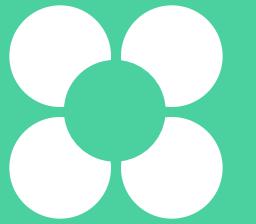


Командная работа в Git & GitHub. Часть 2

Алёна Батицкая

Фронтенд-разработчик, фрилансер



Алёна Батицкая

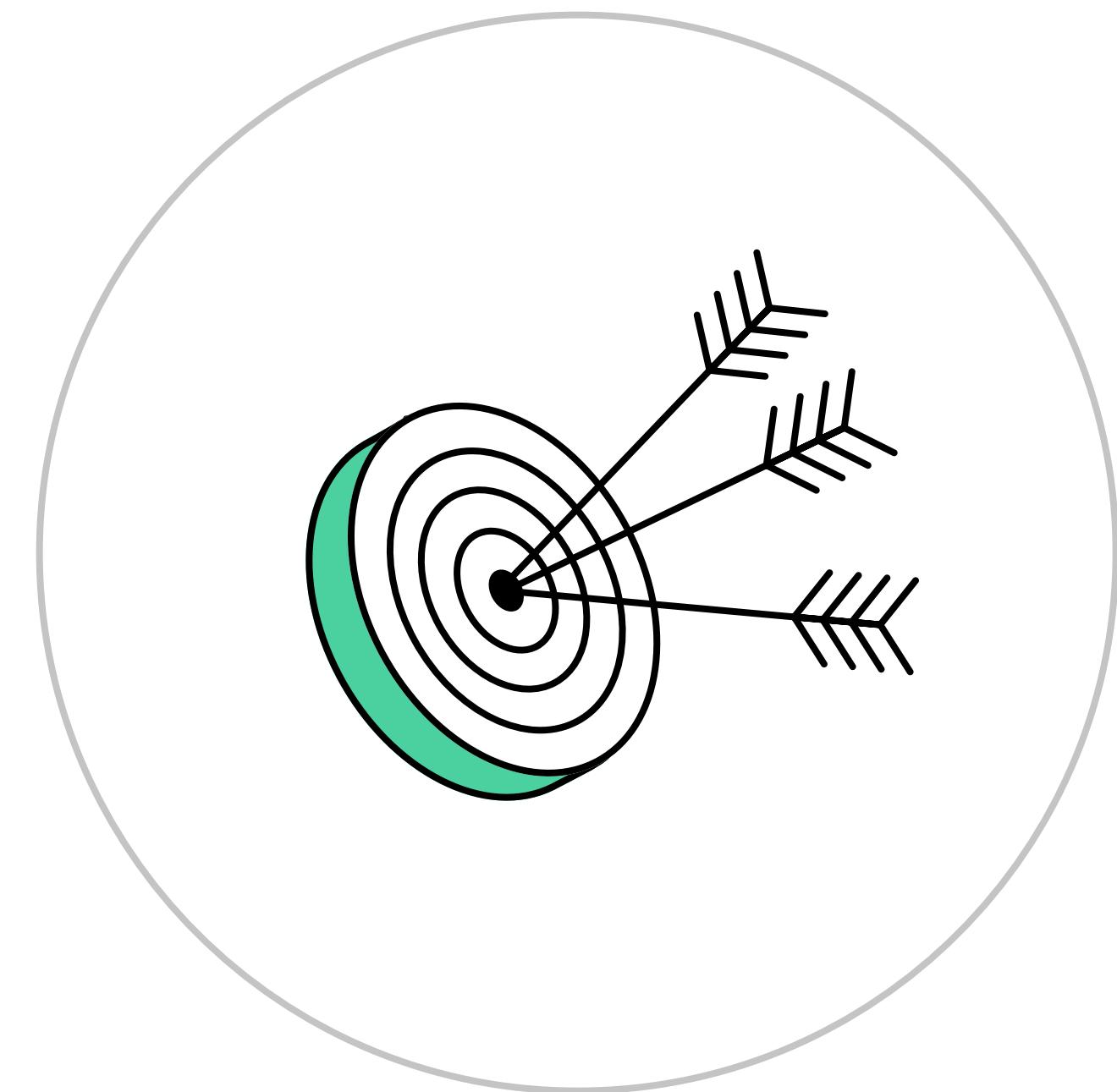
О спикере:

- Фронтенд-разработчик на фрилансе
- Автор и руководитель курсов в Нетологии
- Google Developer Expert for web
- Редактор проекта [«Дока»](#)
- Спикер, автор и переводчик статей по программированию



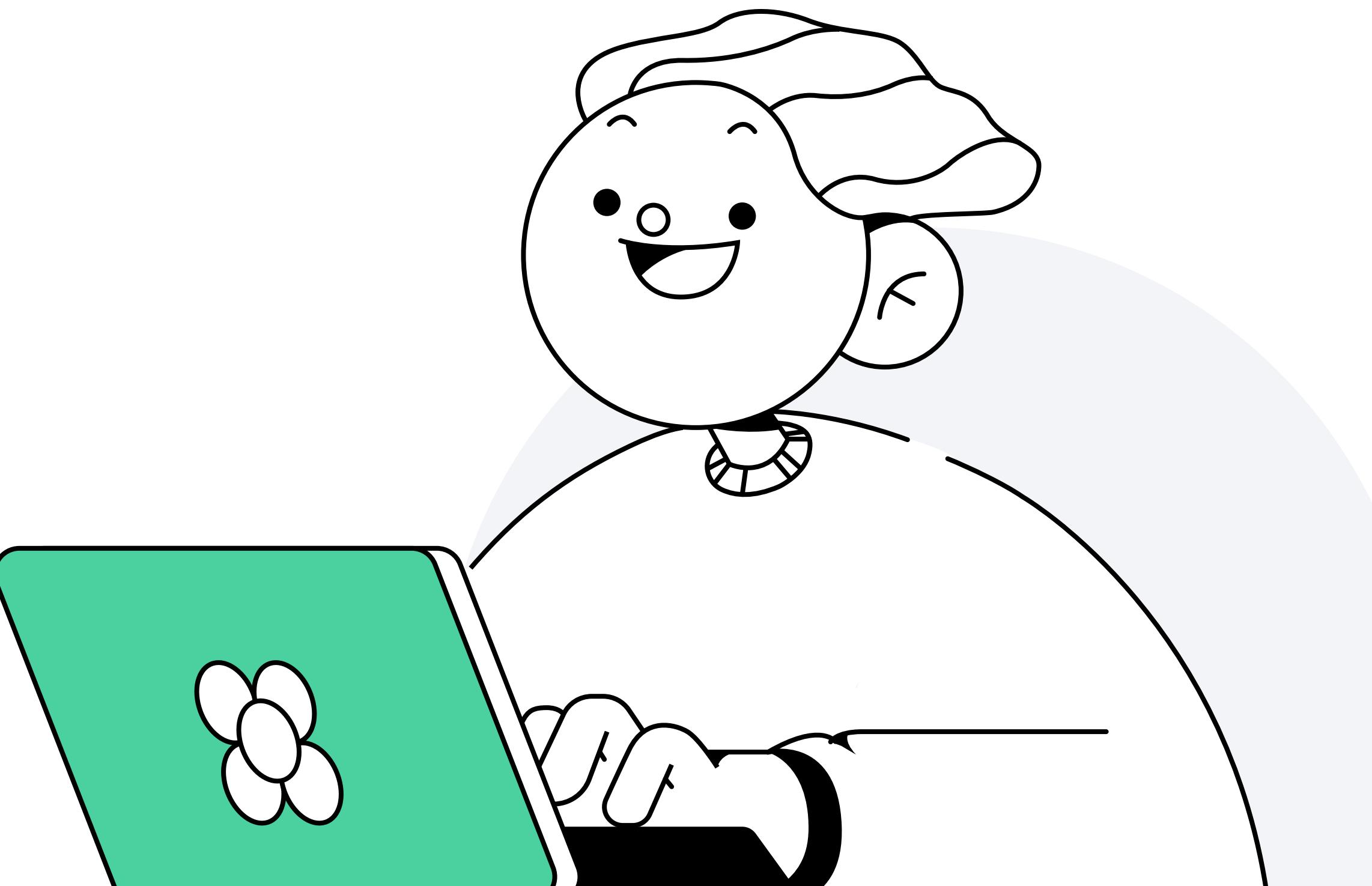
Цели занятия

- Научиться создавать задачи (issues)
- Научиться вносить изменения через pull request
- Научиться публиковать сайт на GitHub Pages
- Научиться удалять репозитории на GitHub



План занятия

- 1 Issues
- 2 Pull request
- 3 GitHub Pages и удаление репозитория

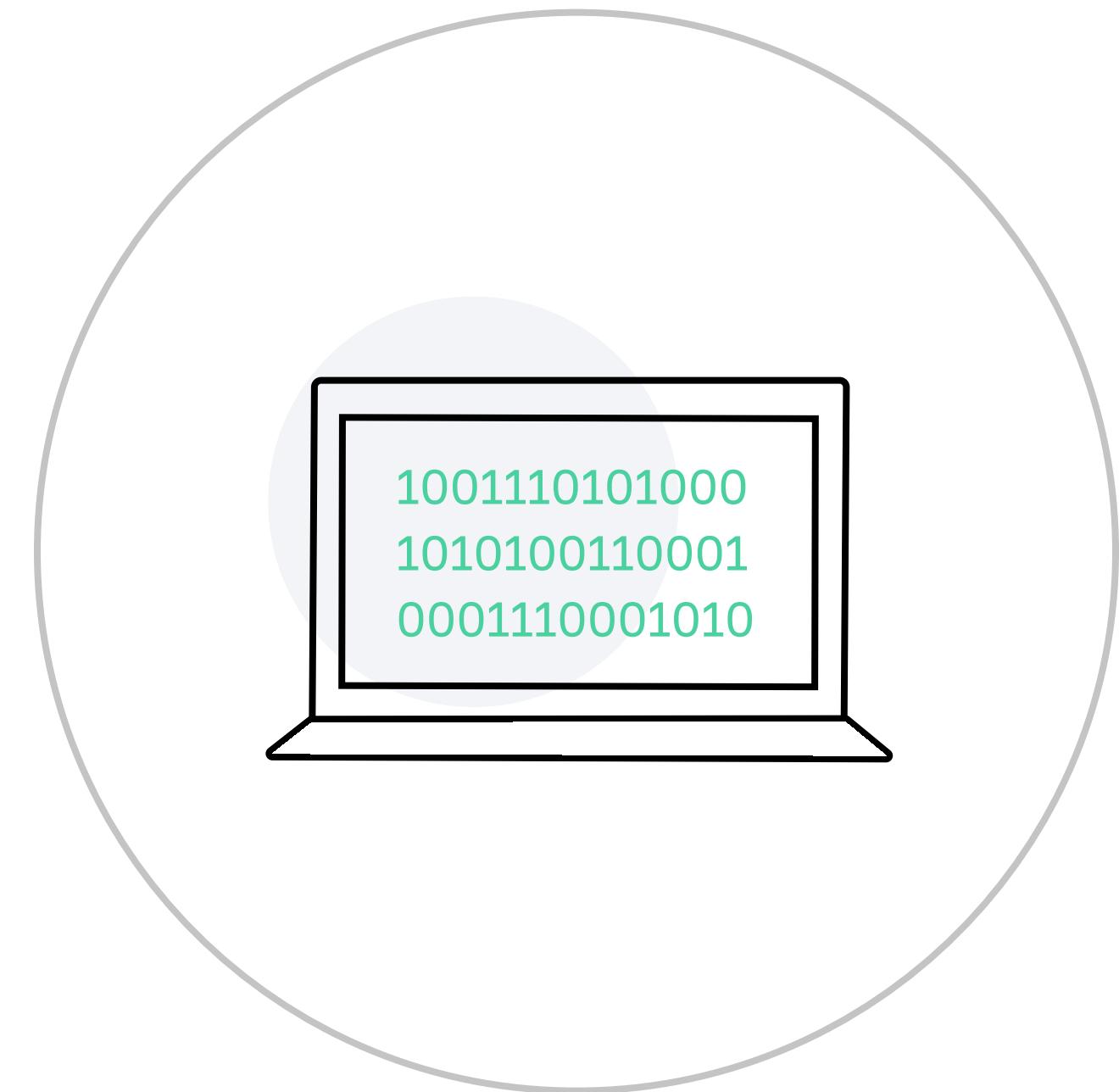


Issues

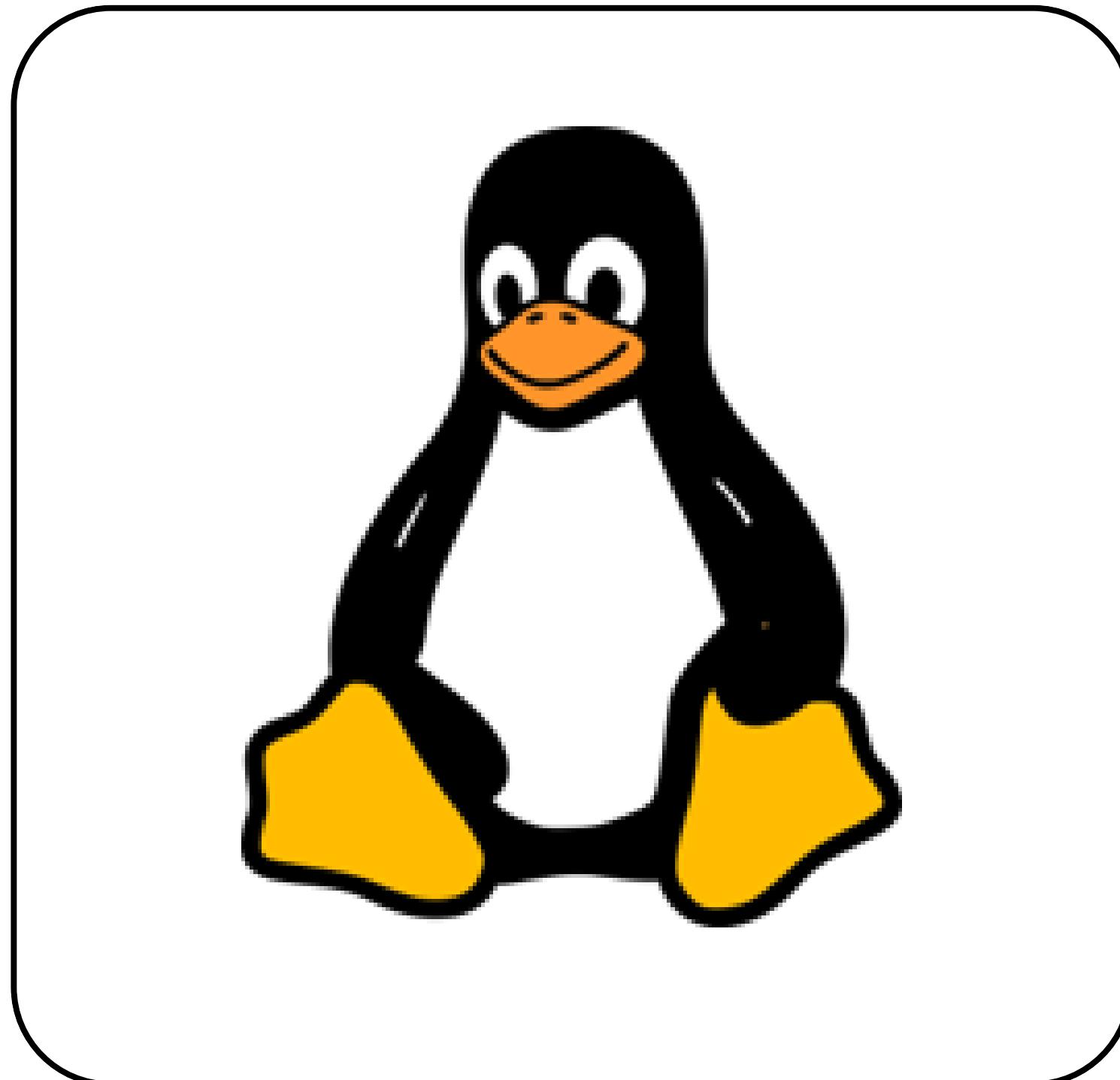
Открытые проекты (open-source projects)

Проекты с открытым исходным кодом:

- разрабатываются силами комьюнити
- каждый желающий может внести в него вклад



Известные open-source проекты



Операционная система Linux



Система контроля версий Git



Система управления базами данных MySQL

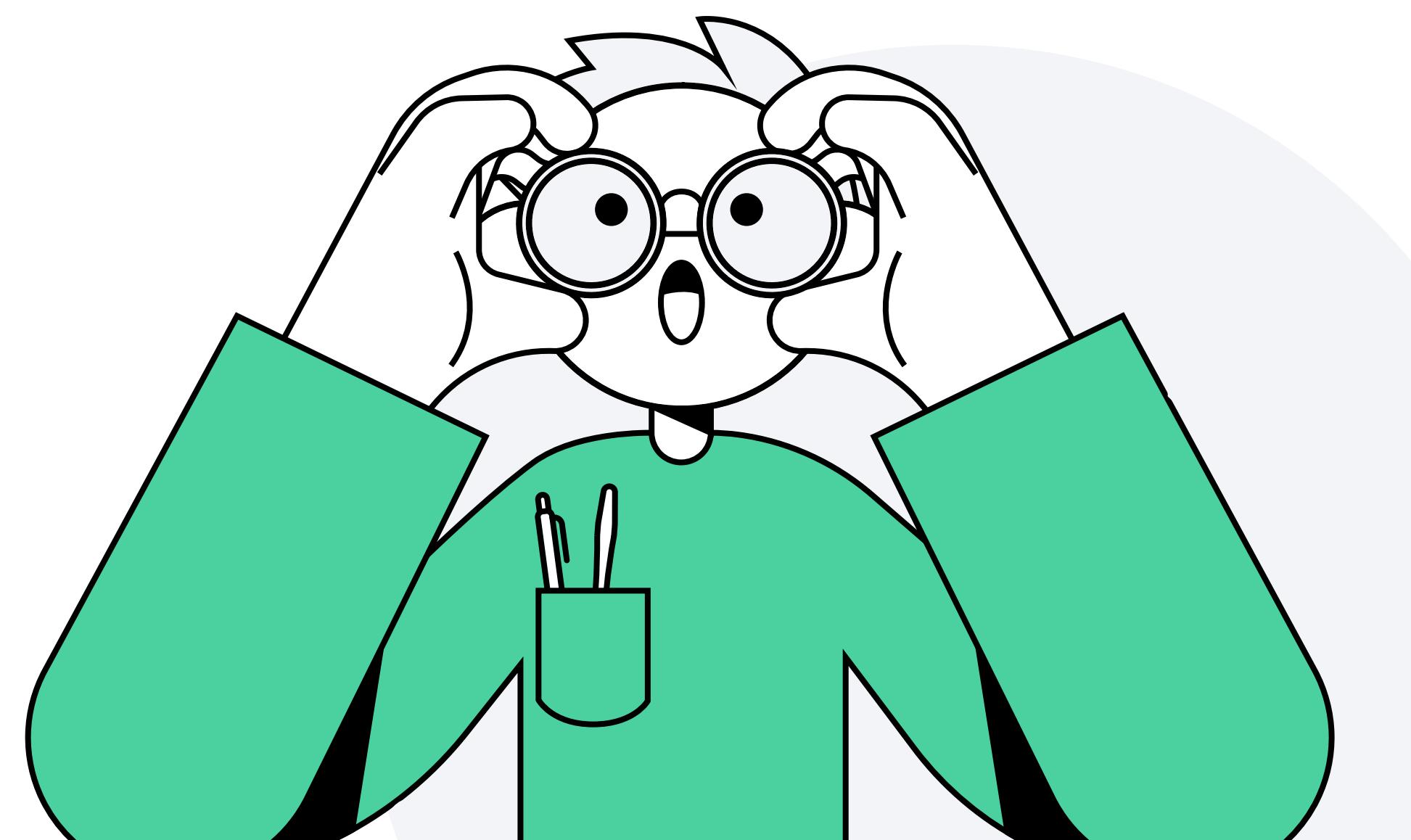
Issues

Система ведения задач в GitHub.
Облегчённый вариант трекера



Применение issues

- Сбор обратной связи от пользователей по ошибкам
- Планирование доработки проекта
- Обсуждение внутри каждой задачи
- Маркировка задач разными лейблами. Сбор задач в группы
- Назначение исполнителя для задачи



“

Многие возможности issues доступны только команде разработки проекта. Если у вас не доступны какие-то поля, то это значит, что вам не выданы соответствующие права. Вы по-прежнему можете создать issues, а кто-то из основной команды разработки настроит остальное

Скринкаст

Создание issues внутри проекта
в удалённом репозитории

Создание issues в GitHub

- 1 Чтобы создать issue на странице репозитория проекта, найдите вкладку **Issues**
- 2 Перейдите на эту вкладку. В правом верхнем углу найдите кнопку **New issue** и нажмите на неё
- 3 Откроется интерфейс создания задачи. Нужно ввести заголовок задачи, а ниже описать суть

Доступен визуальный редактор для форматирования текста: сделать текст жирным или курсивным, создать список, вставить ссылку, прикрепить файлы и вставить картинки прямо в текст. По факту весь текст является **Markdown**, поэтому можно вручную набирать нужные конструкции

Создание issues в GitHub

- 4 Чтобы упомянуть другого участника в тексте задачи, наберите **символ @** и напишите **имя пользователя на GitHub**. Появится всплывающее окно с предложениями. Можно выбрать нужного человека из списка
- 5 Чтобы ссылаться на другие задачи или pull request, наберите **символ #** и напишите **номер задачи или запроса на слияние**. Опять же будет предложен список в выпадающем меню.

В колонке справа есть несколько настроек:

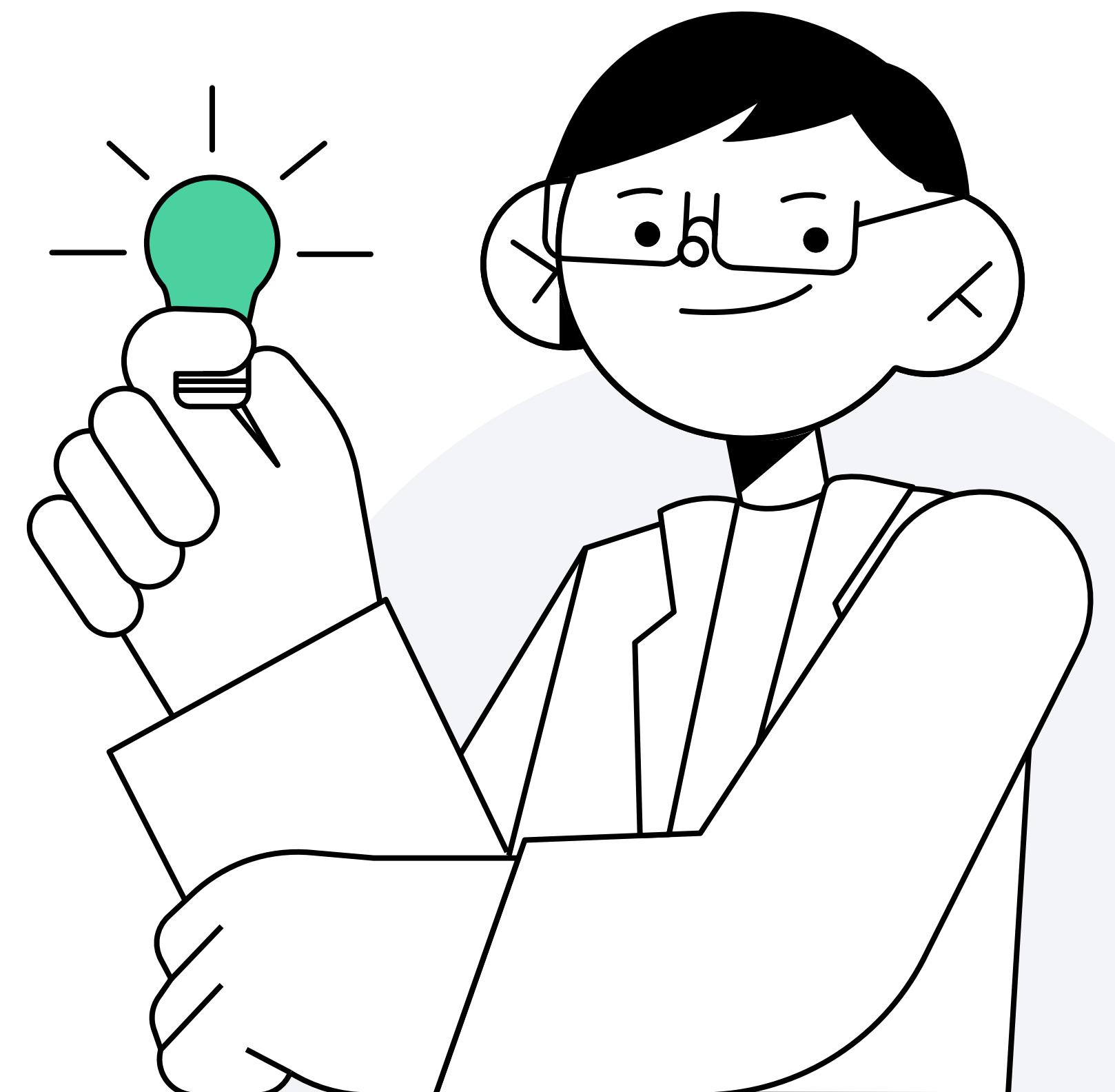
- В **Assignees** можно выбрать, кто именно будет отвечать за эту задачу
- При помощи **Labels** можно отметить, к чему относится эта задача: баг, улучшение, дизайн и так далее. Список лейблов зависит от настроек проекта

Создание issues в GitHub

- 6 Чтобы закончить создание issue, нажмите кнопку **Submit new issue**. Новому issue будет автоматически назначен порядковый номер. По нему можно обращаться к нему в других текстовых полях репозитория
- 7 После того, как issue создано, оно будет видно в списке всех задач репозитория. Владелец и контрибьюторы этого репозитория получат уведомления на почту. Если кто-то прокомментирует задачу или изменит какие-то настройки в ней, то вы тоже получите уведомление на почту.
 - После создания issue, можно отредактировать заголовок или описание. При этом все issues нельзя удалить. Их можно только закрыть. Они будут отображаться в этой же вкладке в разделе Closed

Итоги

- 1 Узнали, что для создания задачи или сообщения об ошибке нужно использовать вкладку Issues в GitHub
- 2 Научились создавать issues в GitHub



Pull request

Pull request

Запросы на слияние. Удобный способ предложить свои изменения в командный проект в формате, который предполагает обсуждение и код-ревью



Pull request

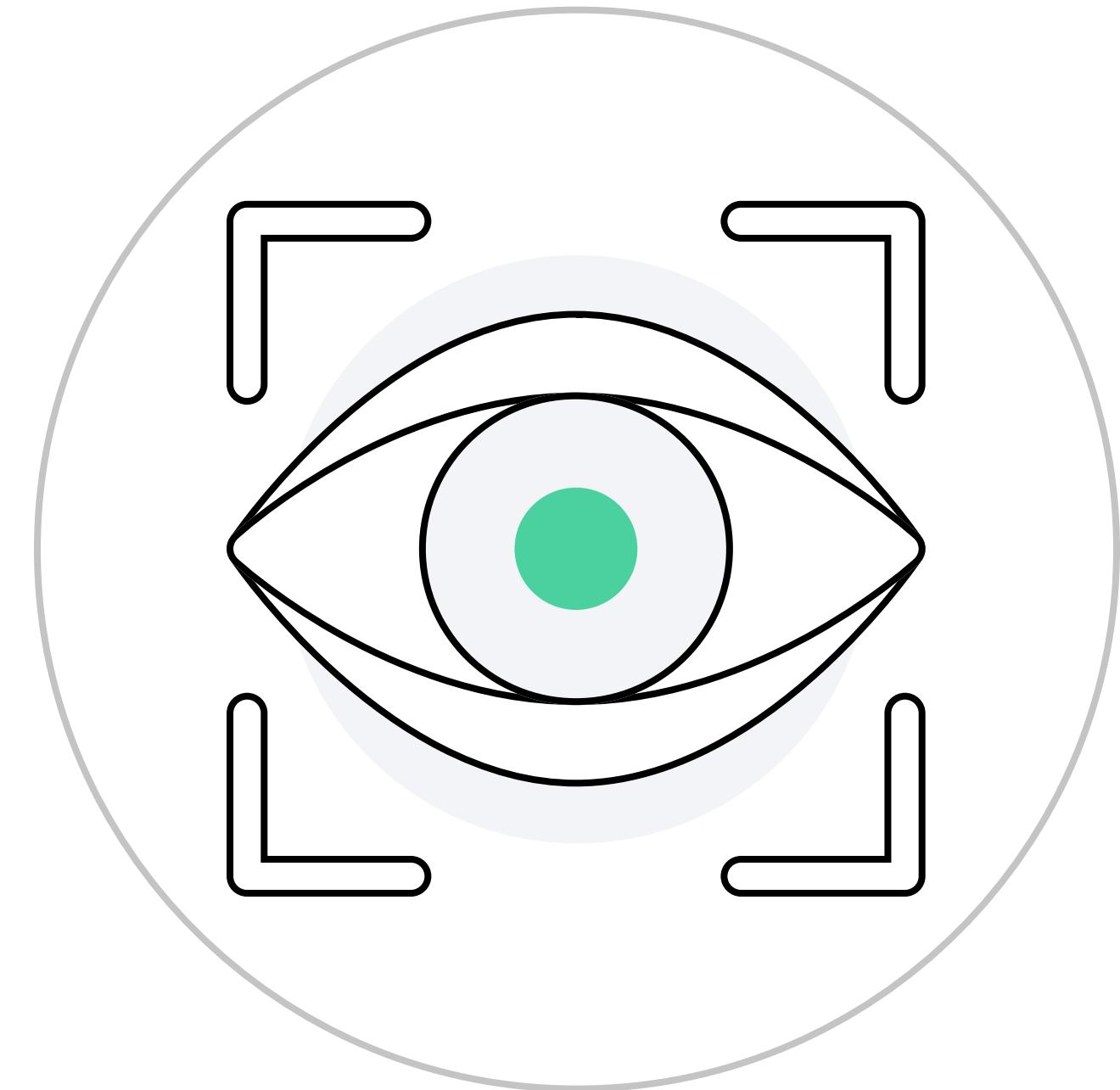
Используется:

- чтобы не засорять историю проекта коммитами
- внести изменения не в свой проект



Код-ревью

Процесс проверки кода на ошибки, работоспособность и соответствие внутренним правилам разработки команды



Скринкаст

Создание pull request, когда есть права на работу
в репозитории и когда их нет

Pull request при наличии прав на работу в репозитории

- 1 В локальной копии проекта создайте отдельную ветку. В ней внесите все нужные изменения, отправьте ветку в удалённый репозиторий
- 2 Перейдите на вкладку **Pull requests** на странице репозитория на GitHub и нажимаете кнопку **New pull request**
- 3 Выберите, изменения из какой ветки вы хотите предложить (правый выпадающий список)
- 4 Выберите, в какую ветку вы хотите влить изменения (левый выпадающий список). Как правило, главной веткой проекта является `main`
- 5 Если содержимое веток отличается друг от друга, то ниже будет виден список коммитов в вашей ветке, а также список файлов с подсвеченными изменениями
- 6 После этого нажмите кнопку **Create pull request**

Pull request при отсутствии прав на работу в репозитории

- 1 Создайте форк репозитория. В собственном форке добавьте изменения в основную ветку `main` или для соблюдения порядка создайте новую ветку под конкретные изменения
- 2 Чтобы создать запрос на слияние, перейдите на страницу исходного репозитория и откройте вкладку **Pull requests**, нажмите кнопку **New pull request**. Также можно прямо на главной странице форка найти и нажать кнопку **Contribute** и в открывшемся всплывающем окне нажать кнопку **Open pull request**. Кнопка будет активна, только если ваша копия проекта отличается от исходной
- 3 Дальше вы попадёте на страницу выбора источника изменений и ветки, в которую нужно выполнить слияние. Разница с предыдущим способом будет в том, что вашей ветки нет в проекте, она находится за его пределами. Чтобы выбрать её, нужно нажать на ссылку **compare across forks**. Появятся два дополнительных выпадающих списка, в которых можно выбрать репозитории. Левый трогать не нужно: там уже проставлен репозиторий проекта. В правом выберите ваш форк проекта. Если в своём форке вы вносили изменения не в ветку `main`, то выберите её в соседнем выпадающем списке

Описание изменений и создание pull request

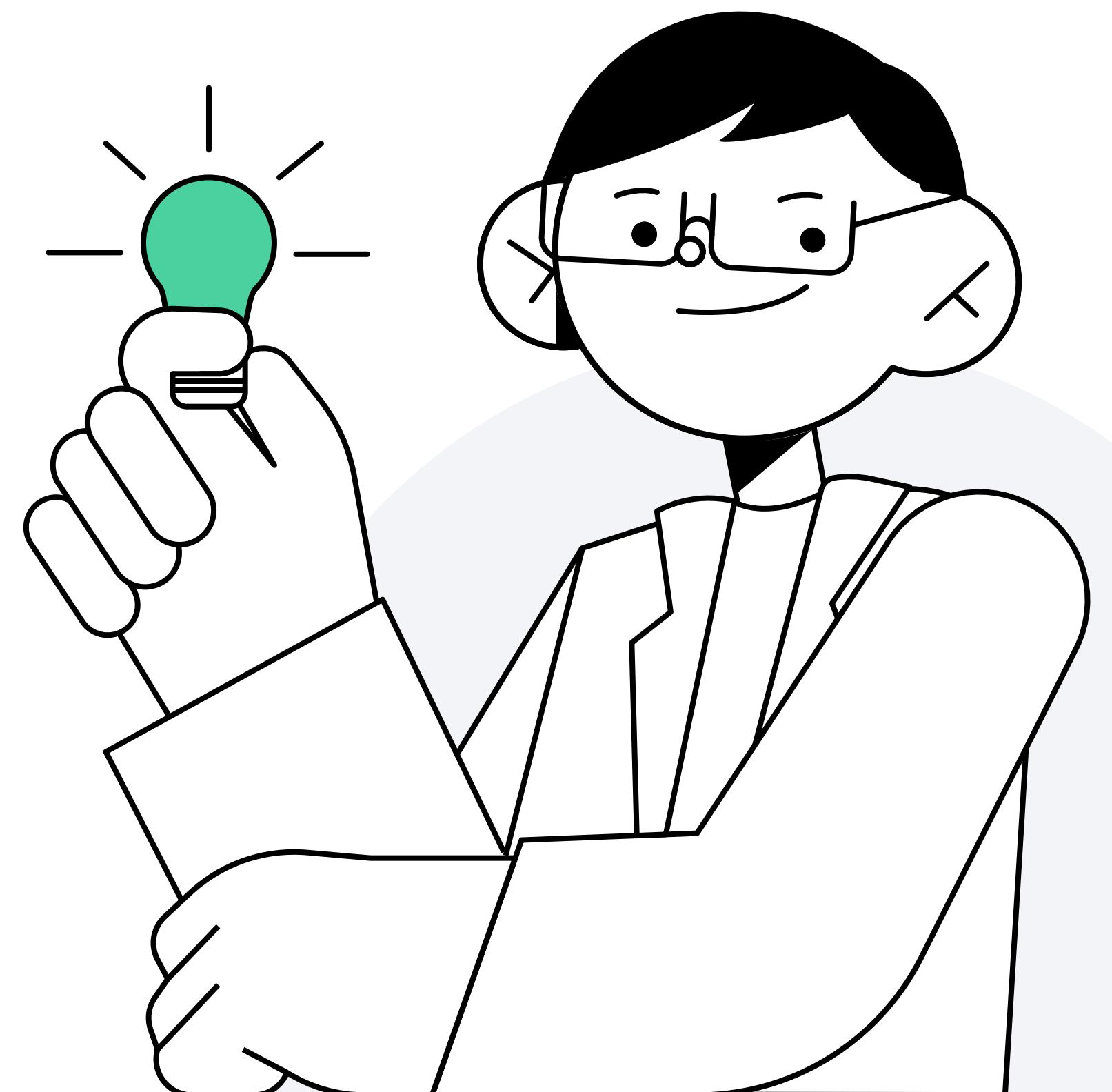
- 4 После нажатия кнопки **Create pull request** откроется интерфейс создания pull request, похожий на тот, что видели при создании issue. Нужно написать заголовок и добавить описание того, что вы предлагаете изменить. В правой колонке есть новая настройка: **Reviewers**. В ней, если у вас есть права на работу в этом репозитории, можно выбрать того, кто должен провести код-ревью изменений.

Важная особенность Markdown в рамках pull request – так называемые «закрывающие» слова. Если написать их в тексте запроса на слияние и дальше указать номер issue, то при слиянии изменений из этого pull request issue автоматически будет закрыто. Задача будет считаться решённой. Ссылку на подсказку по закрывающим словам можно найти в правой колонке. Чаще всего пишут «close #2030» или «fix #2030»

- 5 Чтобы закончить создание pull request, нажмите кнопку **Create pull request**. Ему будет присвоен порядковый номер. Запрос на слияние станет виден всем, кто зайдёт на страницу репозитория. Владелец и контрибьюторы проекта получат уведомления. Как и issues, pull request нельзя удалить, можно только закрыть

Итоги

- 1 Познакомились с pull request как способом внесения изменений в командный проект
- 2 Научились делать pull request для проекта, когда есть права на работу в репозитории и когда их нет. Для этого можно создать новую ветку или форкнуть исходный репозиторий на GitHub



GitHub Pages. Удаление репозитория

GitHub Pages

Бесплатный хостинг для статических файлов, созданных на базе HTML, CSS, JavaScript и Markdown.

Его удобно использовать для демонстрации своей работы заказчику и работодателю



Скринкаст

Публикация сайта на GitHub Pages

Публикация сайта на GitHub Pages

- 1 Для публикации сайта перейдите на вкладку **Settings** на странице репозитория. В левой колонке найдите пункт **Pages**
- 2 На открывшейся странице в первом выпадающем списке выберите ветку, код из которой должен использоваться для сайта. Дальше, при необходимости, выберите конкретную папку в структуре ваших файлов или оставьте /root по умолчанию. Нажмите **Save**
- 3 После сохранения вы увидите ссылку, по которой теперь доступен ваш сайт. Публикация сайта может занять до 30 минут. Не пугайтесь ошибки 404 при переходе по ссылке сразу после сохранения настроек. Подождите немного и обновите страницу. Аналогично с применением изменений. Они могут отобразиться на сайте не сразу после коммита
- 4 Чтобы отменить публикацию, в выпадающем списке с ветками выберите **None** и нажмите кнопку **Save**. Сайт будет снят с публикации и ссылка перестанет работать

Скринкаст

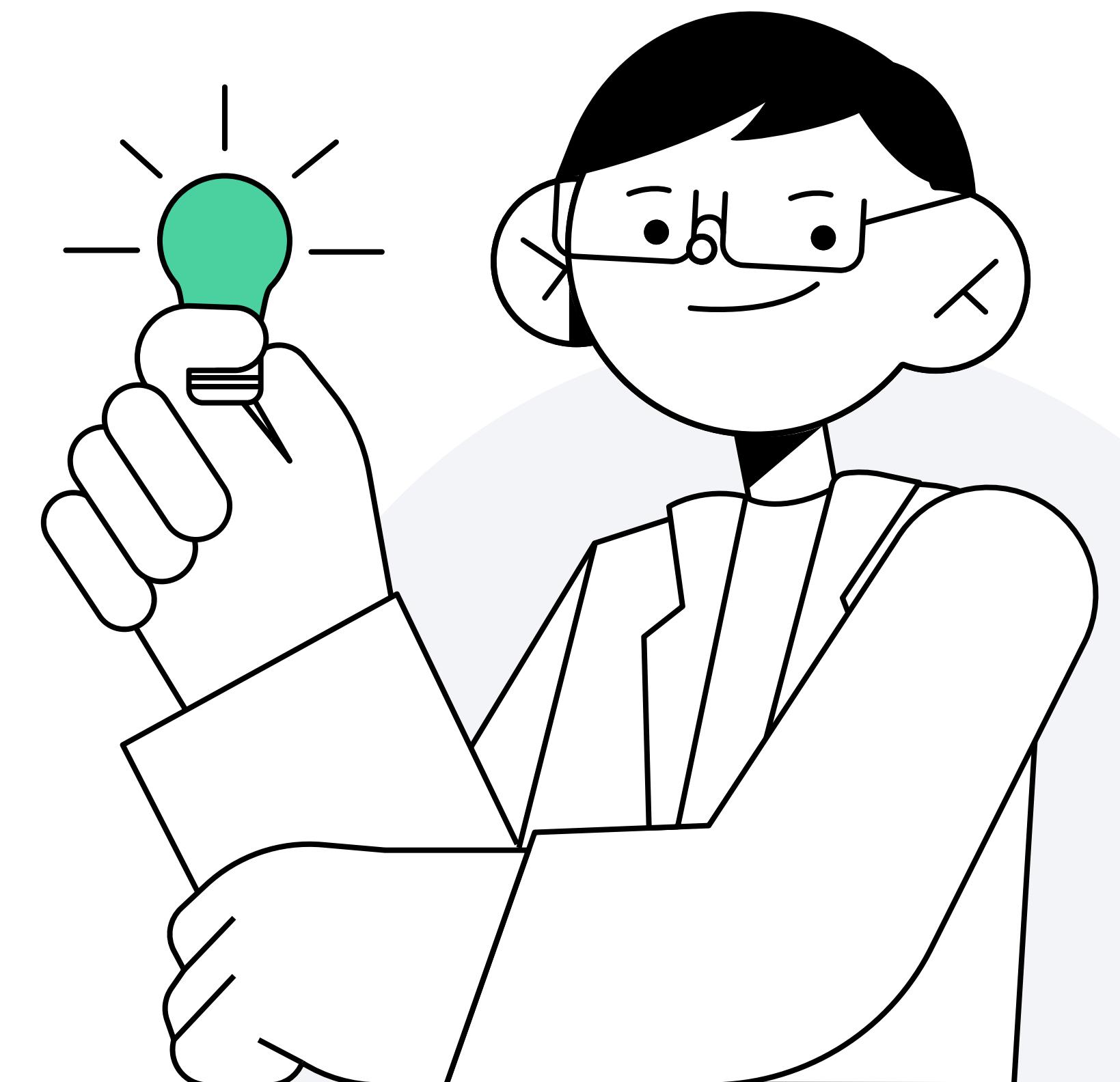
Удаление репозитория с GitHub

Удаление репозитория с GitHub

- 1 Перейдите на вкладку **Settings**. В разделе **General** прокрутите страницу до самого конца и найдите блок **Danger zone**
- 2 Чтобы удалить репозиторий, нажмите кнопку **Delete this repository**
- 3 Во всплывающем окне введите название репозитория, чтобы подтвердить намерение его удалить. **Будьте аккуратны – восстановить репозиторий на GitHub будет нельзя**

Итоги

- 1 Узнали, что для публикации простых статических сайтов можно использовать бесплатный сервис GitHub Pages
- 2 Научились публиковать собственный сайт для демонстрации работы заказчикам и работодателям на GitHub Pages
- 3 Научились удалять репозиторий из GitHub с помощью функции Danger zone



Спасибо за внимание

Алёна Батицкая

Фронтенд-разработчик, фрилансер

