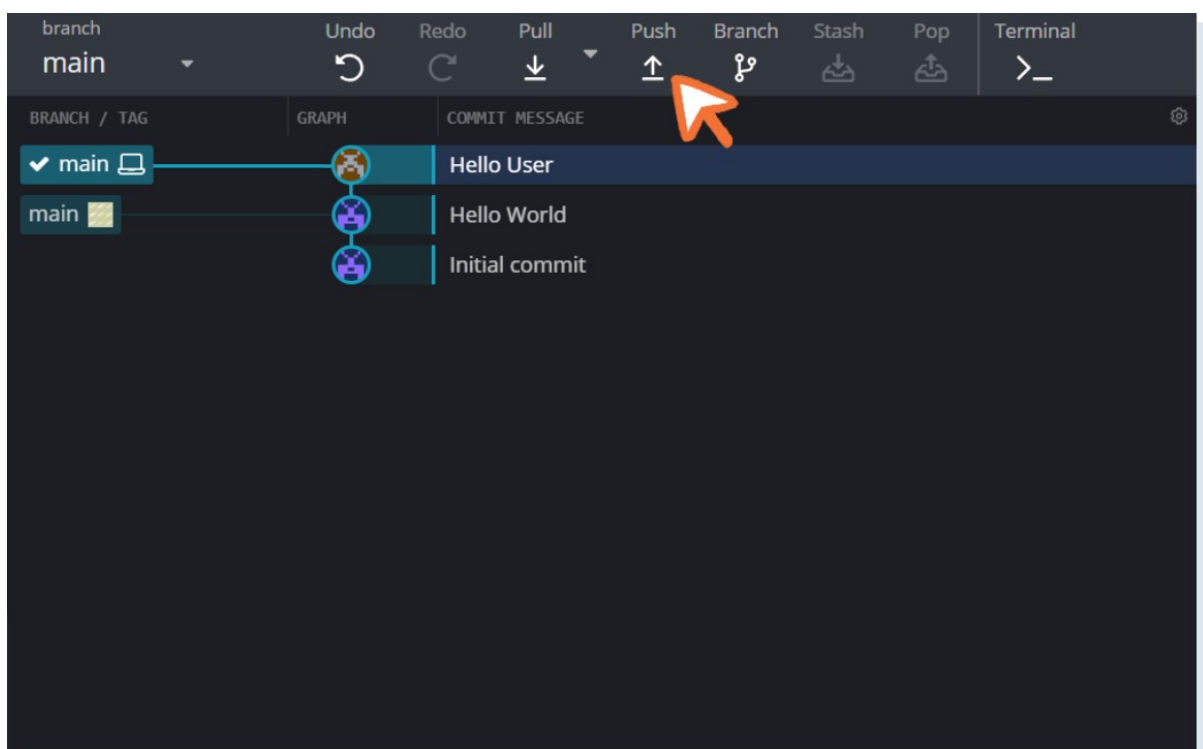
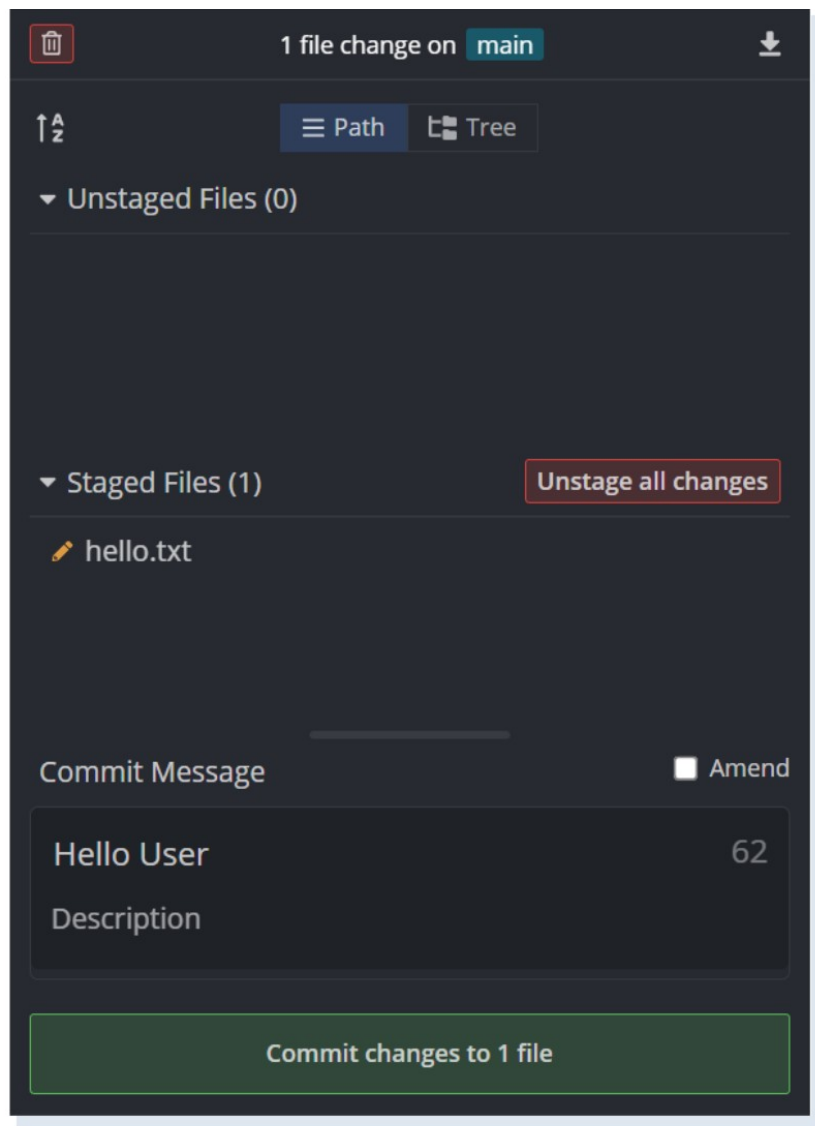


7. Теперь, если открыть репозиторий на компьютере, там будет новый файл, в который можно будет внести изменения

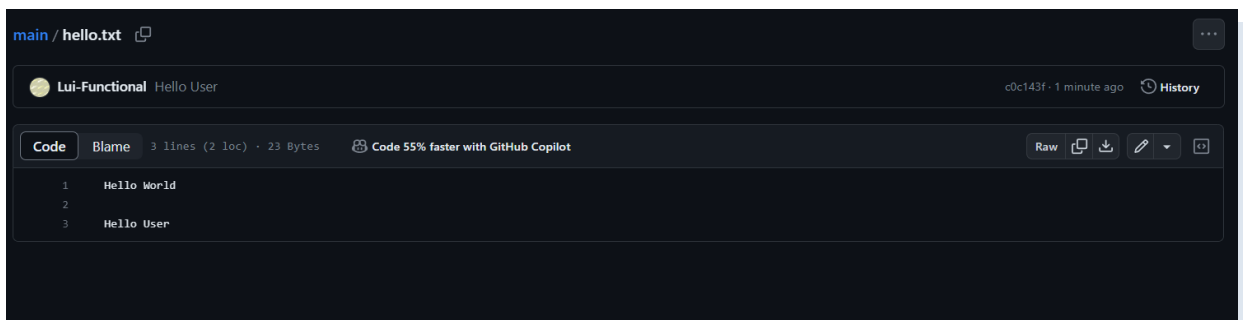
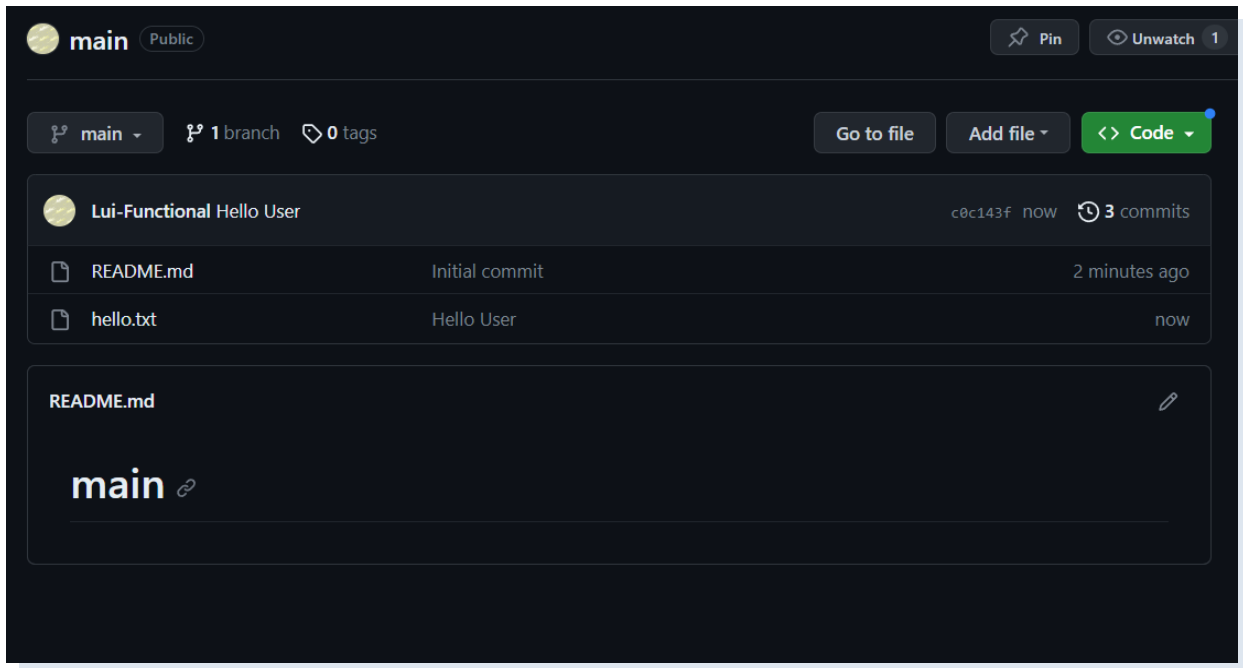


8. Теперь в **Git Kraken** мы можем сохранить эти изменения и передать их в общедоступный репозиторий



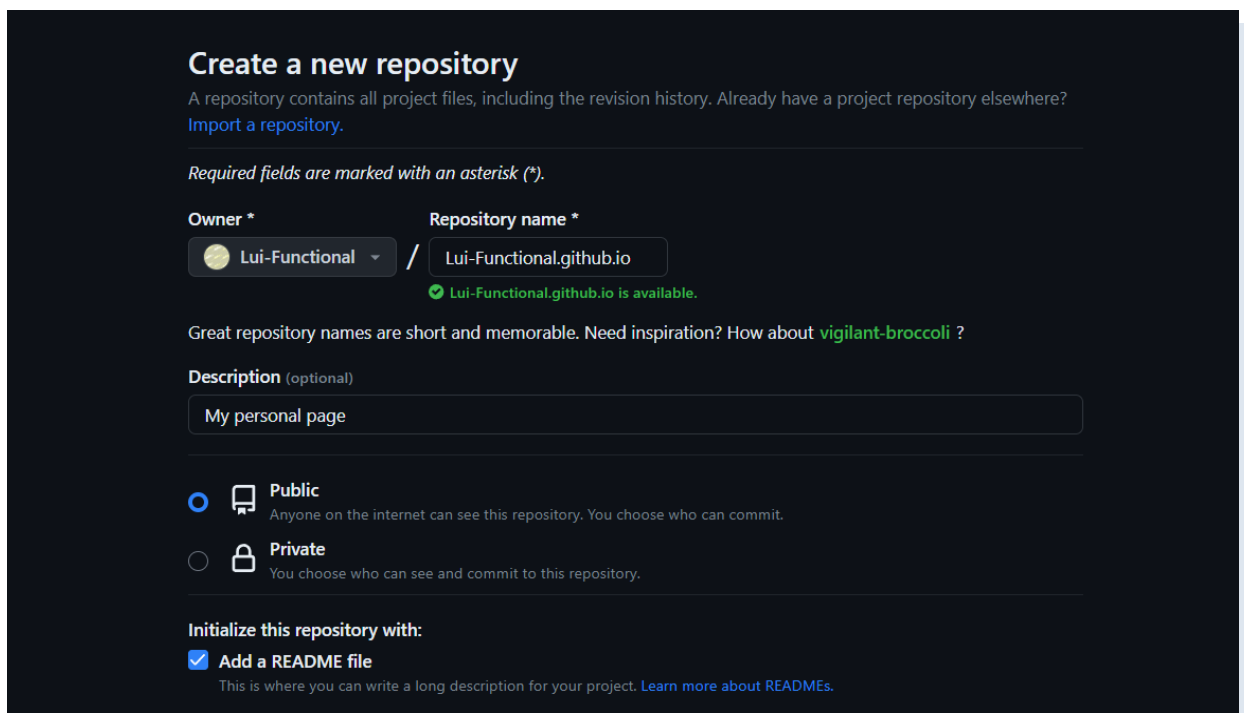


9. После этого изменения должны отразиться и на странице в **GitHub**



Пункт 5: Создание веб-страницы с помощью GitHub

1. Для начала создаём новый репозиторий с названием `[username].github.io`




Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner * / Repository name *


 Lui-Functional / Lui-Functional.github.io


✔ Lui-Functional.github.io is available.

Great repository names are short and memorable. Need inspiration? How about [vigilant-broccoli](#) ?

Description (optional)

My personal page

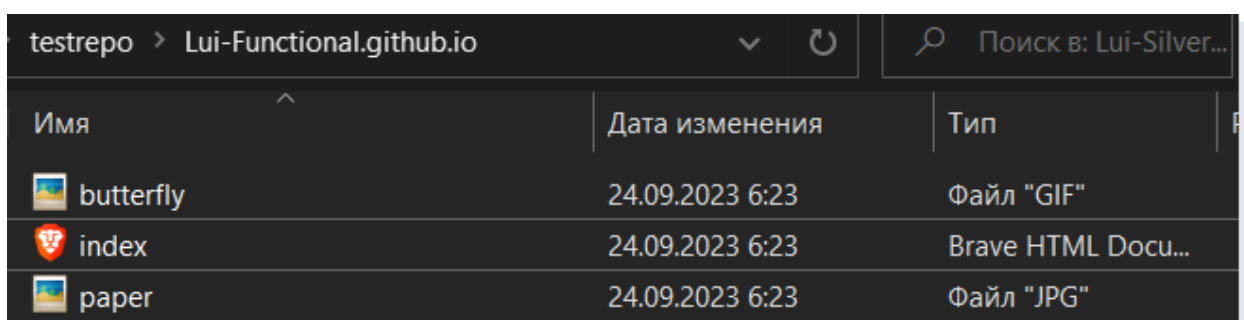
☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.




☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

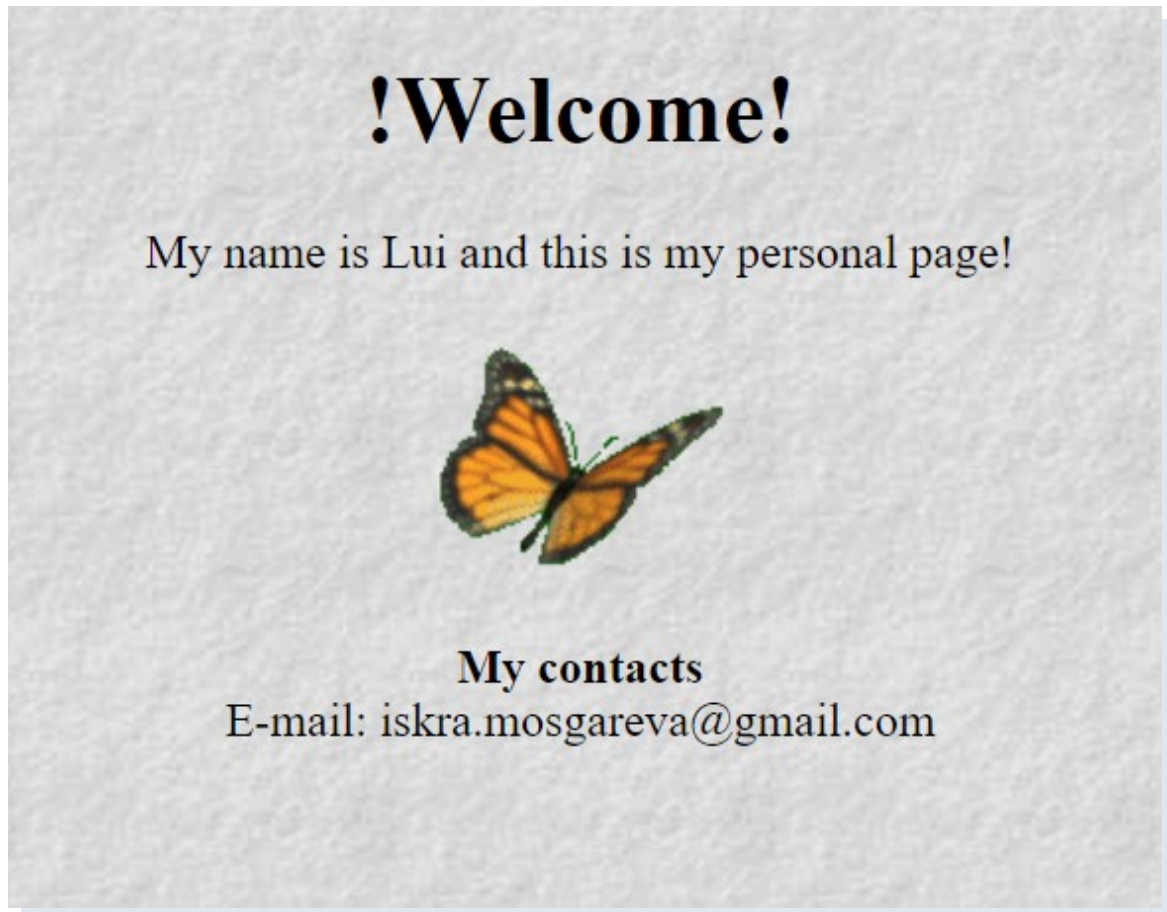
☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

2. Затем объединяем его с локальным репозиторием и добавляем туда **html** файл и все прилагающиеся файлы (изображения, аудиофайлы, и т.д.)



Имя	Дата изменения	Тип
 butterfly	24.09.2023 6:23	Файл "GIF"
 index	24.09.2023 6:23	Brave HTML Docu...
 paper	24.09.2023 6:23	Файл "JPG"

3. Сохраняем и проверяем адрес, совпадающий с названием репозитория. В данном случае это **lui-functional.github.io**



Таким образом **GitHub** можно использовать, как хостинг для своих веб-страниц