

Лабораторная работа 1 Hello GitHub

Средство контроля версий Git

Git — это система управления версиями с открытым исходным кодом. Она упрощает совместную работу над проектами с помощью *распределенной системы управления версиями* файлов (бывают ещё локальные и централизованные), которые хранятся в *репозиториях*.

GitHub — это служба размещения в Интернете репозиториях Git. Некая платформа для размещения кода для контроля версий и совместной работы над одним проектом из любого места. С помощью GitHub можно создавать, распространять и поддерживать программное обеспечение.

Цель:

познакомиться с одним из средств контроля версий через веб-сервис GitHub.

Задачи:

1. Завести аккаунт на GitHub (если еще нет)
2. Познакомиться с основными возможностями GitHub:
 - 2.1 Создать репозиторий
 - 2.2 Создать ветку
 - 2.3 Внести и сохранить изменения
 - 2.4 Слить изменения
3. Использование консоли для Git и для GitHub

1 Создание аккаунта

Тут вопросов возникнуть не должно. Заходим на <https://github.com/> и нажимаем “Sign up”, понадобится адрес электронной почты, придумать пароль и имя пользователя.

2 Основные возможности

2.1 Создание репозитория

Как правило в одном репозитории располагается один проект. Он может содержать папки и файлы, любые файлы, не только код, всё, что нужно для проекта.

Обычно в репозитории располагается README файл написанный с помощью синтаксиса **Markdown** (упрощенный язык разметки). В README содержится информация о проекте.

Для создания репозитория в верхнем правом углу на странице вашего аккаунта нужно нажать кнопку “+” и в всплывающем меню выбрать пункт “**New repository**” (“Новый репозиторий”).

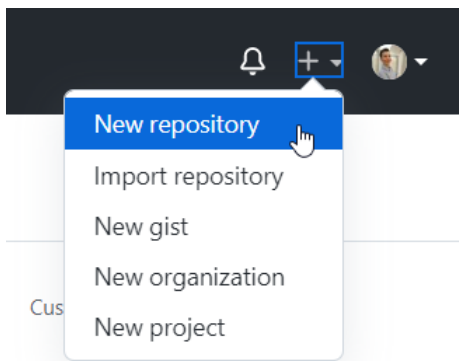


Рисунок 1. Создание репозитория

Нужно придумать имя репозитория, назовём наш учебный репозиторий “Information System Tools”.

Заполним поля:

Repository name = Information-Systems-Tools

Description = <напишите короткое описание проекта>

Нужно выбрать, что проект будет публичный, отмечаем “**Public**”.

Выбираем добавить файл README, “**Add a README file**”.

И нажимаем кнопку “**Create repository**” (“Создать репозиторий”).

Репозиторий готов, переходим к его наполнению.

2.2 Создание ветки

Создание ветки (branch) позволяет иметь различные версии репозитория в одно и то же время. Изначально в проекте одна ветка, которая называется main. Если требуется добавить какую-то функцию и при этом нельзя менять основной проект, то можно создать дополнительную ветку проекта для отладки этой функции. Можно производить различные изменения в дополнительной ветке и это никак не будет влиять на содержимое и работу главной ветки, пока мы не сольём (merge) дополнительную ветку с главной.

При создании ответвления main в новую ветку копируется текущее состояние main. Если кто-то другой обновил main, пока вы работали в своей ветке, то можно будет перенести (pull) эти обновления в вашу ветку.

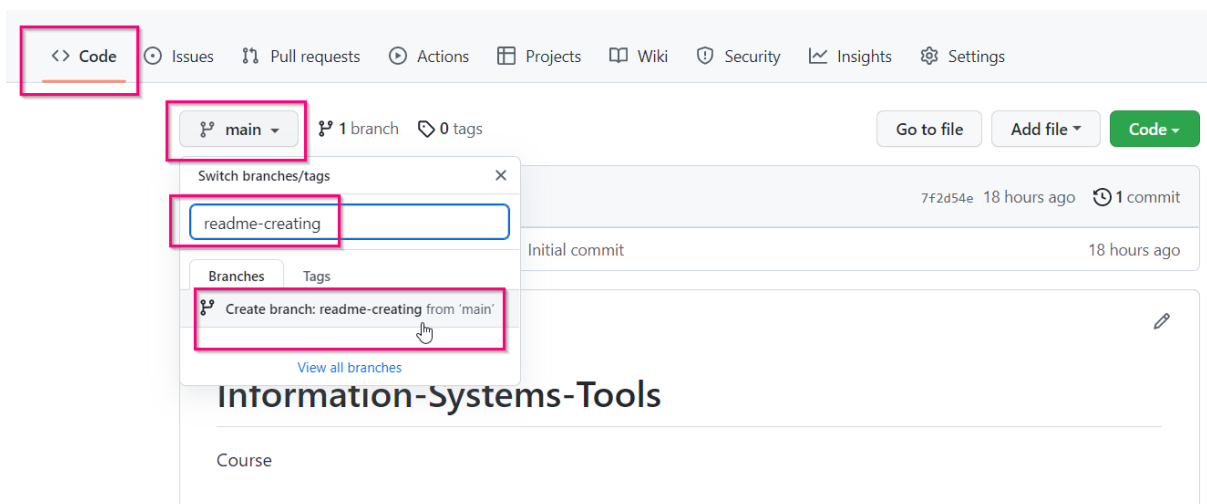


Рисунок 2. Создание ветки

Непосредственное создание новой ветки (смотрите рисунок 2):

1. Переходим на вкладку **Code**.
2. Открываем выпадающее меню или панель справа от **main**
3. Вводим имя новой ветки в поле **“Find or create a branch...”**, пусть будет **“readme-creating”**.
4. Нажимаем **“Create branch: <имя вашей ветки> from “main”**”.

Теперь имеем две ветки и они на данный момент идентичны.

2.3 Внести и сохранить изменения

Добавим изменение в нашу новую ветку **readme-creating**.

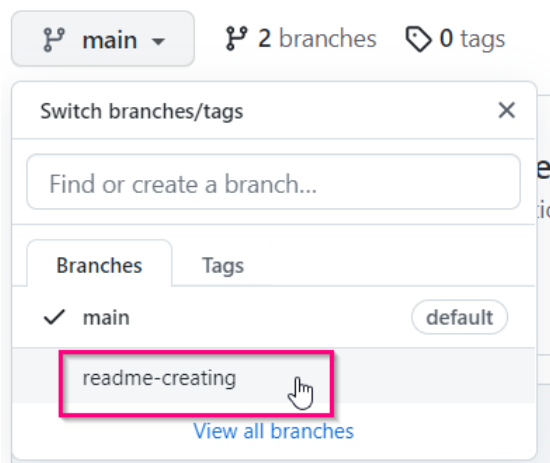


Рисунок 3 Переключение между ветками

Если выполняете работу последовательно, то на данном этапе новая ветка будет активна. Иначе необходимо перейти в новую ветку, это можно сделать через ниспадающую панель справа от **main**, как показано на рисунке 3.

В Git сохранение изменений или фиксация изменений называется так же как в базах данных – **commit**. Далее будем использовать этот термин.

Каждый коммит должен сопровождаться пояснением, почему было внесено данное изменение. Пишите подробно и по пунктам, что изменилось, чтобы другим и вам в будущем было понятно зачем были сделаны изменения. (Данная рекомендация касается не только Git).

Приступим:

1. Открываем **readme** файл, кликаем по нему.
2. Файл **readme** должен выглядеть как на рисунке 4, и чтобы начать его редактировать кликаем “**Edit this file**”.

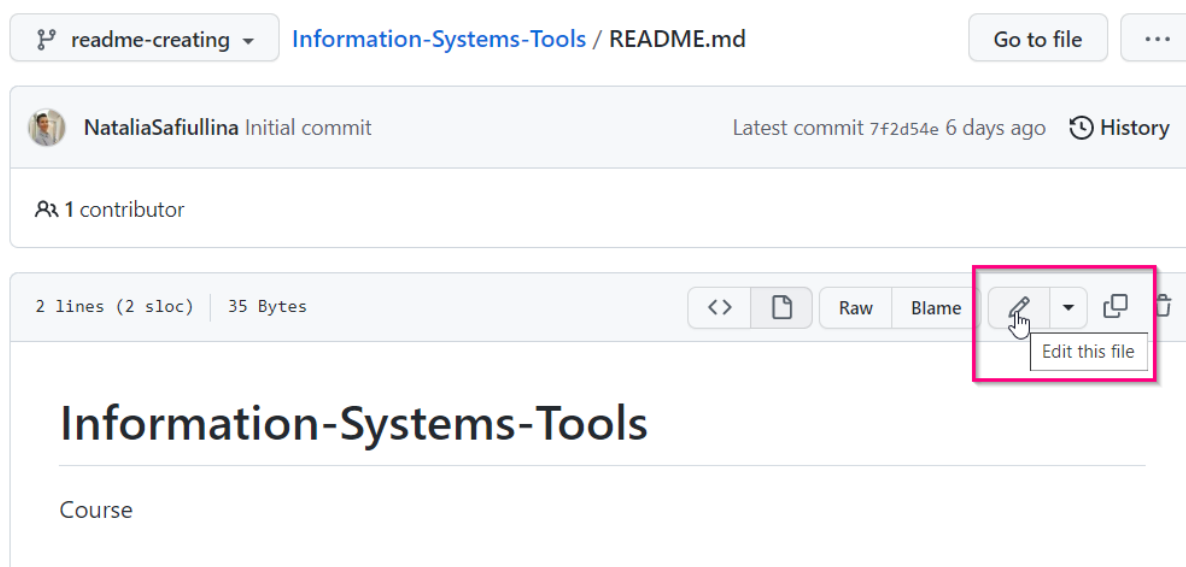


Рисунок 4 Запуск редактирования файла

3. Напишите немного о себе используя как можно больше синтаксиса **Markdown**. Ниже немного примеров синтаксиса:

Формат	Синтаксис
Заголовки различных уровней	# H1 ## H2 ### H3
Жирный шрифт	**bold text**
Курсив	<i>*italicized text*</i>
Цитата	> ...

Программный код	`...`
Ссылка	[текст который будет ссылкой] (https://www.example.com)
Картинка	! [подпись] (image.jpg)
Нумерованный список	1. ... 2. ... 3. ...
Ненумерованный список	- ... - ... - ...

Поищите в интернете остальные элементы синтаксиса, если они вам нужны.

Просмотреть предварительный результат редактирования можно перейдя во вкладку **Preview** (рисунок 5).

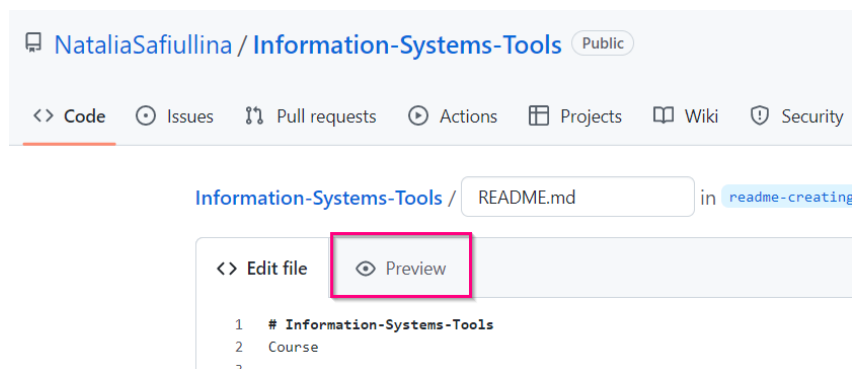


Рисунок 5 Предварительный просмотр README файла

- Ниже поля редактирования находится область **Commit changes** (рисунок 6). Перед сохранением изменений добавьте описание изменений.

Commit changes

Исправлены орфографические ошибки

Add an optional extended description...

☒ Commit directly to the `readme-creating` branch.
☐ Create a **new branch** for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes
Cancel

Рисунок 6 Область Commit changes

5. Сохраним изменения, кликаем кнопку **“Commit changes”**, изменения сохранятся в ветке **readme-creating**, теперь она отличается от **main**.

2.4 Слить изменения

Если изменения готовы и их можно слить с основной веткой проекта, то начинается **pull request**, т.е. запрос на изменение кода. Он показывает различия в содержании обеих веток.

Обычно слияние происходит при командной работе, т.е. изменения предлагаются на рассмотрение другим людям, но можно предложить и себе для эксперимента и обучения.

1. Переходим на вкладку **Pull Requests** (рисунок 7).
2. Нажимаем кнопку **New pull request**.

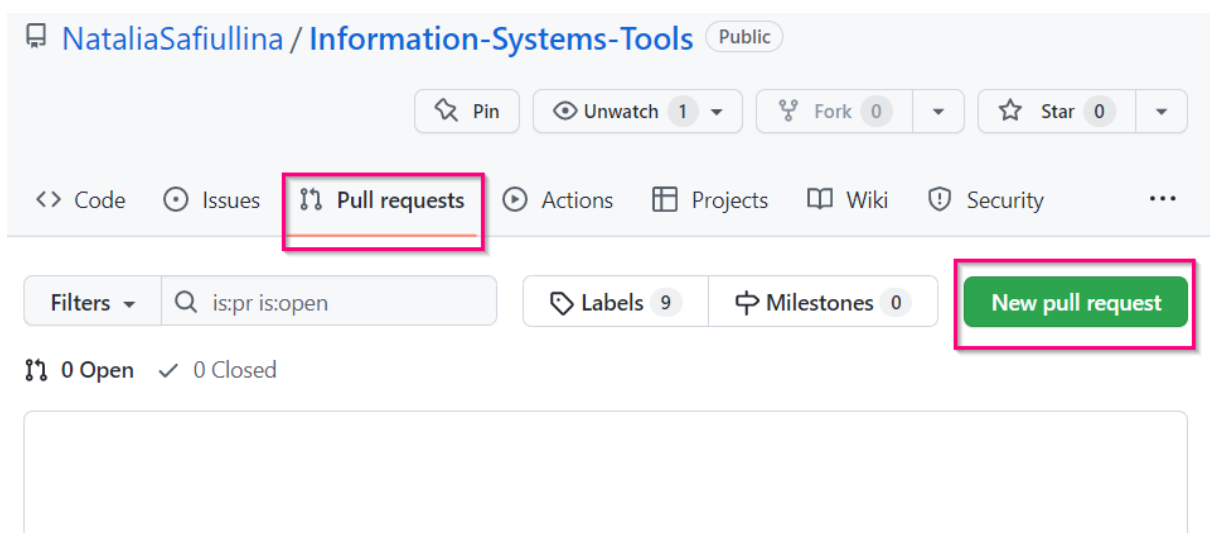


Рисунок 7 Создание pull request

3. В области **Example Comparisons** (рисунок 8) выбираем ветку **readme-creating** и в качестве ветки, с которой сравниваем, указана **main**.

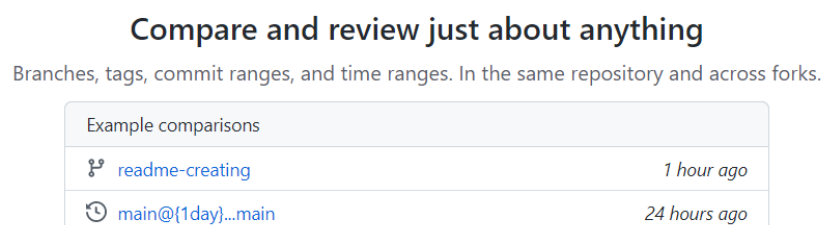


Рисунок 8 Выбор ветки для слияния

4. На следующем шаге видим что мы хотим предложить, проверяем не ошиблись ли мы с выбором изменений. Git информирует какие изменения есть, добавления, удаления и прочее.
5. Теперь создаём запрос на изменения, нажимаем **Create pull request**, необходимо придумать название и краткое описание (рисунок 9).

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: main ← compare: readme-creating

✓ Able to merge. These branches can be automatically merged.

Создание файла README

Write Preview

Н В I ≡ <> 🔗 ≡ ≡ ≡ @ ↗ ↶

Просто создание файла описания репозитория.
В котором содержатся несколько примеров разметки текста.

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

Create pull request

Remember, contributions to this repository should follow our [GitHub](#)

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development ⓘ
Use [Closing keywords](#) in the description to automatically close issues

Рисунок 9 Описание изменений

6. Кликаем **Create pull request**

Соавторы могут проверить, что вы предлагаете и вынести свои предложения. Переходим к последнему этапу непосредственному объединению веток.

Если в двух версиях будут конфликты, то GitHub предупредит об этом и не даст объединить ветки, пока конфликт не будет разрешён.

В наших изменениях не должно быть никаких конфликтов, поэтому:

1. Нажимаем **Merge pull request**.

2. Нажимаем **Confirm merge**.
3. Нажимаем **Delete branch**, чтобы удалить ветку **readme-creating**.

3 Консоль

(задание для использования консоли появится позже)

4 Контрольные вопросы

1. Для чего нужны инструменты контроля версий?
2. Что такое Git, что такое GitHub? Чем отличаются?
3. Что такое pull request?