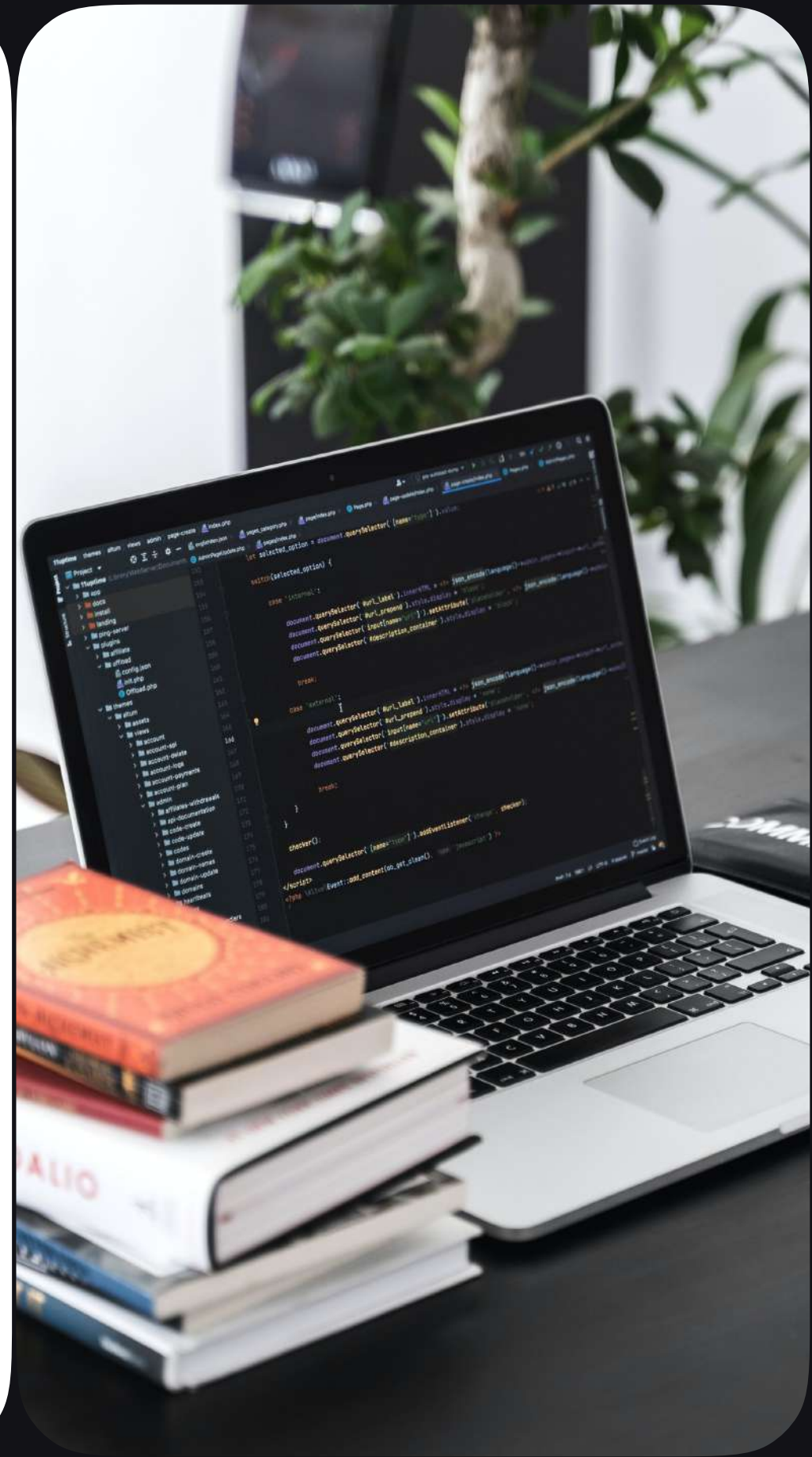


Модуль 3 Занятие 12

Система контроля версий. Git и Github



**Система
контроля
версий**

Git

Github

Практика

Система контроля версий



При разработке приложения мы сталкиваемся с ситуацией, когда необходимо иметь несколько версий одной программы — для этого используются системы контроля версий.

Система контроля версий (VCS, Version Control Systems) — это специальная программа, которая позволяет хранить несколько версий программы, при необходимости возвращаться к более ранним версиям, просматривать изменения, внесенные в код, и много другое. Система контроля версий также упрощает совместную разработку программы для нескольких разработчиков.

Git

Git — это одна из популярных систем контроля версий. Автором является Линус Торвальдс, создатель ядра Linux. Первая версия Git была выпущена в 2005 году для управления разработкой ядра Linux.

Git — это небольшая программа, которая работает через команды в терминале.



Репозиторий

Если вы начинаете отслеживать версии файлов вашего проекта с помощью Git, то папка с проектом становится репозиторием.



Репозиторий, в контексте систем контроля версий — это либо локальная папка на вашем компьютере, где хранятся версии проекта, либо хранилище на удаленном сервисе, например GitHub.



Репозиторий может быть локальным и удаленным.

GitHub

GitHub — сервис удаленного хранения репозиториев, основанный на системе контроля версий Git.



GitHub обладает огромным функционалом, сами создатели называют GitHub «социальной сетью для разработчиков».





Git и GitHub — не одно и то же

Git



Небольшая программа на компьютере



Система контроля версий



Позволяет работать с локальным репозиторием

vs

GitHub



Веб-сервис удаленного хранения репозиториев



Основан на системе контроля версий Git



«Социальная сеть для разработчиков»

Регистрация на GitHub

Зарегистрируйтесь
на **github.com**:



Укажите email

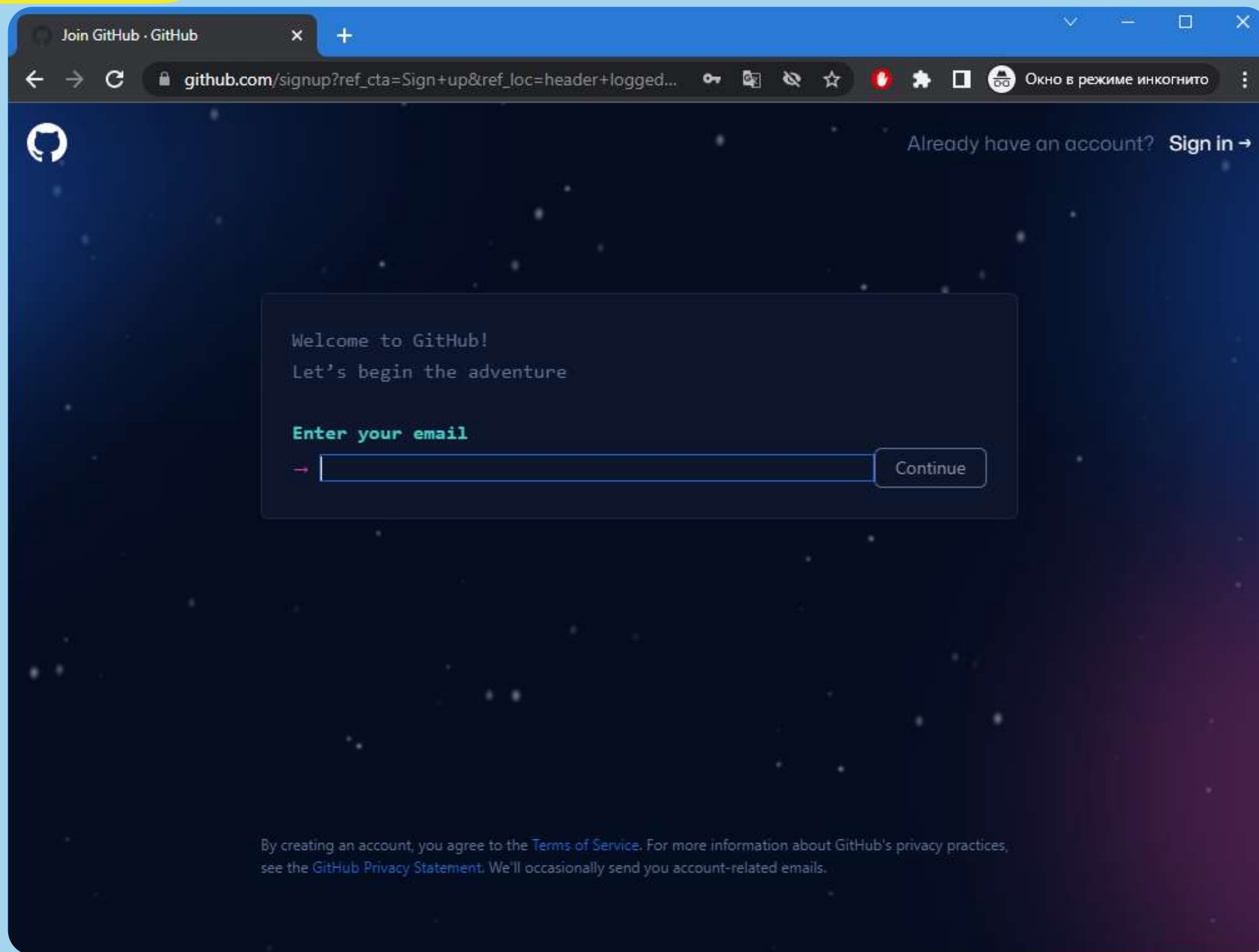


Пароль



Имя пользователя

После создания аккаунта
подтвердите адрес
электронной почты.



The screenshot shows the GitHub registration page in a web browser. The browser's address bar displays the URL `github.com/signup?ref_cta=Sign+up&ref_loc=header+logged...`. The page features the GitHub logo in the top left and a link to "Sign in" in the top right. The main content area has a dark background with the text "Welcome to GitHub! Let's begin the adventure". Below this, there is a prompt "Enter your email" followed by an input field and a "Continue" button. At the bottom of the page, there is a small disclaimer: "By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails."

Работа с GitHub

После регистрации и входа в аккаунт вы сможете создать удаленный репозиторий.

Работать с удаленным репозиторием можно разными способами:



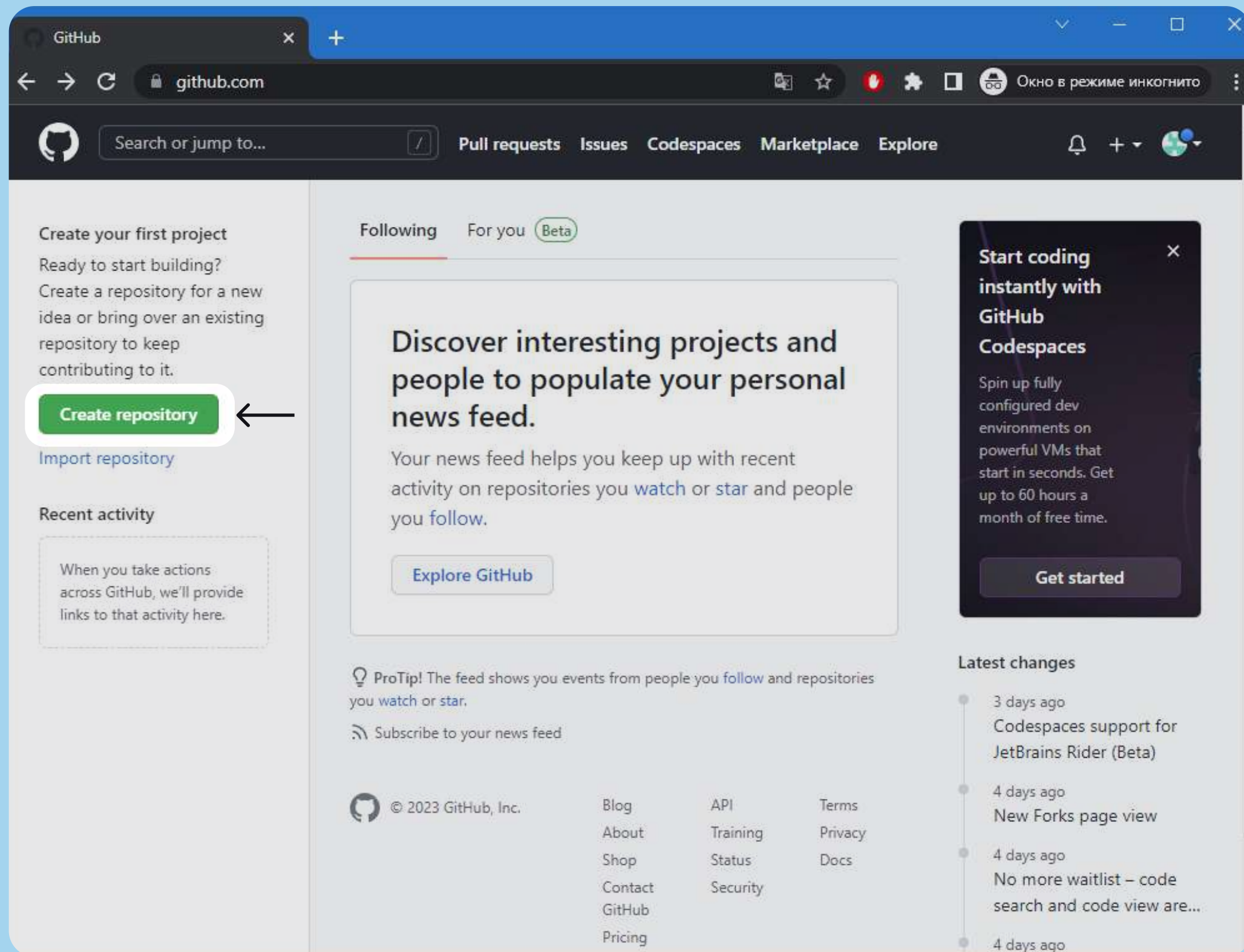
Использовать GitHub Desktop



Через терминал, используя команды Git



Через встроенные средства работы с VSC в средах разработки, например, PyCharm.



Создание репозитория

Создайте репозиторий, нажав на кнопку Create repository.

Для создания репозитория укажите его имя, выберите пункт Add a README file и добавьте .gitignore.

The screenshot shows the GitHub 'Create a new repository' page. The browser address bar shows 'github.com/new'. The page title is 'Create a new repository'. Below the title, it says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'.

The 'Owner' is set to 'podelev'. The 'Repository name' is '_test_repository', which is highlighted with a green checkmark and a black circle with the number '1' and an arrow pointing to it.

Below the name, it says 'Great repository names are short and memorable. Need inspiration? How about [fictional-doodle?](#)'.

The 'Description (optional)' field is empty.

Under 'Initialize this repository with:', there are two options: 'Public' (selected) and 'Private'. Below 'Public' is the text 'Anyone on the internet can see this repository. You choose who can commit.' Below 'Private' is the text 'You choose who can see and commit to this repository.'

Under 'Initialize this repository with:', there is a section 'Skip this step if you're importing an existing repository.' Below this, there are two checkboxes: 'Add a README file' and 'Add .gitignore'. The 'Add a README file' checkbox is highlighted with a white box and a black circle with the number '2' and an arrow pointing to it. Below it is the text 'This is where you can write a long description for your project. [Learn more.](#)'. The 'Add .gitignore' checkbox is highlighted with a white box and a black circle with the number '3' and an arrow pointing to it. Below it is the text 'Choose which files not to track from a list of templates. [Learn more.](#)'.

Below the checkboxes, there is a dropdown menu for '.gitignore template:' with 'Python' selected.

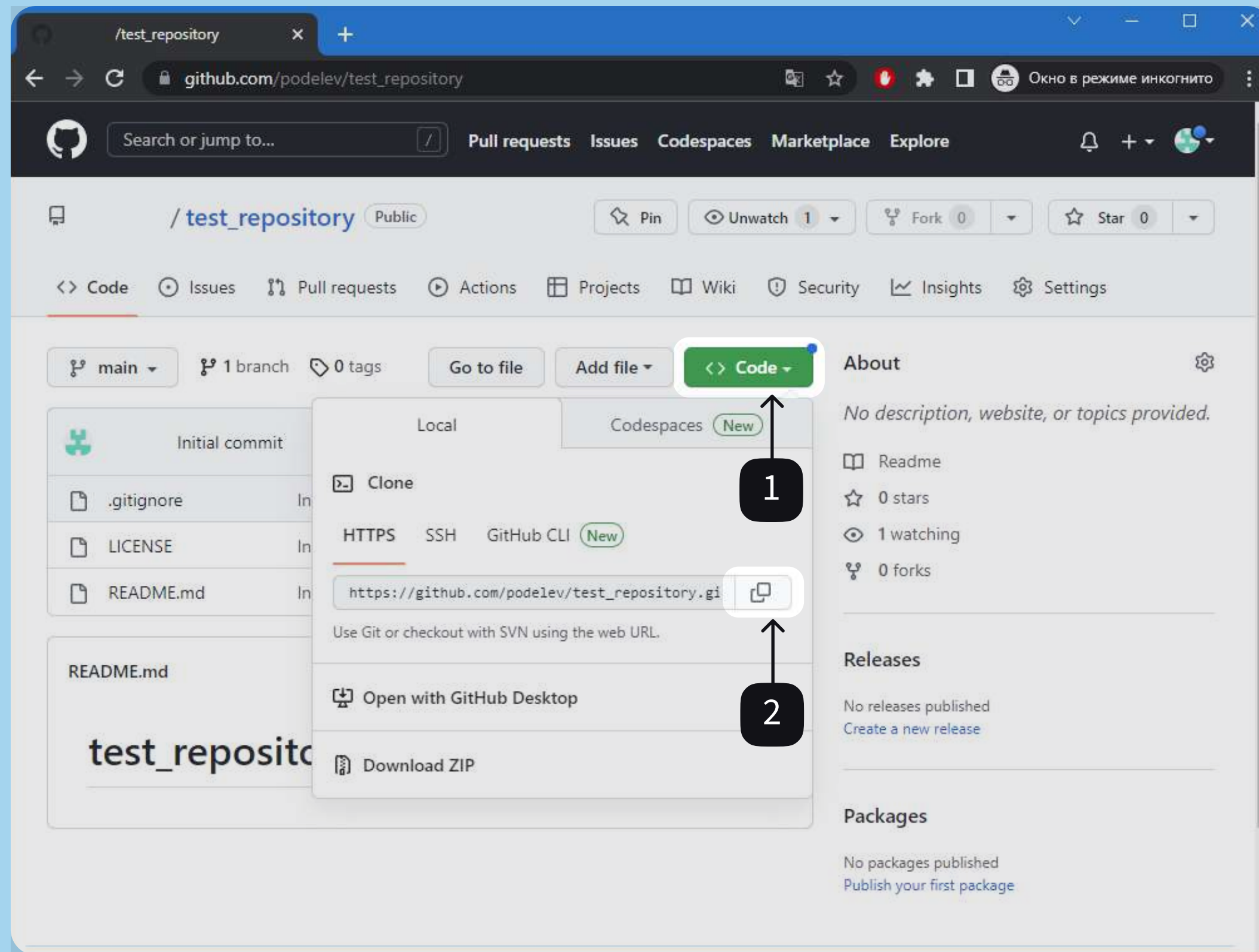
At the bottom, there is a section 'Choose a license'.

Ссылка на репозиторий

Клонирование репозитория — это создание его копии на компьютере.



Скопируйте ссылку на репозиторий, чтобы клонировать его на свой компьютер.



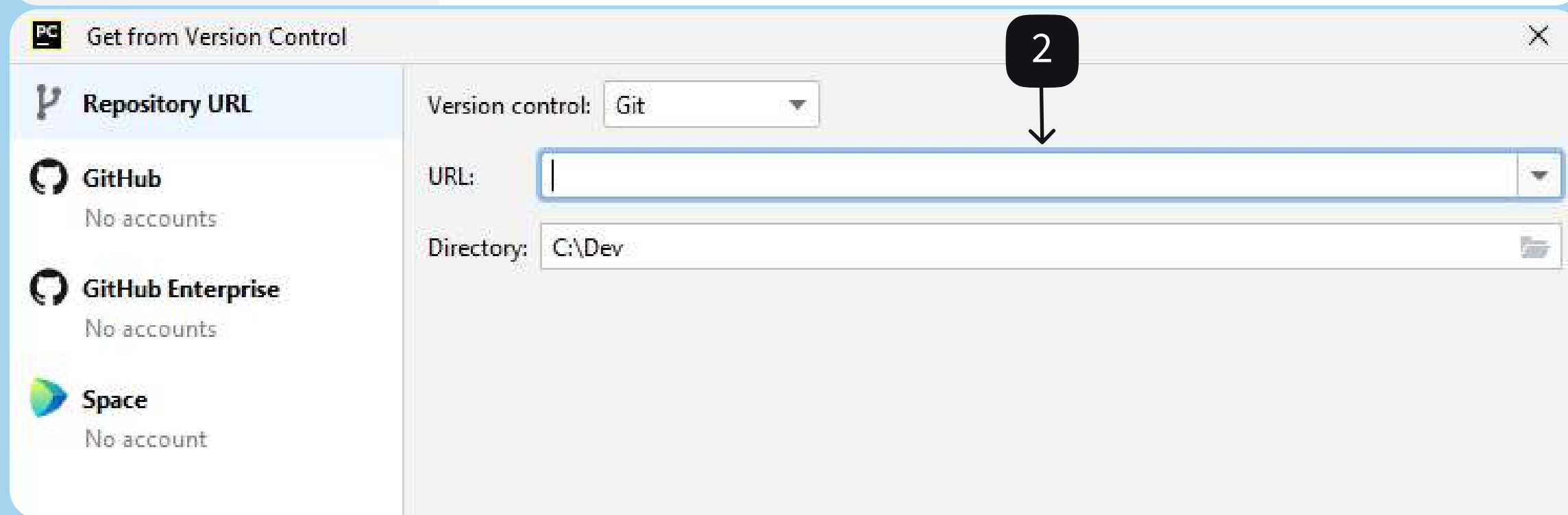
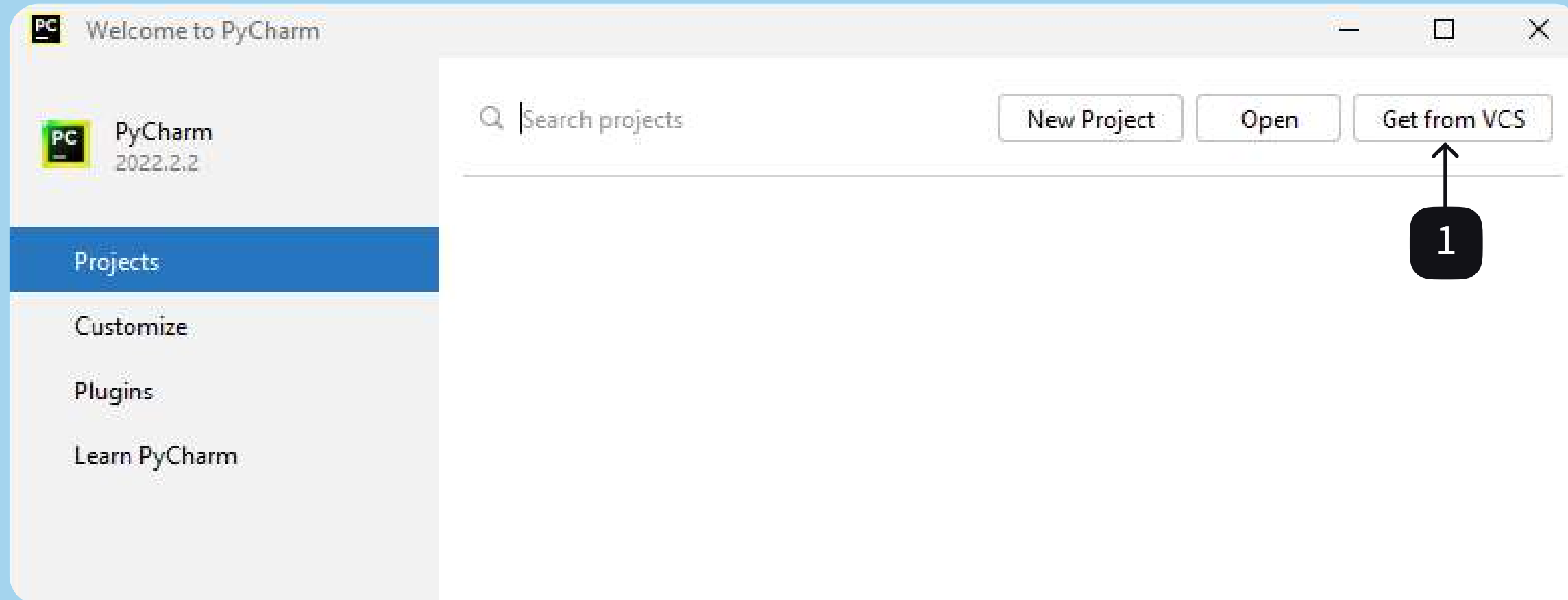
Clone

Далее работаем
в интерфейсе PyCharm:

1 Нажимаем кнопку **Get from VCS**

2 В открывшемся окне
вставляем ссылку
на скопированный
репозиторий.

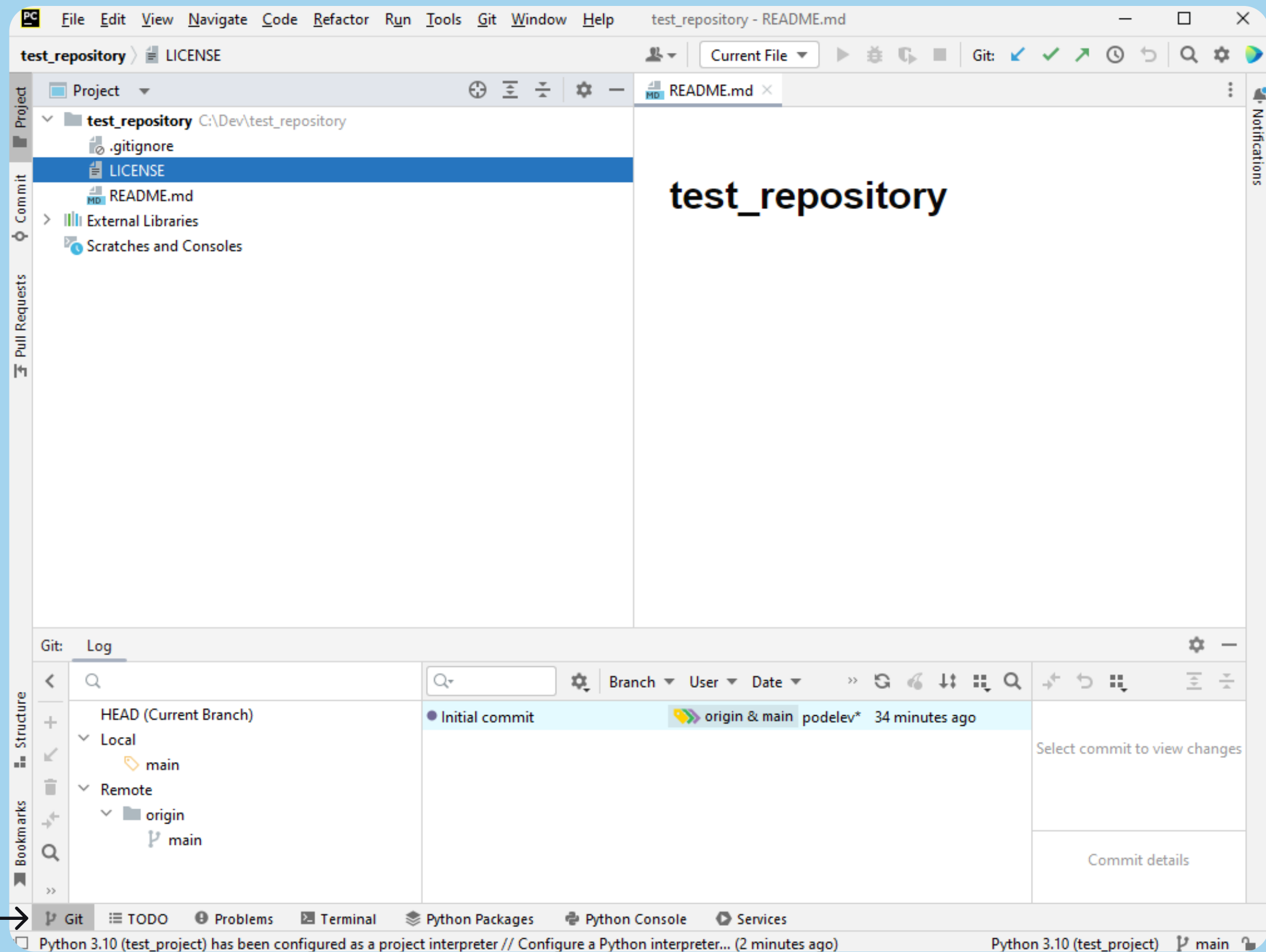
3 Нажимаем Clone.



Работа с Git в PyCharm

После того, как репозиторий будет клонирован, откройте панель **Git** и ознакомьтесь с ее содержимым.

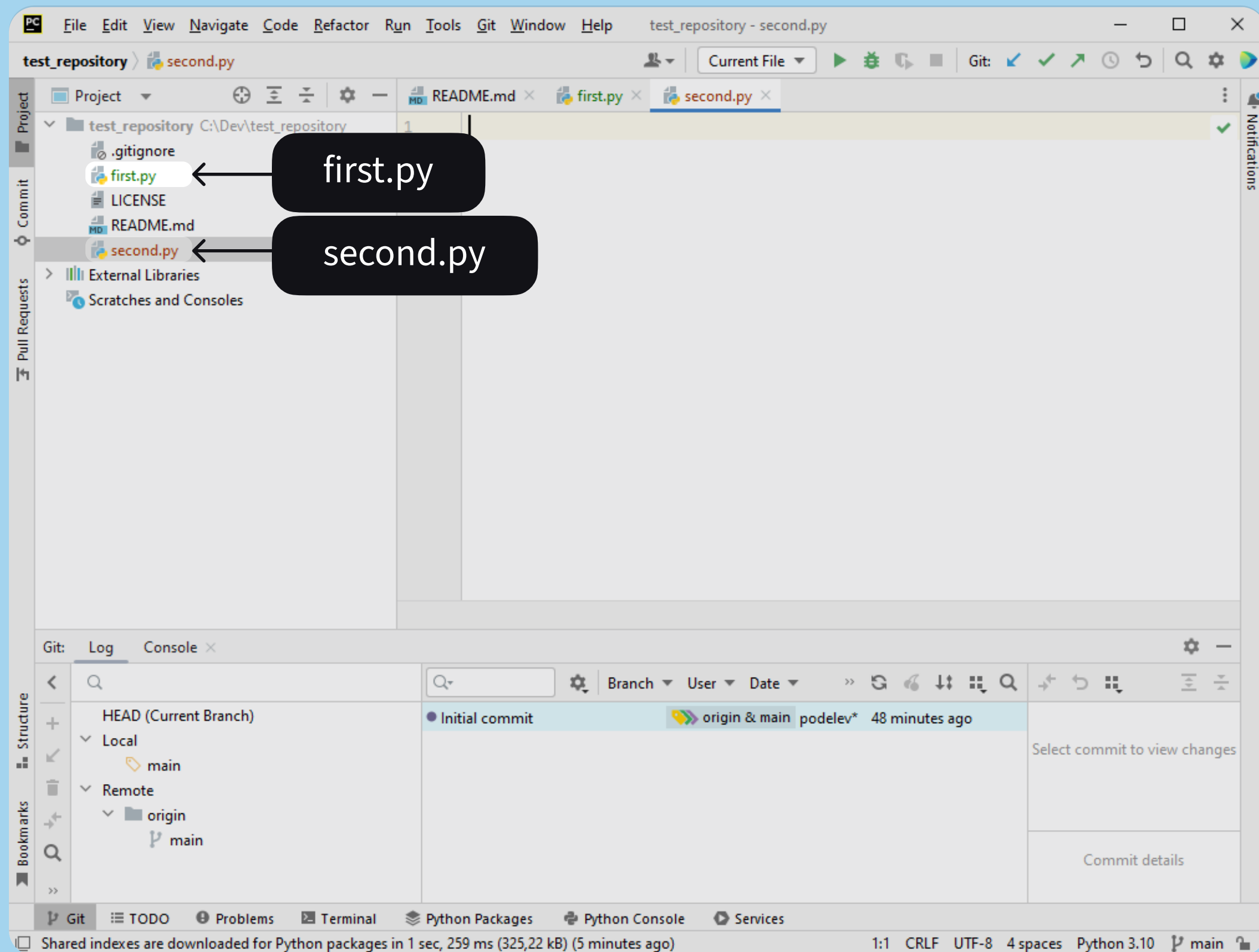
Git



Отслеживание файлов

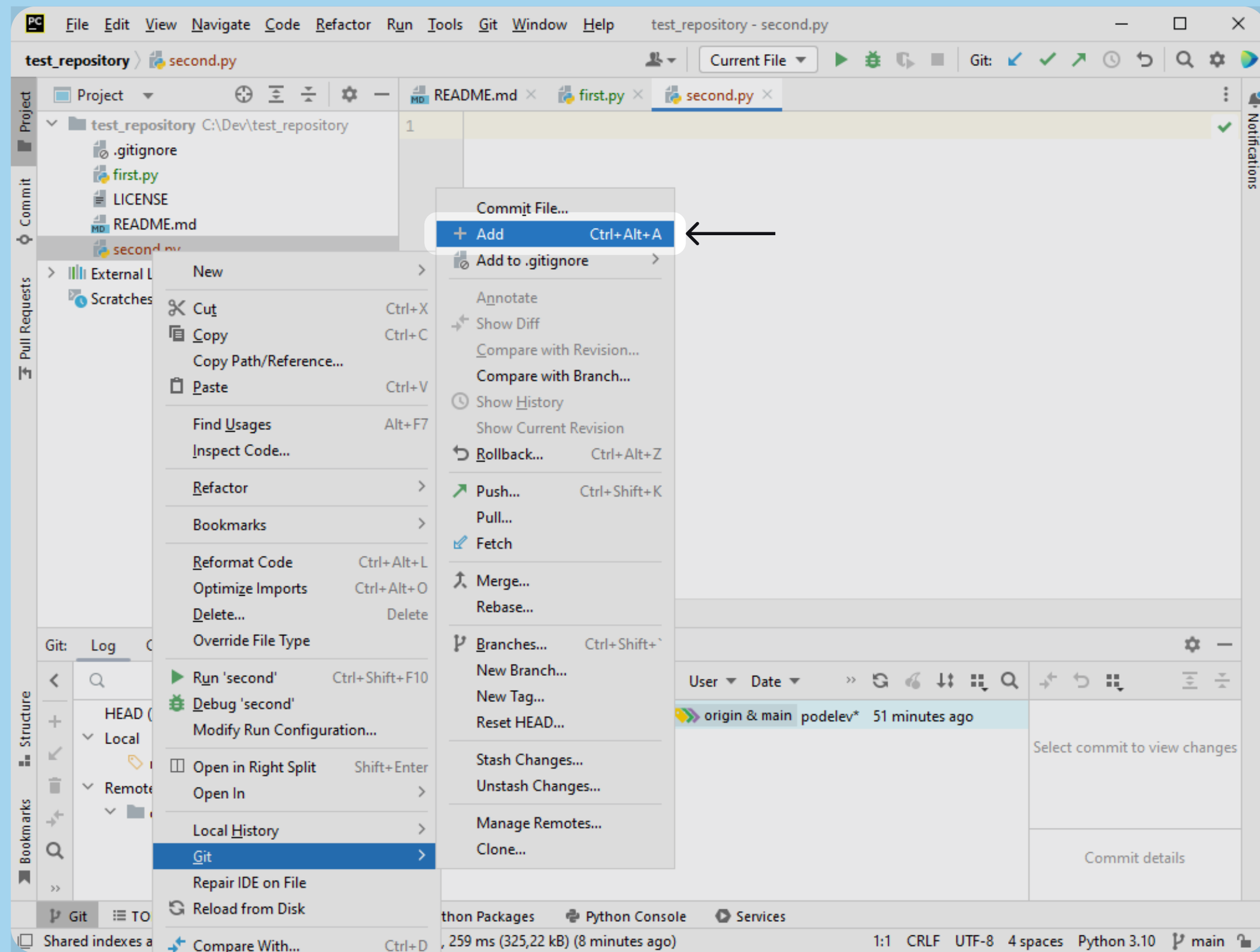
Добавим файлы **first.py** и **second.py**. При создании файлов один файл добавьте в «отслеживаемые», а второй файл — нет.

Отслеживаемые файлы — это те файлы, о которых знает Git. Они могут быть также измененными и неизмененными.



Добавление в отслеживаемые

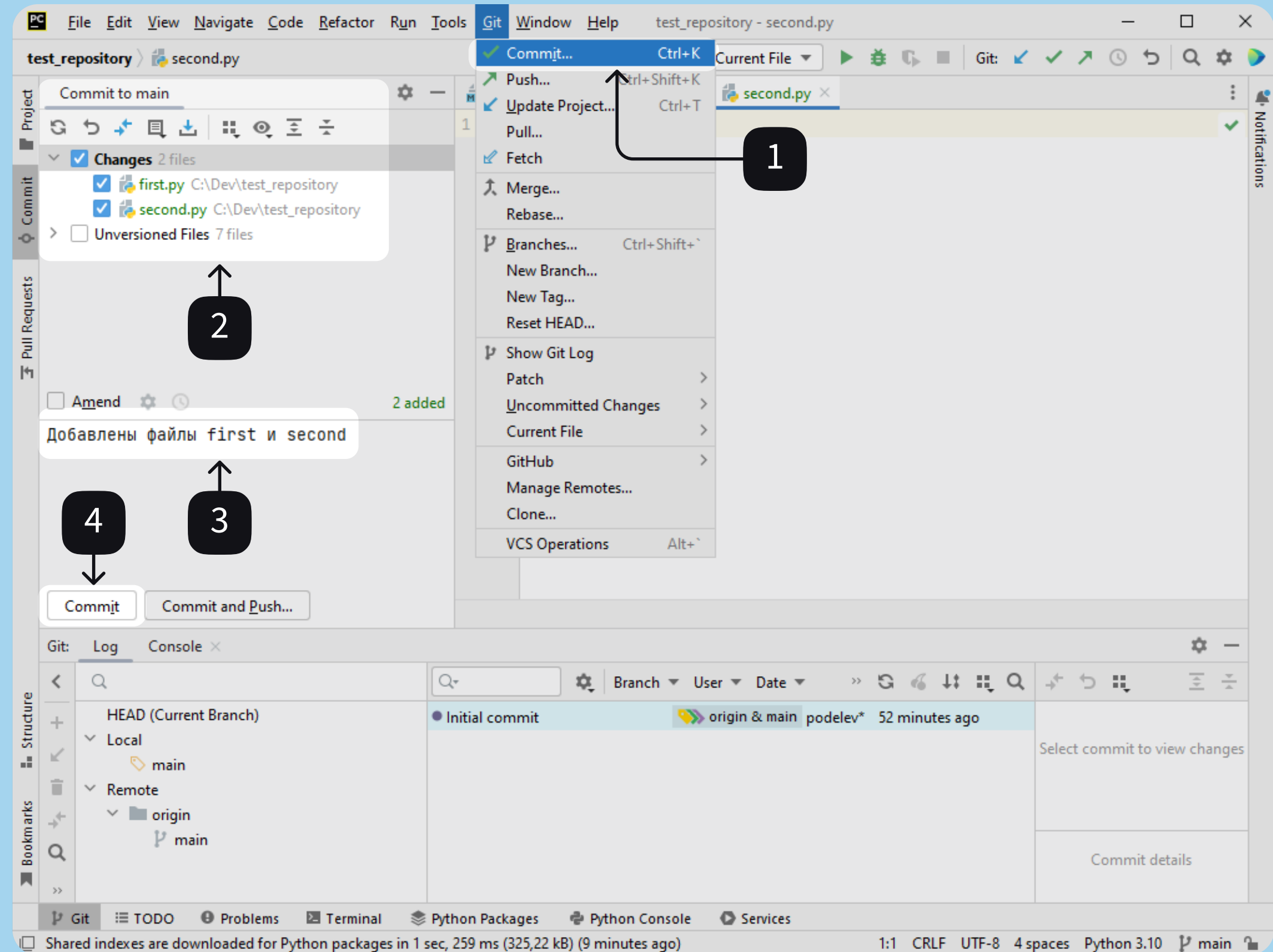
Если неотслеживаемый файл
надо добавить
в отслеживаемые, то можно
это сделать через меню **Git**,
нажав на файл правой
кнопкой мыши.



Commit

Commit (Коммит) — это фиксирование изменений в отслеживаемых файлах. Коммит сохраняет состояние файлов.

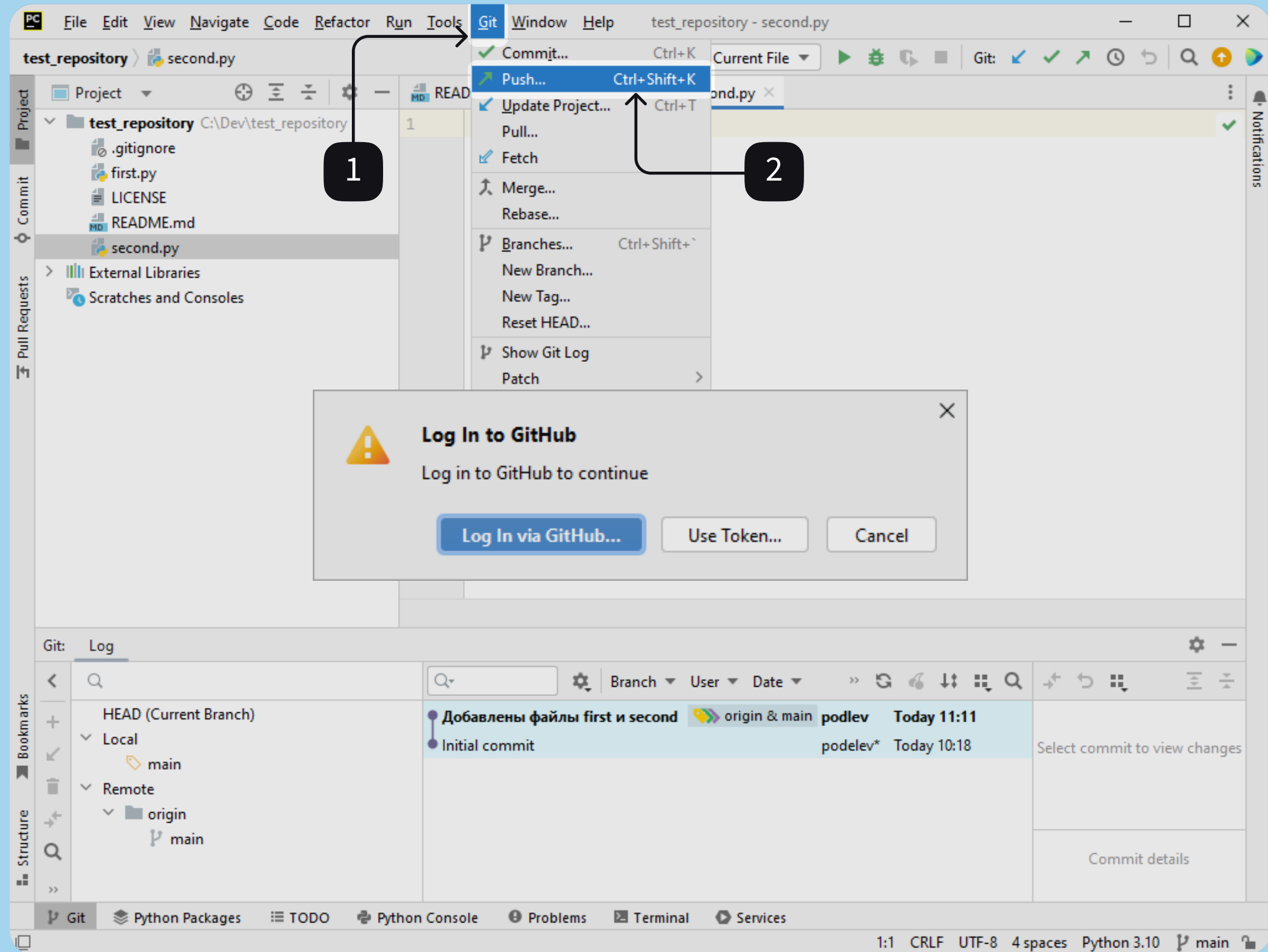
Для того, чтобы выполнить Commit, воспользуйтесь меню Git.



Push

Коммиты сохраняются локально, а для отправки локальных изменений в удаленный репозиторий **GitHub**, необходимо выполнить команду **Push**.

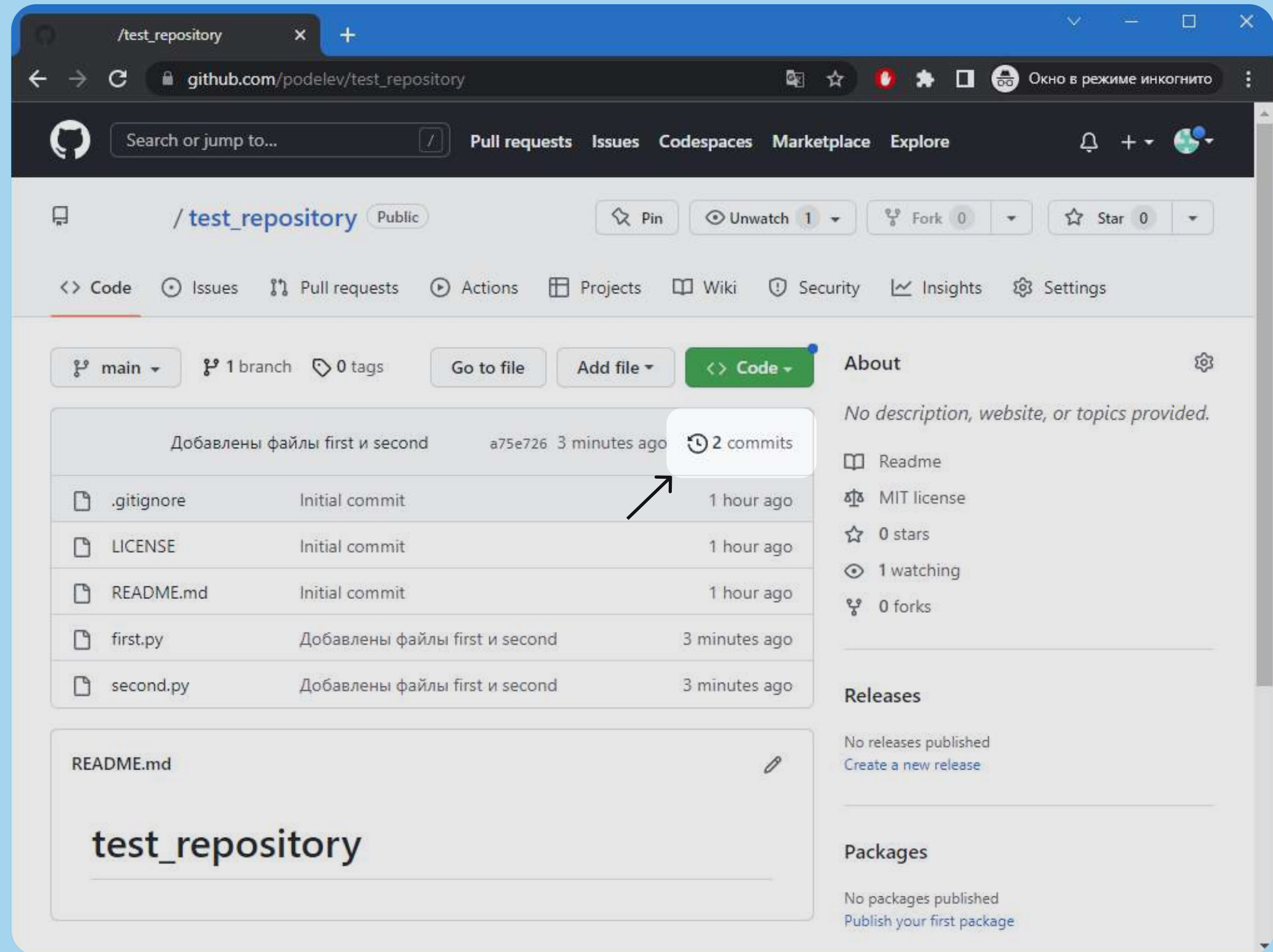
Для этого выполните вход в свой аккаунт GitHub.



Изменения зафиксированы

Перейдите в свой репозиторий **GitHub** и удостоверьтесь, что все изменения зафиксированы.

Найдите, где можно посмотреть все коммиты на GitHub.



Промежуточные итоги



Git — система контроля версий.



GitHub — сервис удаленного хранения репозиториев, основанный на системе контроля версий Git.



Репозиторий — это папка, в которой хранятся разные версии проекта. Может быть локальным и удаленным.



Файлы в репозитории могут быть отслеживаемые и неотслеживаемые.



Commit — это фиксация изменений в отслеживаемых файлах.

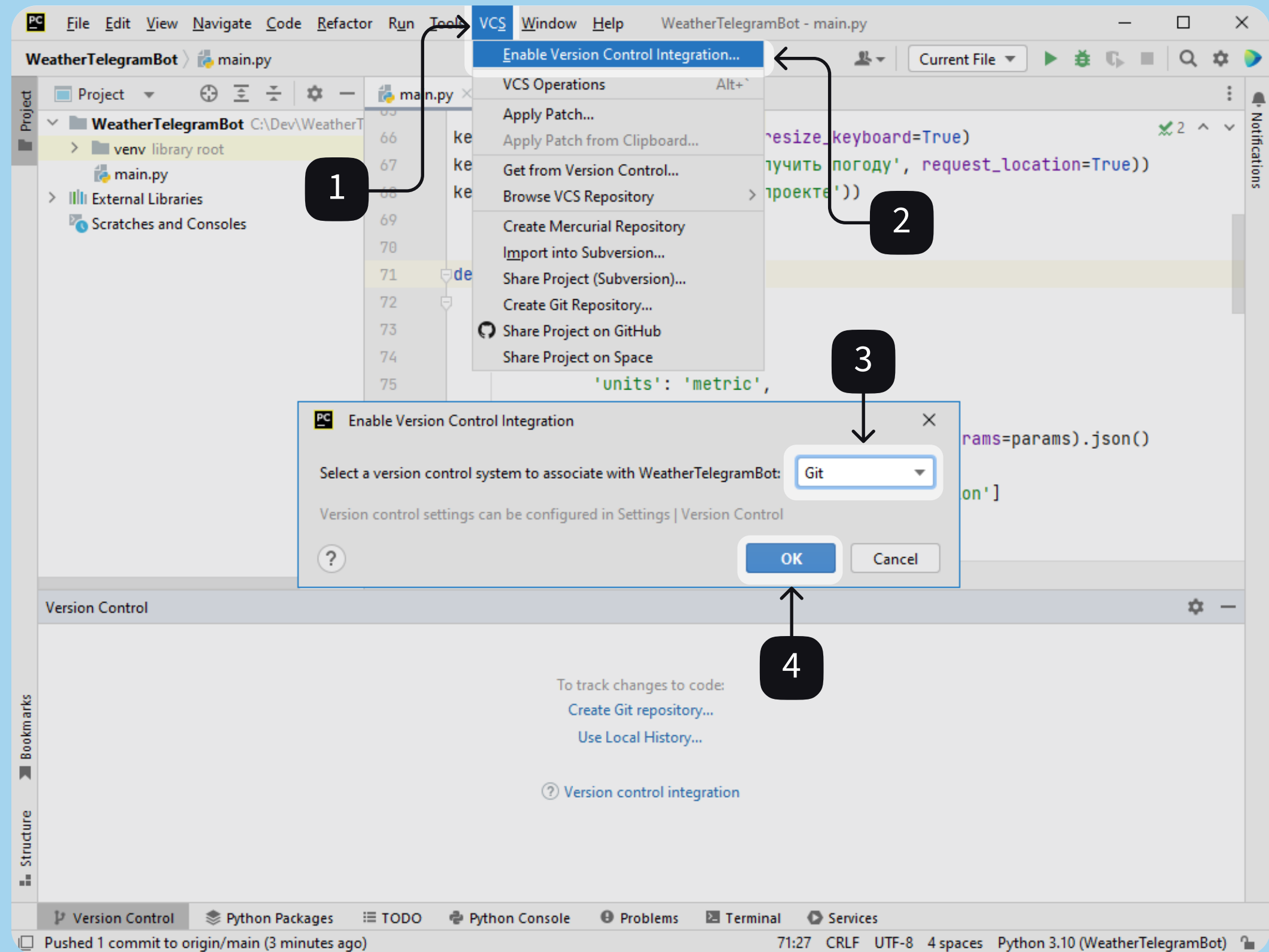


Push — отправка изменений в удаленный репозиторий.

Создание локального репозитория

Откройте существующий проект, который вы делали на прошлом занятии, и в меню VCS выберите пункт Enable Version Control Integration.

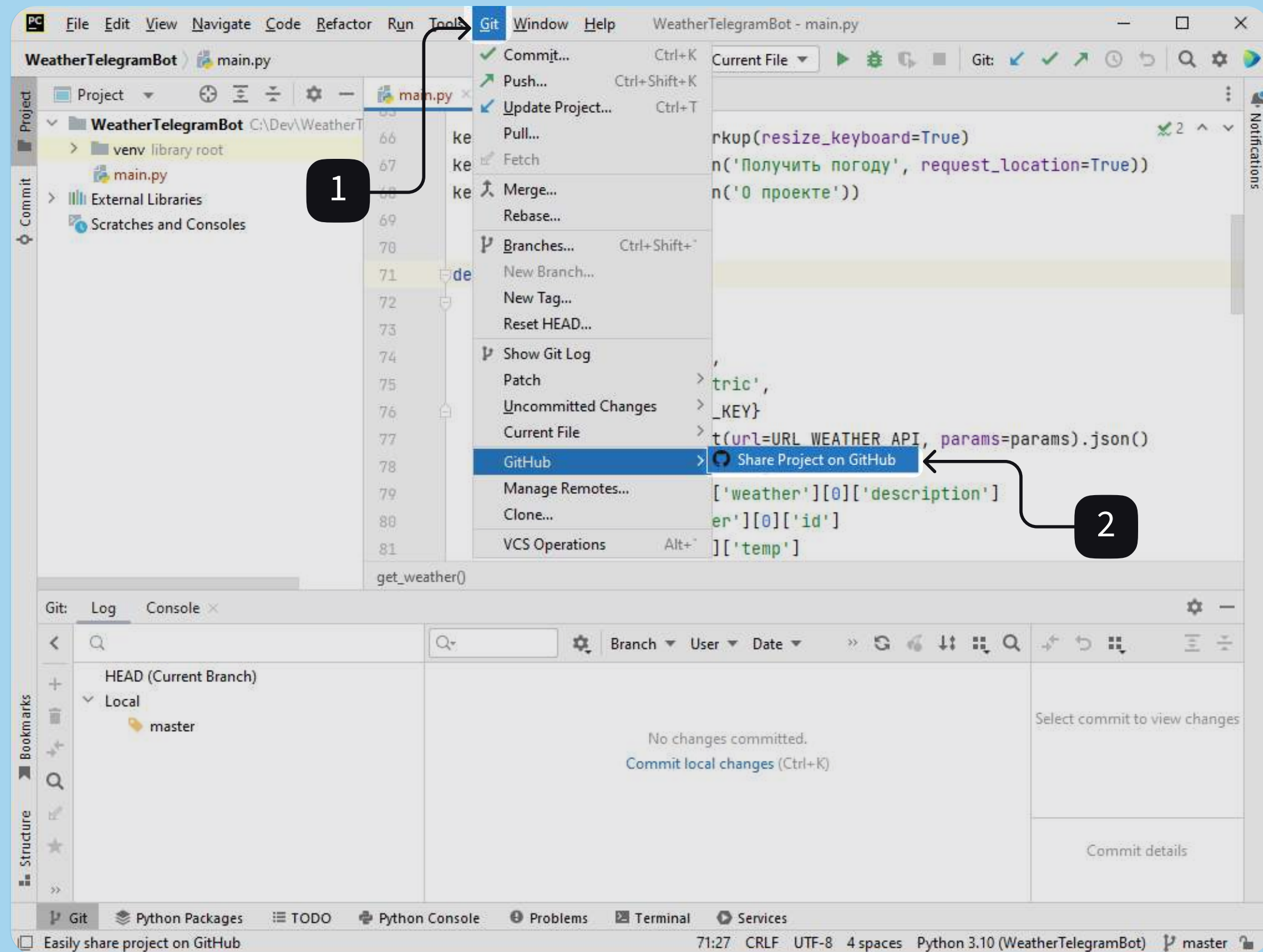
Выберите Git в качестве системы контроля версий и нажмите OK.



Отправка в удаленный репозиторий

После создания локального репозитория в панели меню изменится пункт меню с VCS на Git.

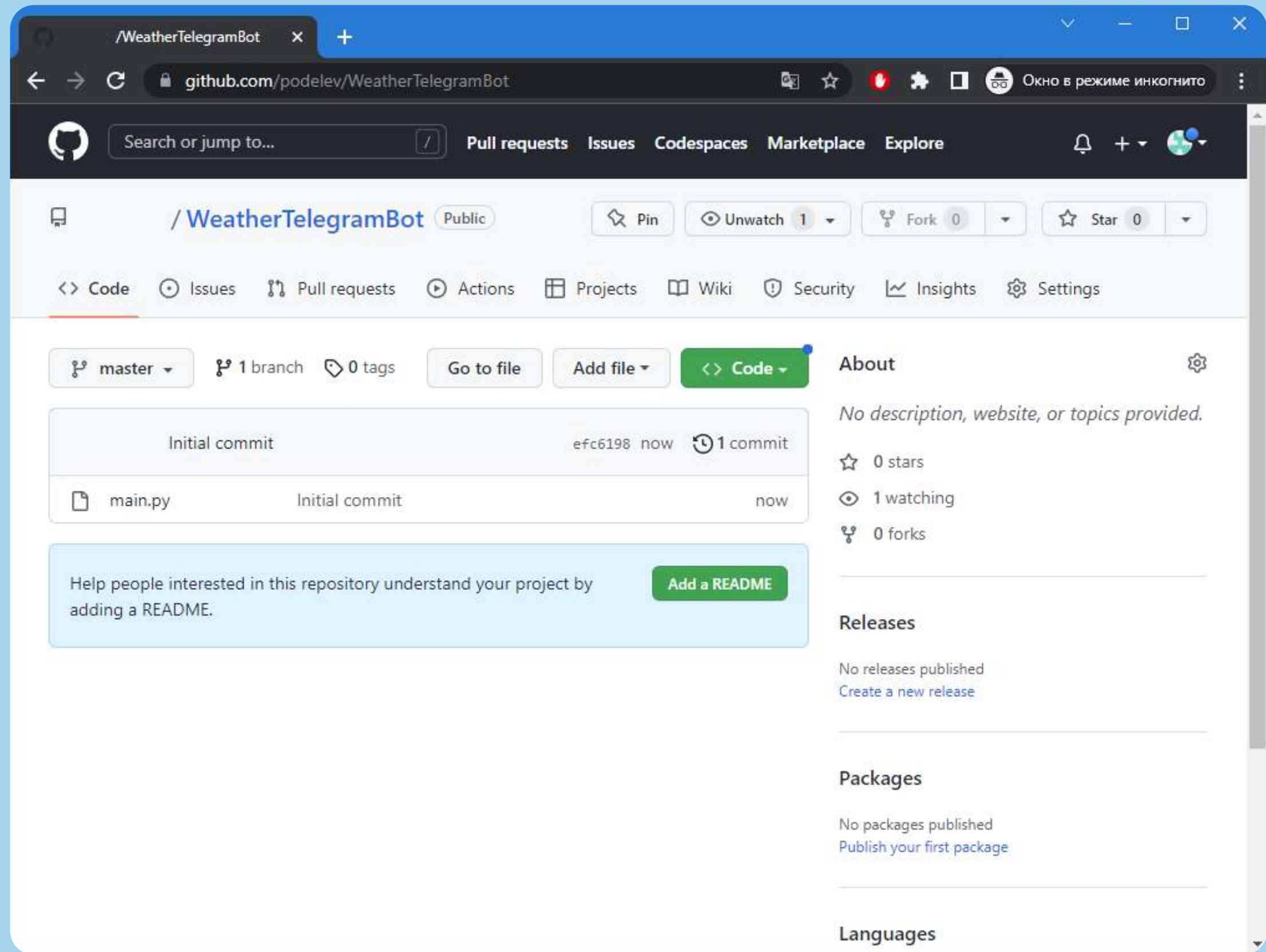
Чтобы отправить проект в удаленный репозиторий, выберите пункт Share Project on GitHub в меню Git и укажите имя удаленного репозитория.



Результат

Перейдите в свой профиль GitHub, найдите созданный репозиторий и удостоверьтесь, что все изменения зафиксированы.

Обратите внимание на то, что GitHub предлагает вам добавить файл README.



README.MD

README («прочти меня») — это файл, который содержит полную информацию о представленном проекте. README часто является первым объектом, который пользователь видит в репозитории. Файлы README обычно включают в себя следующие сведения:



Что делает проект и чем он полезен



Как пользователи могут приступить к работе с проектом



Где пользователи могут получить помощь по проекту



Кто поддерживает проект и вносит вклад в проект



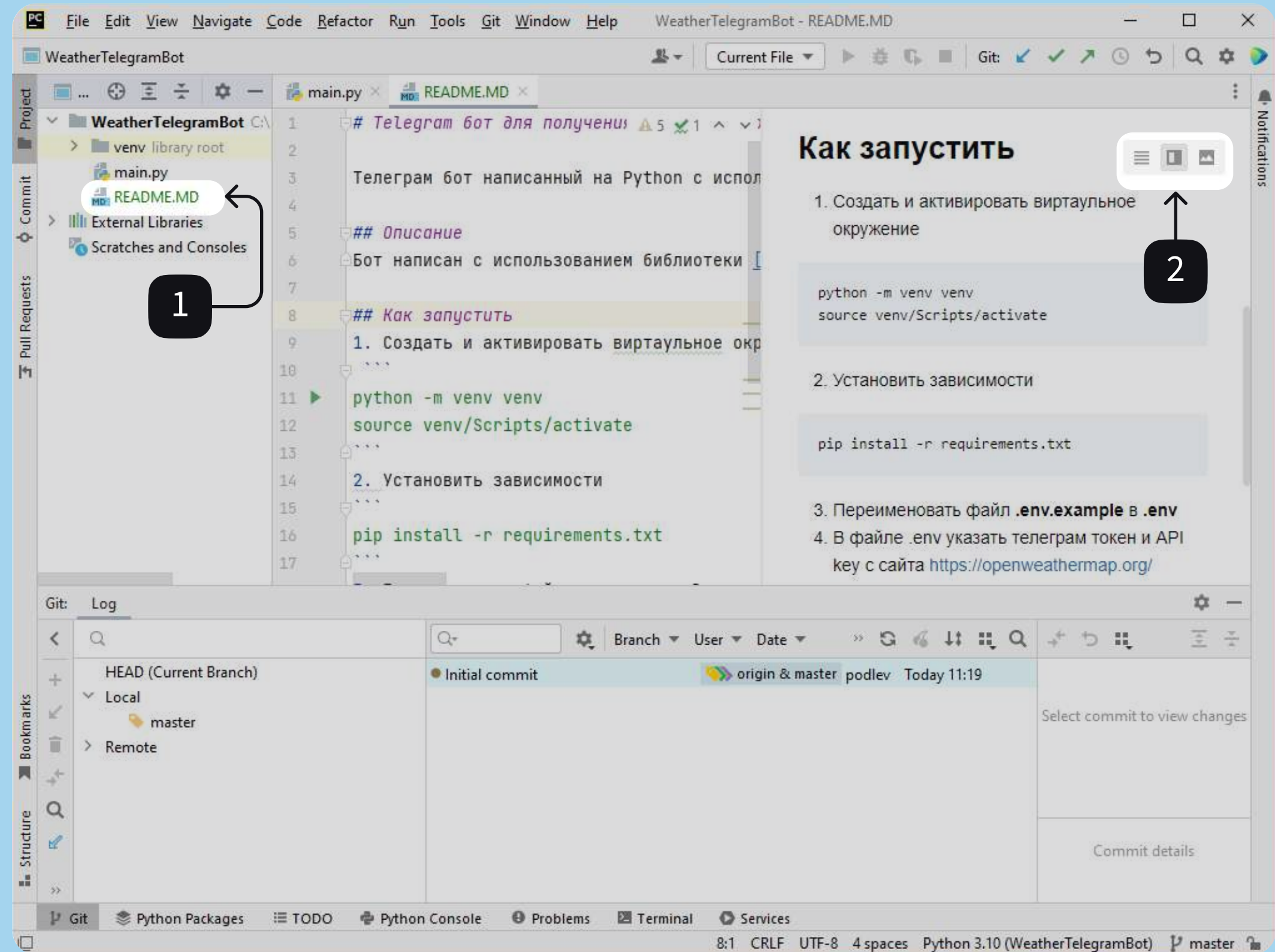
Расширение **.md** — это сокращение от слова **markdown**, это язык для форматирования текста. Этот язык используется для написания README для описания проектов на GitHub.

Как создать README

Создайте файл **README.MD** и добавьте в отслеживаемые, затем переключите разделение экрана на режим Editor and Preview.



Можно заполнять README



Pull

Для извлечения и загрузки содержимого из удаленного репозитория и обновления локального репозитория этим содержимым используется команда Pull.

