

Python-0

lesson 0

Основы работы с Github, Git.

Что такое Git? Зачем он нужен?

Представим что ты рисуешь картину...

сохранить

+фон
картины

автор:
Tom1

сохранить

+дерево на
картине



автор:
Tom1

сохранить

+люди



автор:
Tom1

И в этом моменте **Tom1** решил
вернуть к 3 шагу чтобы поменять
людей..



сохранить

+изменил
людей на
картинке



автор:
Tom1

красный цвет (коммиты) (или комментарии)	commit
оранжевый цвет (автор, тот кто это делал)	author
синий цвет (push - кнопка, или кнопка отправки, кнопка сохранения)	push

Основные методы работы#1 с GIT

git init - Создание нового репозитория.

CMD: **git init**

git clone - Клонирование удаленного репозитория.

CMD: **git clone** <URL>

git add - Добавление файлов для отслеживания.

CMD: **git add** <file1> <file2>

Основные методы работы#2 с GIT

git commit - Фиксация изменений.

CMD: **git commit** -m "Описание изменений"

git status - Проверка состояния репозитория.

CMD: **git status**

git pull - Получение изменений из удаленного репозитория и объединение их с текущей веткой.

CMD: **git pull**



Основные методы работы#3 с GIT

git push - Отправка локальных изменений в удаленный репозиторий.

CMD: **git push**

git branch - Просмотр списка веток.

CMD: **git branch**

git checkout - Переключение между ветками или коммитами.

CMD: **git checkout** <branch-name> # Переключение на ветку

CMD: **git checkout** <commit-hash> # Переключение на коммит

git merge - Объединение веток.

CMD: **git merge** <branch-name>

Основные методы работы#4 с GIT

git log - Просмотр истории коммитов.

CMD: **git log**

git reset - Отмена коммитов и изменений.

CMD: **git reset** <commit-hash>

git stash - Временное сохранение изменений.

CMD: **git stash**

подробности о методах есть в документации GIT: <https://git-scm.com/docs/git>

Установка **GIT**.

Офиц. сайт **GIT**:
<https://git-scm.com/downloads>

Как проверить что **GIT** установлен?

CMD: git -v

```
git version 2.37.3.windows.1
```

Что такое Github? Зачем он нужен?



GitHub

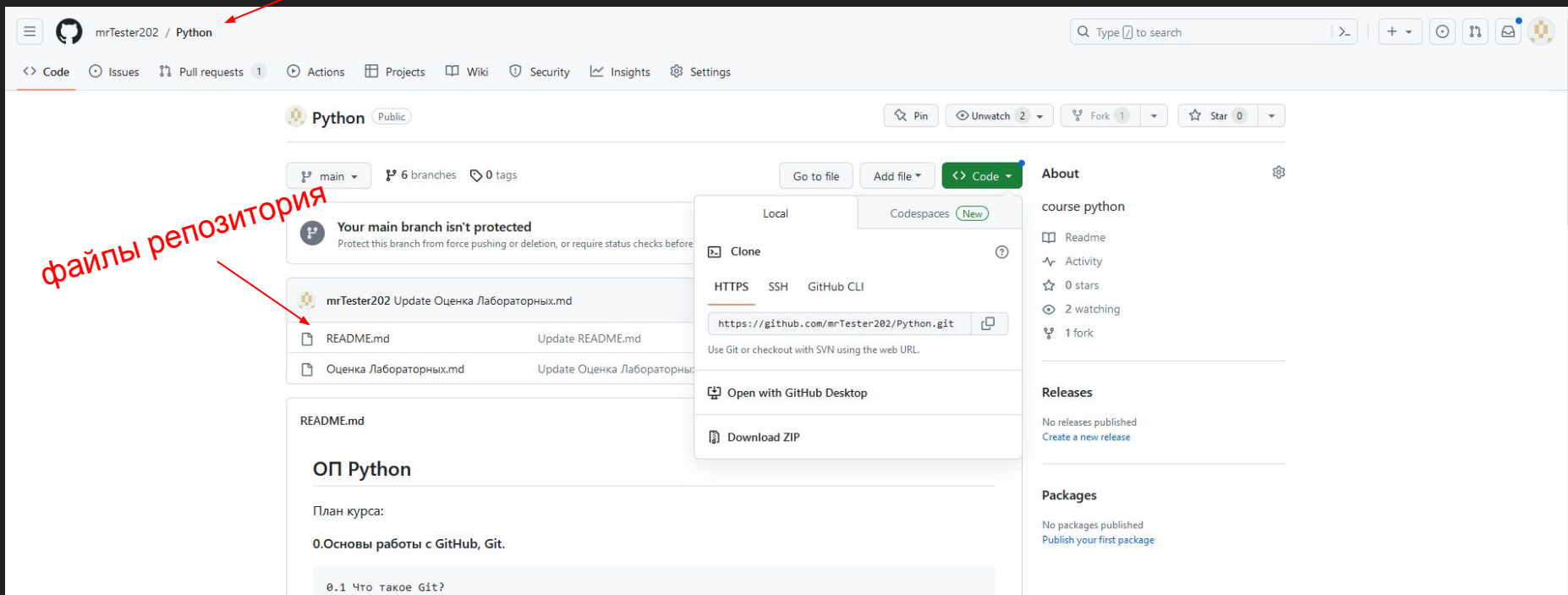
Что значит репозиторий? Что значит
форкнуть репозиторий?

Репозиторий — это все файлы, находящиеся под контролем версий, вместе с историей их изменения и другой служебной информацией. *(иными словами все файлы вашего проекта)*

Пример репозитория

репозиторий - Python

файлы репозитория



Ветка в репозитории - это как отдельная линия развития вашего проекта. Если вы хотите добавить что-то новое или починить ошибку, создаете свою ветку, работаете в ней и, когда всё готово, объединяете ее с основной линией (веткой). Это помогает в управлении изменениями и предотвращает путаницу. (т.е каждая ветка может отвечать за отдельную часть работы в проекте)

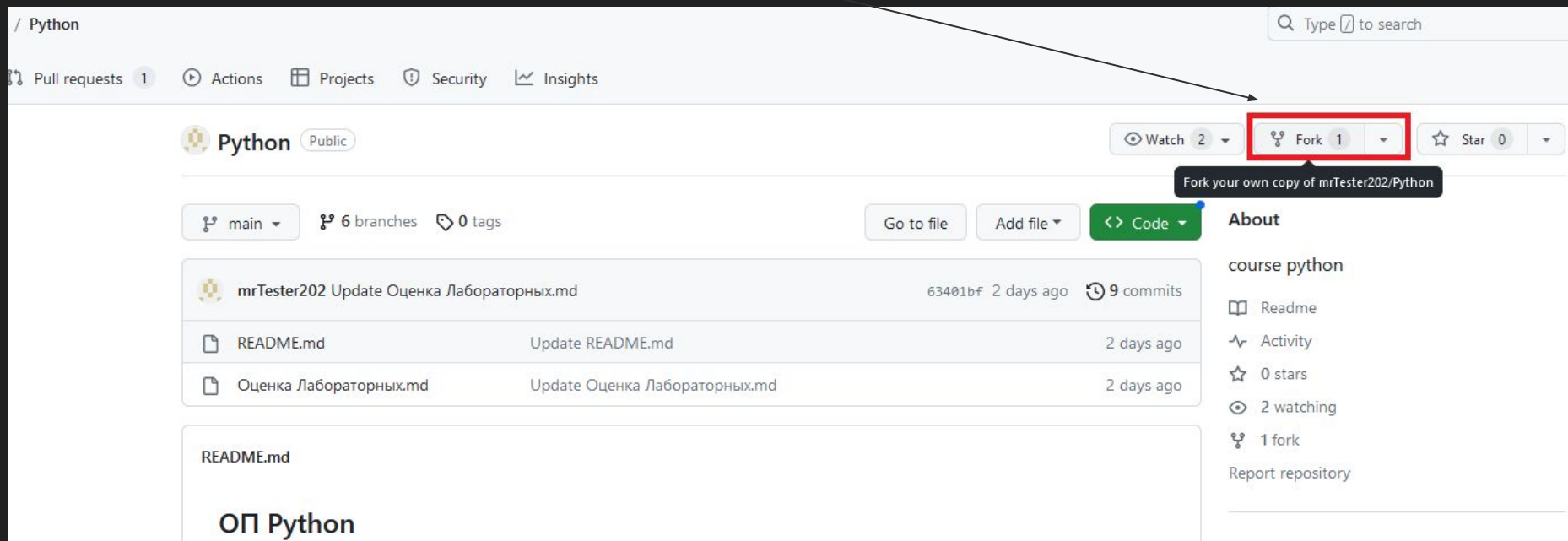
Пример веток репозитория

ветки репозитория

The screenshot displays the GitHub interface for a repository named 'Python'. At the top, navigation links include 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The repository header shows 'Python' as a public repository with options to 'Pin', 'Unwatch' (1), 'Fork' (0), and 'Star' (0). Below this, three yellow banners indicate recent pushes for 'section10' (6 minutes ago), 'section11' (5 minutes ago), and 'section12' (5 minutes ago), each with a 'Compare & pull request' button. A 'Switch branches/tags' dropdown menu is open, showing a list of branches: 'main' (selected, default), 'exam', 'section0', 'section1', 'section2', 'section3', 'section4', 'section5', 'section6_stdata', and 'section6'. The main content area shows the commit history for the 'main' branch, with the latest commit '2c919ee' from 15 minutes ago containing 2 commits. The commit message is 'Add files via upload'. On the right sidebar, the 'About' section is empty, 'Releases' shows 'No releases published', and 'Packages' shows 'No packages published'.

Форкнуть репозиторий на GitHub - это значит
сделать собственную копию чужого репозитория.

для того чтобы форкнуть репозиторий
нажмите на **Fork**



/ Python

Search: Type to search

Pull requests 1 Actions Projects Security Insights

Python Public

Watch 2 Fork 1 Star 0

Fork your own copy of mrTester202/Python

main 6 branches 0 tags

Go to file Add file <> Code

mrTester202 Update Оценка Лабораторных.md 63401bF 2 days ago 9 commits

README.md	Update README.md	2 days ago
Оценка Лабораторных.md	Update Оценка Лабораторных.md	2 days ago

README.md

ОП Python

About

course python

- Readme
- Activity
- 0 stars
- 2 watching
- 1 fork
- Report repository

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Owner *

tech

Repository name *

Python

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

course python

☐ Copy the `main` branch only

Contribute back to mrTester202/Python by adding your own branch. [Learn more.](#)

① You are creating a fork in your personal account.

Create fork

Уберите галочку здесь

Укажите имя репозитория (может быть любым)

нажмите - Create fork

Как добавлять файлы проекта?

Два основных способа добавления.

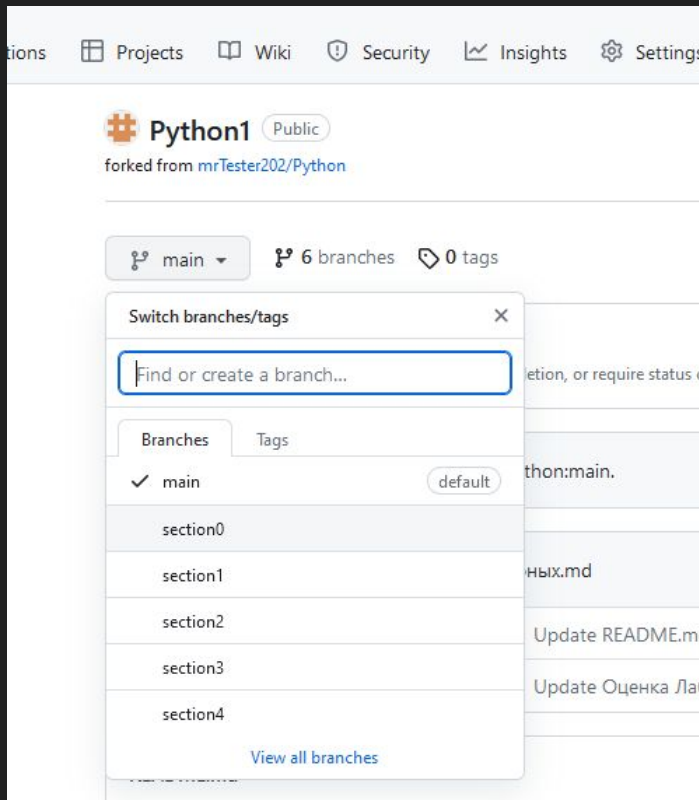
```
graph TD; A[Два основных способа добавления.] --> B[через IDE  
(среды разработки  
кода)]; A --> C[через Github  
(сайт или приложение)];
```

через IDE
(среды
разработки
кода)

через Github
(сайт или
приложение)

Добавление файлов средствами
Github (сайт) - самый простой и
понятный способ.

Переходите в ваш форкнутый репозиторий..



Выбор ветки. Например у меня section2 - содержит файлы к 1-й лабораторной работе (значит файлы лабы я буду загружать туда)

Добавление файлов..

Нажимаете **Add file** и выбираете **create new file** если хотите создать файл прямо в гитхабе или **Upload files** если хотите загрузить уже готовые файлы проекта.

The screenshot shows the GitHub interface for a repository named 'Python'. At the top, there are navigation links: Pull requests (1), Actions, Projects, Security, and Insights. Below this, the repository name 'Python' is displayed with a 'Public' badge. A search bar is visible in the top right corner with the placeholder text 'Type / to search'. The repository's main branch is 'main', and it has 6 branches and 0 tags. A red box highlights the 'Add file' button, which has opened a dropdown menu with two options: 'Create new file' (highlighted in blue) and 'Upload files'. Below the repository name, there is a table of recent commits. The first commit is by 'mrTester202' with the message 'Update Оценка Лабораторных.md'. The second commit is 'Update README.md'. The third commit is 'Update Оценка Лабораторных.md'. The right sidebar contains the 'About' section with the repository description 'course python', a 'Readme' link, 'Activity' link, '0 stars', '2 watching', '1 fork', and a 'Report repository' link.

Python Public

Go to file Add file > Code >

Create new file
Upload files

9 commits

mrTester202	Update Оценка Лабораторных.md	
README.md	Update README.md	2 days ago
Оценка Лабораторных.md	Update Оценка Лабораторных.md	2 days ago

README.md

About

course python

Readme

Activity

0 stars

2 watching

1 fork

Report repository

Upload (загрузка файлов проекта)..


Python1 /

выбираем файлы
на компьютере
здесь

Drag files here to add them to your repository

Or [choose your files](#)

Файлы не выбраны.

 Commit changes

первый комментарий |

commit (комментарий -
обязательно к заполнению)

Add an optional extended description...

☒ Commit directly to the `section2` branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

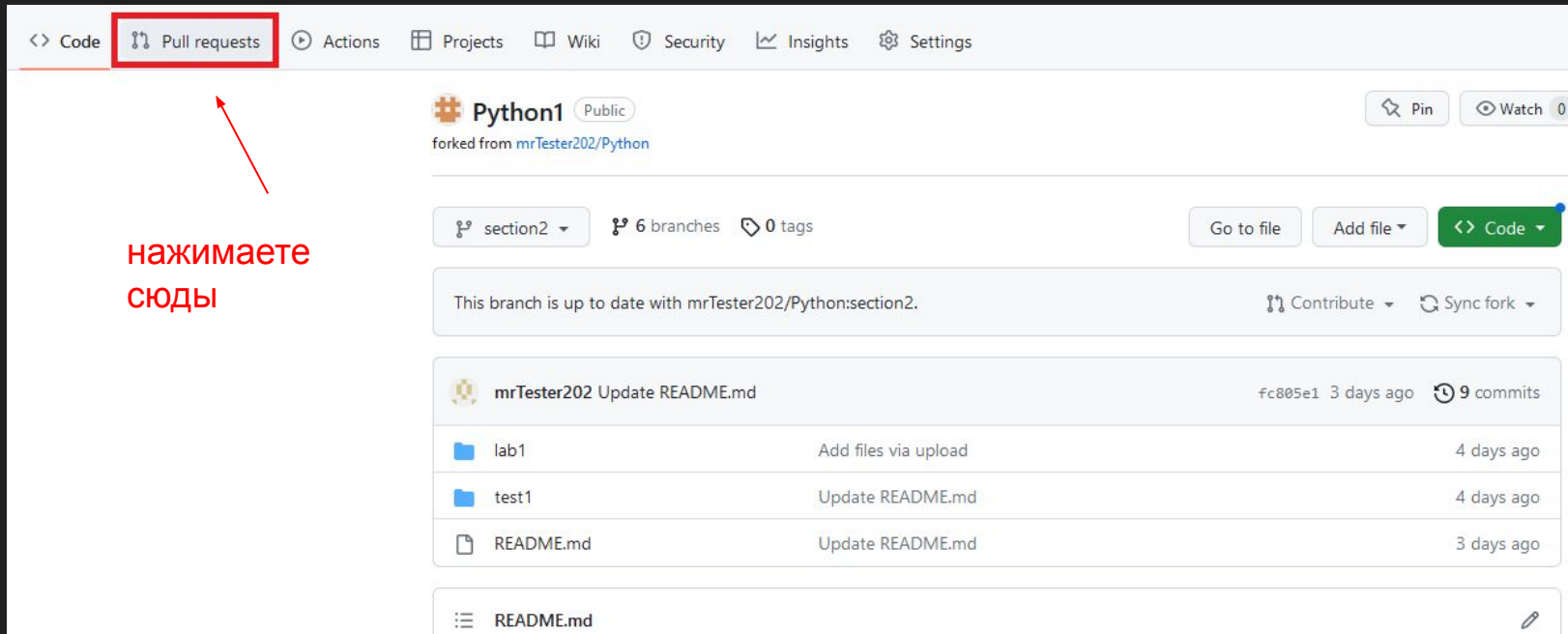
Commit changes Cancel

Что такое пулл-реквест?

Пулл-реквест - это предложение внести изменения в репозиторий на GitHub. Когда вы создаете пулл-реквест, вы предлагаете свои изменения к исходному коду проекта и просите владельца репозитория или других участников проекта просмотреть их, обсудить и, возможно, принять.

Как его правильно оформить?

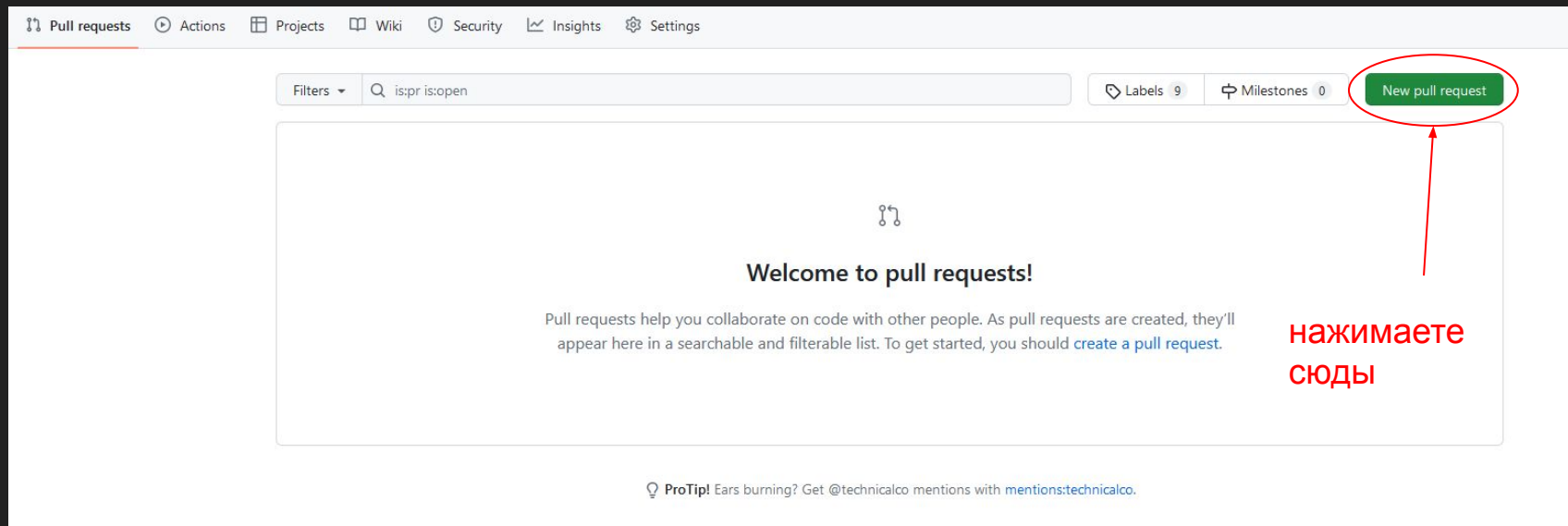
*После того как вы разумеется уже загрузили все файлы (сделали например полностью лабораторную работу)



The screenshot shows the GitHub interface for a repository named 'Python1', which is public and forked from 'mrTester202/Python'. The top navigation bar includes links for Code, Pull requests (highlighted with a red box), Actions, Projects, Wiki, Security, Insights, and Settings. A red arrow points from the text 'нажимаете сюда' to the 'Pull requests' link. Below the repository name, there are buttons for 'Go to file', 'Add file', and 'Code'. A status bar indicates 'This branch is up to date with mrTester202/Python:section2.' and provides options to 'Contribute' or 'Sync fork'. A list of recent commits is shown, including 'mrTester202 Update README.md' and several file updates like 'lab1', 'test1', and 'README.md'.

нажимаете
сюда

Далее..



Меняем ветку на ту в которой происходили изменения

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base repository: mrTester202/Python ▾ base: main ▾ ... head repository: technicalco/Python1 ▾ compare: section2 ▾

✗ Can't automatically merge. Don't worry, you can still create the pull request.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

8 commits

4 files changed

Commits on Jun 20, 2023

Update README.md

mrTester202 committed last week

Update README.md

mrTester202 committed last week

Choose a head ref

Find a branch

Branches

Tags

main

default

section0

section1

✓ section2

section3

section4

Create pull request

2 contributors

нажимаем
потом эту
кнопку

Verified



8ba1078



Verified



3f5268d



Л/Р №1, ФИО

← заголовок пулл-реквеста

Write

Preview

H B I ≡ <> 🔗 ≡ ≡ ≡ @ 🗨️ ↶ 🗑️

Что было в лабе...

Описание

нажимаем
потом эту
кнопку

Attach files by dragging & dropping, selecting or pasting them.

☒ Allow edits by maintainers ?

Create pull request

ИТОГИ#1

1. Управление версиями:

- Git позволяет отслеживать изменения в коде и документации, создавая историю версий. Это позволяет вернуться к предыдущим версиям, если что-то сломалось, и управлять изменениями.

2. Совместная работа:

- GitHub предоставляет платформу для совместной разработки. Множество разработчиков может работать над одним проектом, делать изменения и обсуждать их.

3. Хранение и обмен кодом:

- GitHub предоставляет удаленное хранилище для кода, которое можно легко обменивать между разработчиками. Это значительно упрощает процесс совместной разработки.

4. Отслеживание ошибок и запросов на улучшение (Issues и Pull Requests):

- GitHub позволяет создавать Issues для отслеживания ошибок и задач. С помощью Pull Requests можно предлагать изменения и улучшения кода, а также проводить их обзор.

ИТОГИ#2

5. Документация и вики-страницы:

- GitHub позволяет создавать и хранить документацию и вики-страницы проекта, что делает его более доступным и информативным для участников.

6. Интеграция с другими инструментами:

- GitHub интегрируется с множеством инструментов для автоматизации процессов разработки, непрерывной интеграции и развертывания.

7. Открытый исходный код:

- Множество проектов на GitHub являются открытыми и доступными для общественности. Это способствует обмену знаниями и сотрудничеству.