Как устроен и работает GitHub

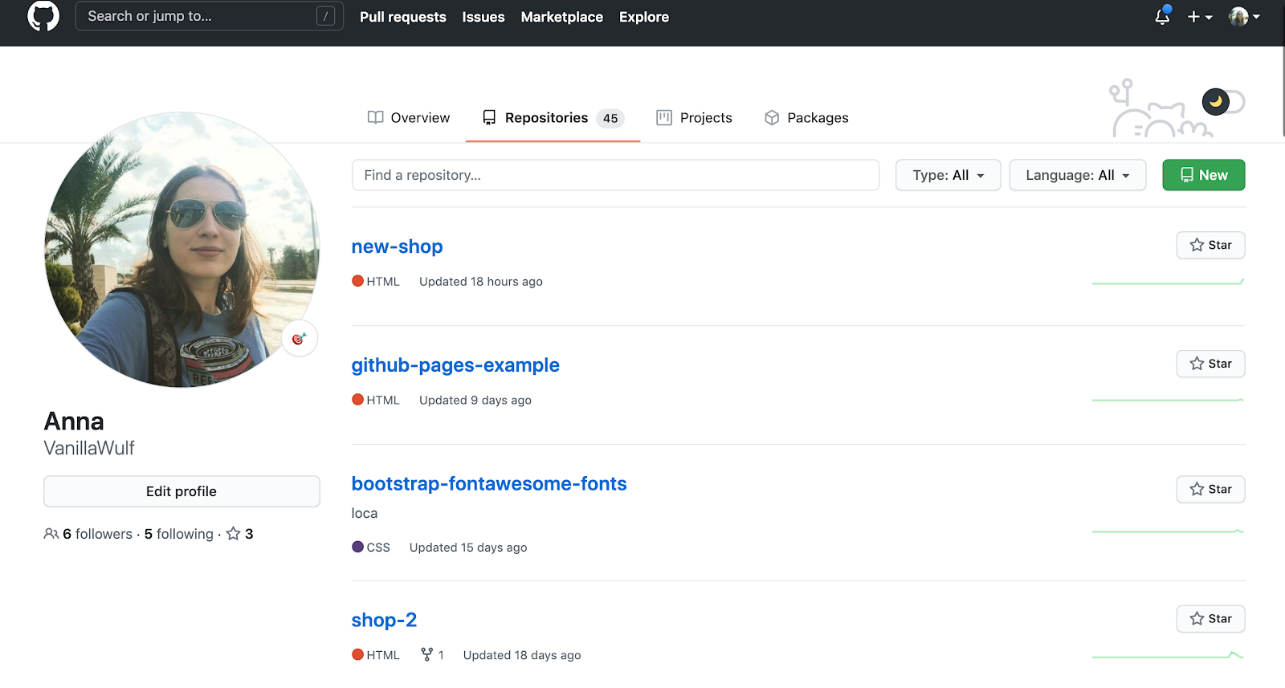
Один из самых популярных хостингов для проектов среди разработчиков

23 марта 20212 минуты16191

GitHub — это хостинг: он позволяет хранить проекты удалённо на сервере и работать с ними из любой точки мира. Доступ к файлам есть у всех, у кого есть ссылка.

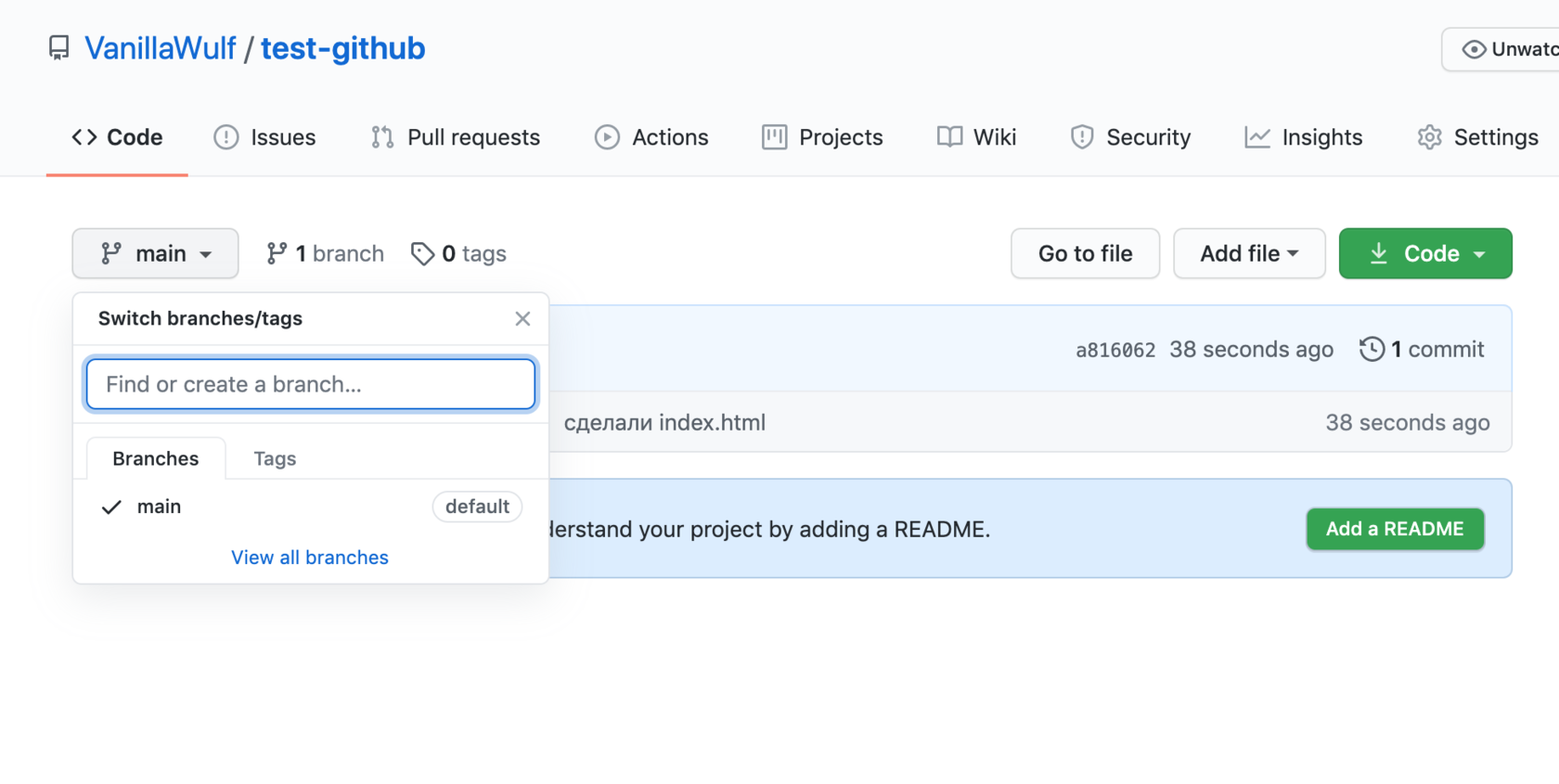
Одна из главных функций GitHub — контроль версий. Все изменения в коде можно отследить, поэтому в командной разработке это незаменимая вещь.

Кроме того, для начинающих разработчиков GitHub — это не только хостинг, но и способ собрать портфолио. Он позволяет продемонстрировать свои навыки и знания в виде реализованных проектов — во вкладке Repositories.

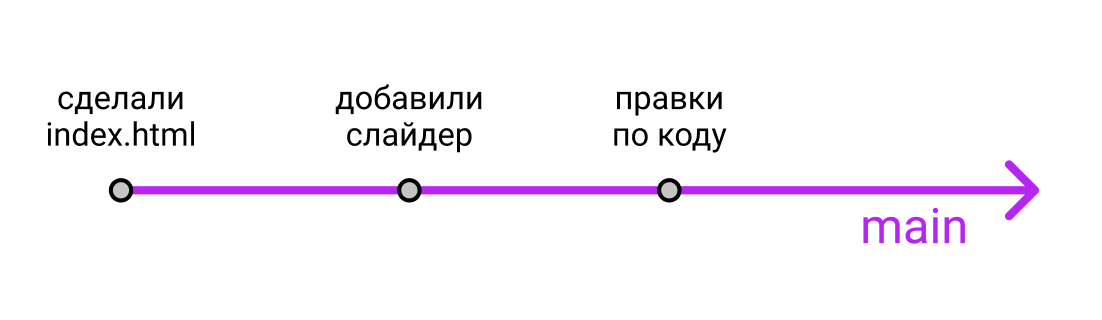


Из чего состоит проект в GitHub

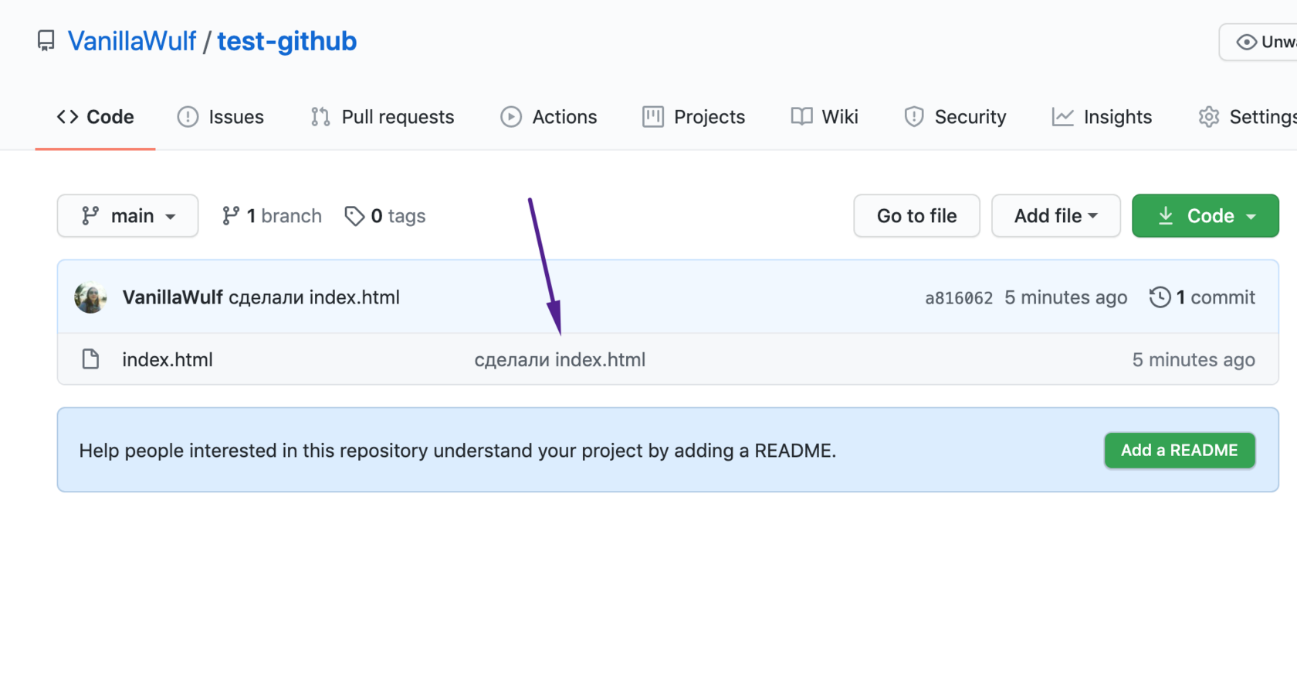
**Ветка main (или master)**. Это, как правило, основная ветка, в которой лежит самая актуальная, рабочая версия проекта.



**Ветки и коммиты (branch & commits).** Ветка в GitHub — это история разработки, состоящая из изменённых файлов и сообщений (коммитов). Нарисуем ветку с коммитами, чтобы визуально её представить.

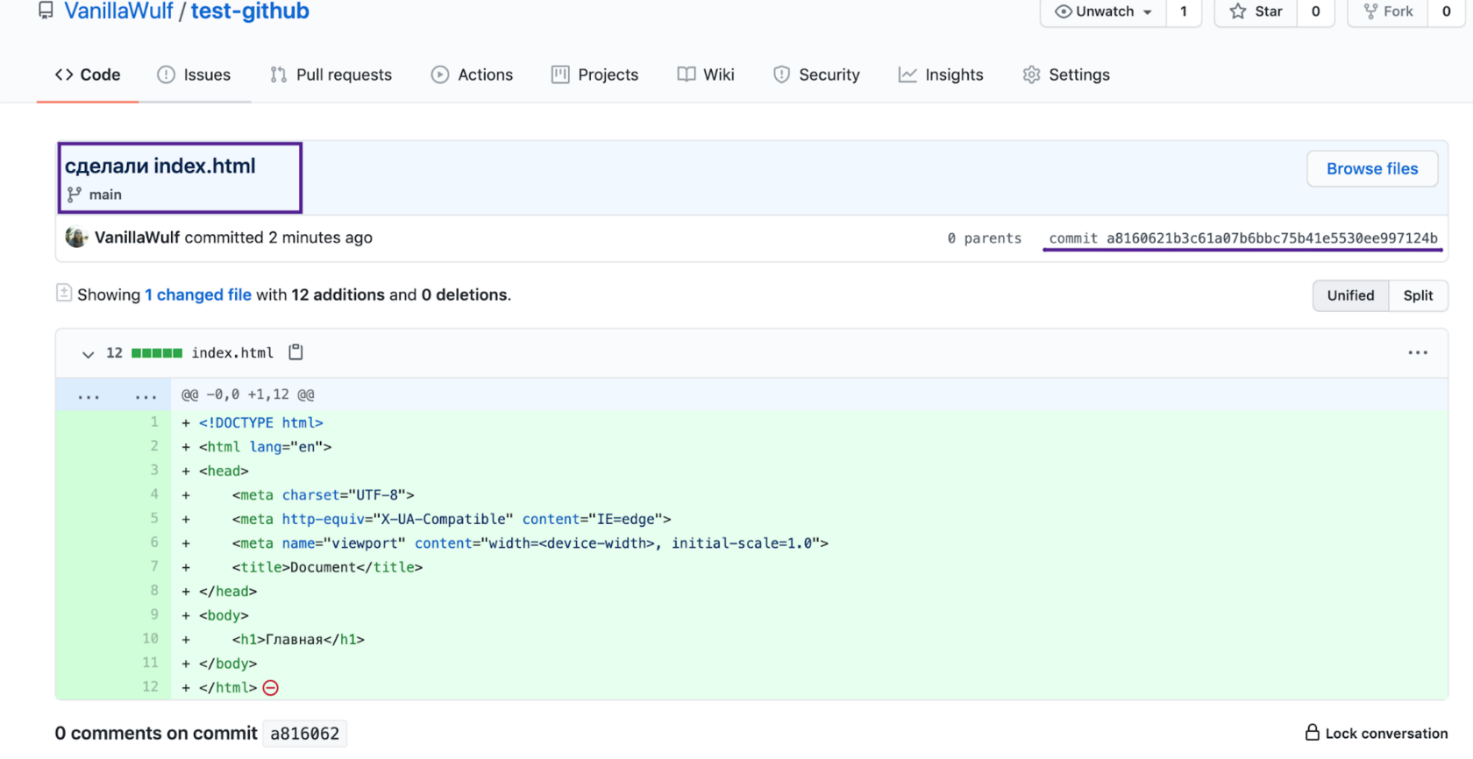


Коммит (commit) — это специальное сообщение, в котором мы указываем, что сделали за эту итерацию. Когда отправим коммит с файлами на сервер, сможем просмотреть изменения, если кликнем по нему на GitHub. Для изучения будем использовать [репозиторий с простым проектом](https://github.com/VanillaWulf/test-github).



Коммит состоит из сообщения (например, «сделали index.html») и файлов, которые мы прикрепили к коммиту. К ним автоматически добавляются время, автор и указатель HEAD.

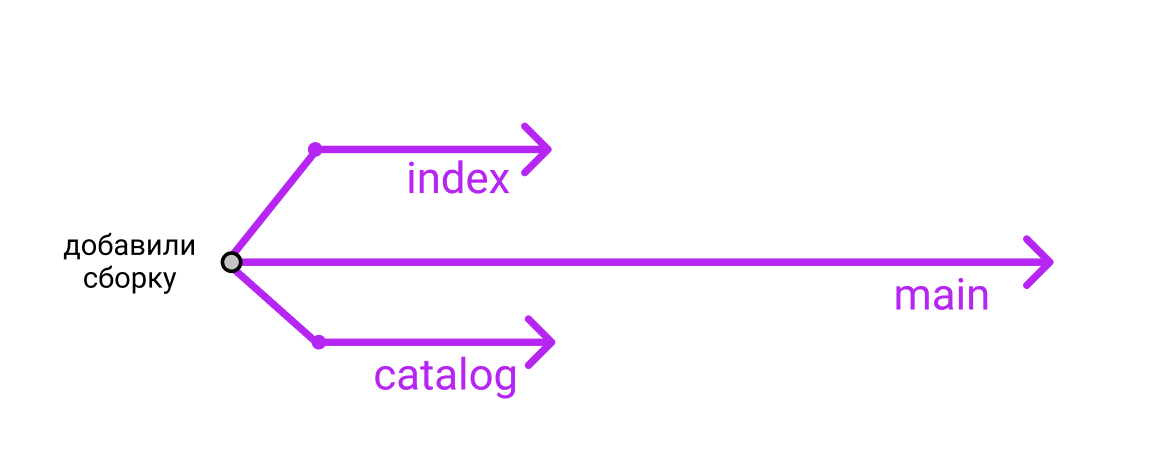
**Указатель HEAD** позволяет гибко откатиться до нужной версии. HEAD на скриншоте ниже выглядит так: a8160621b3c61a07b6bbc75b41e5530ee997124b.  
  
**Пример**. Мы запушили (от слова push, то есть отправили на сервер) коммит с сообщением «правки по коду» в ветку main, но поняли, что они ломают важную логику и версия теперь нестабильна. При помощи указателя коммита HEAD мы можем откатиться до предыдущего коммита («добавили слайдер») и вернуть стабильную версию, которая была в коммите «сделали index.html».



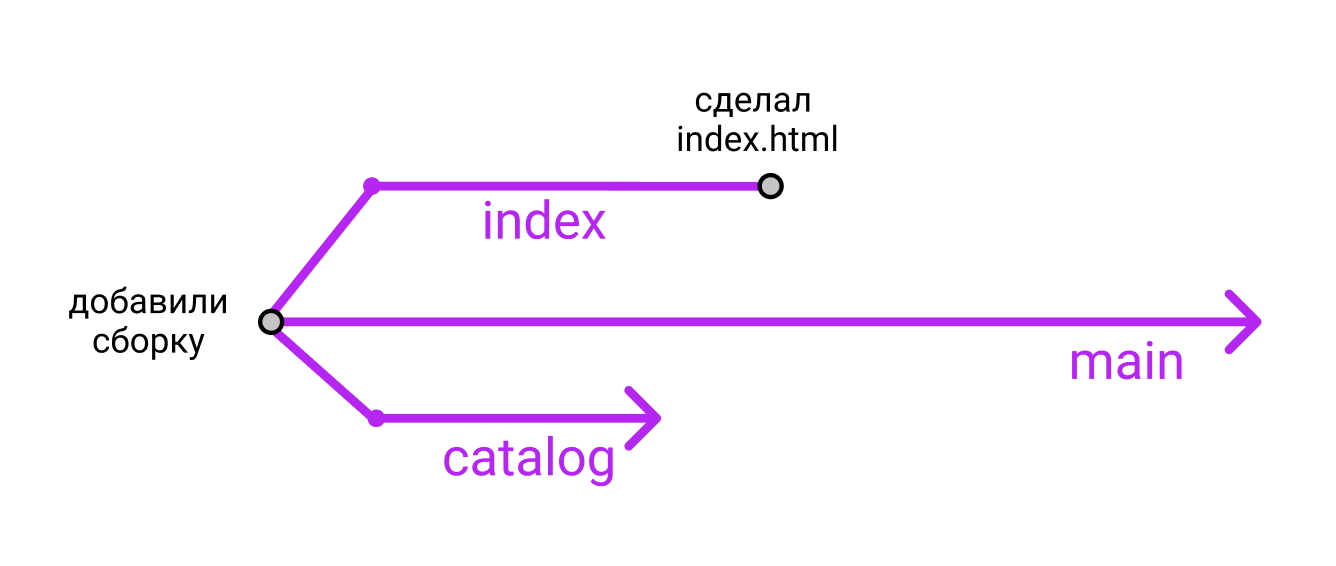
Стабильная и актуальная версия проекта, как правило, лежит в ветке main. Но мы можем создавать и свои ветки для новых задач, чтобы:

* не мешать другим разработчикам работать над задачами,
* не сломать текущую версию.

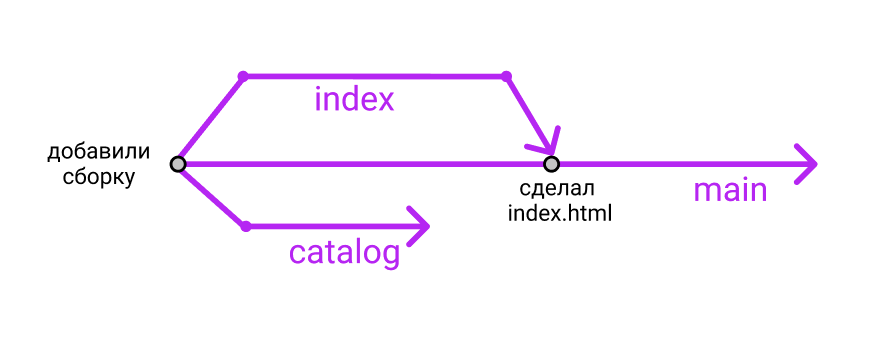
**Пример**. Разработчик Ваня берёт задачку по вёрстке index.html в ветке index. А разработчица Лена — catalog.html, в ветке catalog. На картинке ниже вы увидите, как это будет выглядеть на нашем визуальном отображении веток. Последний коммит в каждой ветке будет «добавили сборку».



Ваня быстрее Лены заканчивает свою задачку, делает коммит «сделал index.html» в свою ветку:



После того как он сольёт свою ветку в main, в ней появится его последний коммит «сделал index.html»:

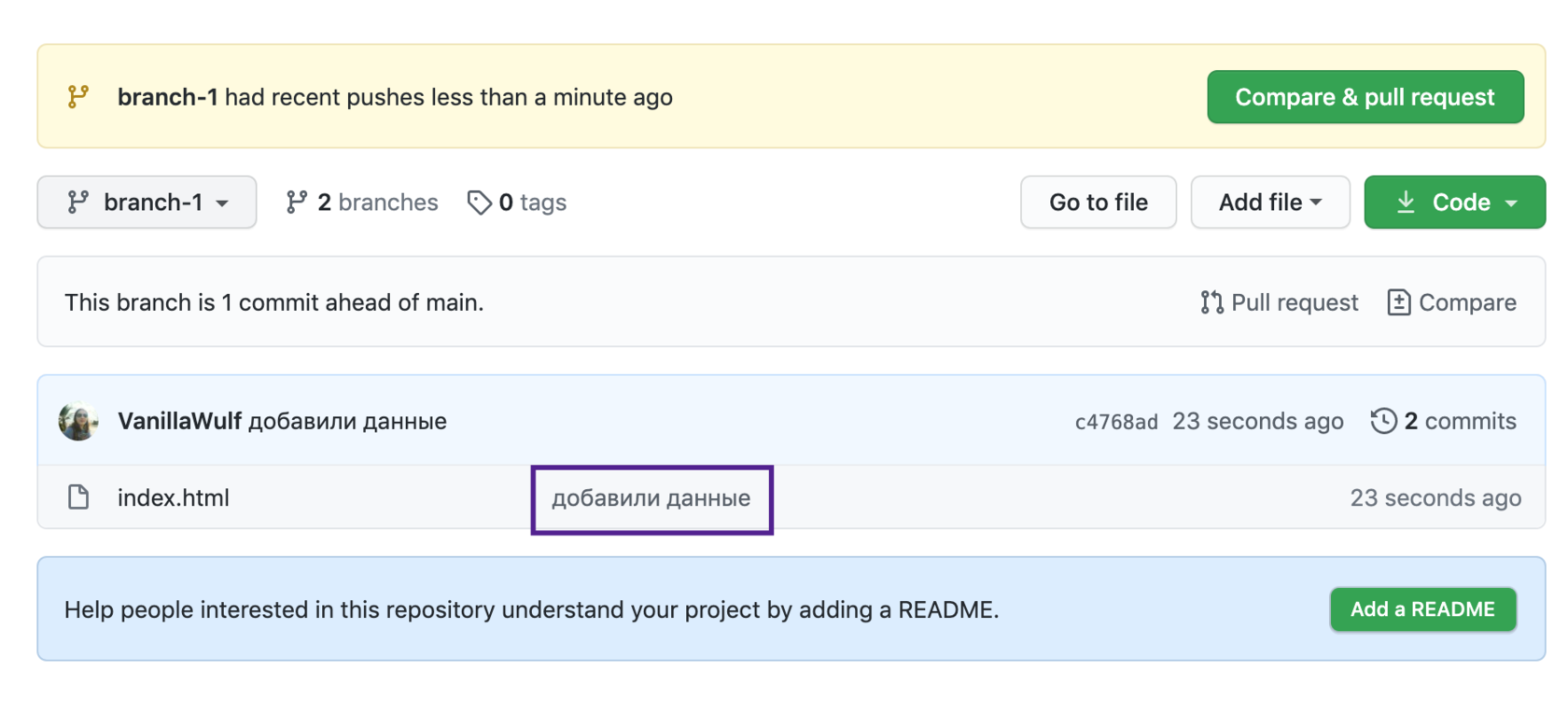


Лена может забрать актуальную версию main (pull) в свою ветку. Тогда история её коммитов будет выглядеть так: «добавили сборку, сделал index.html».

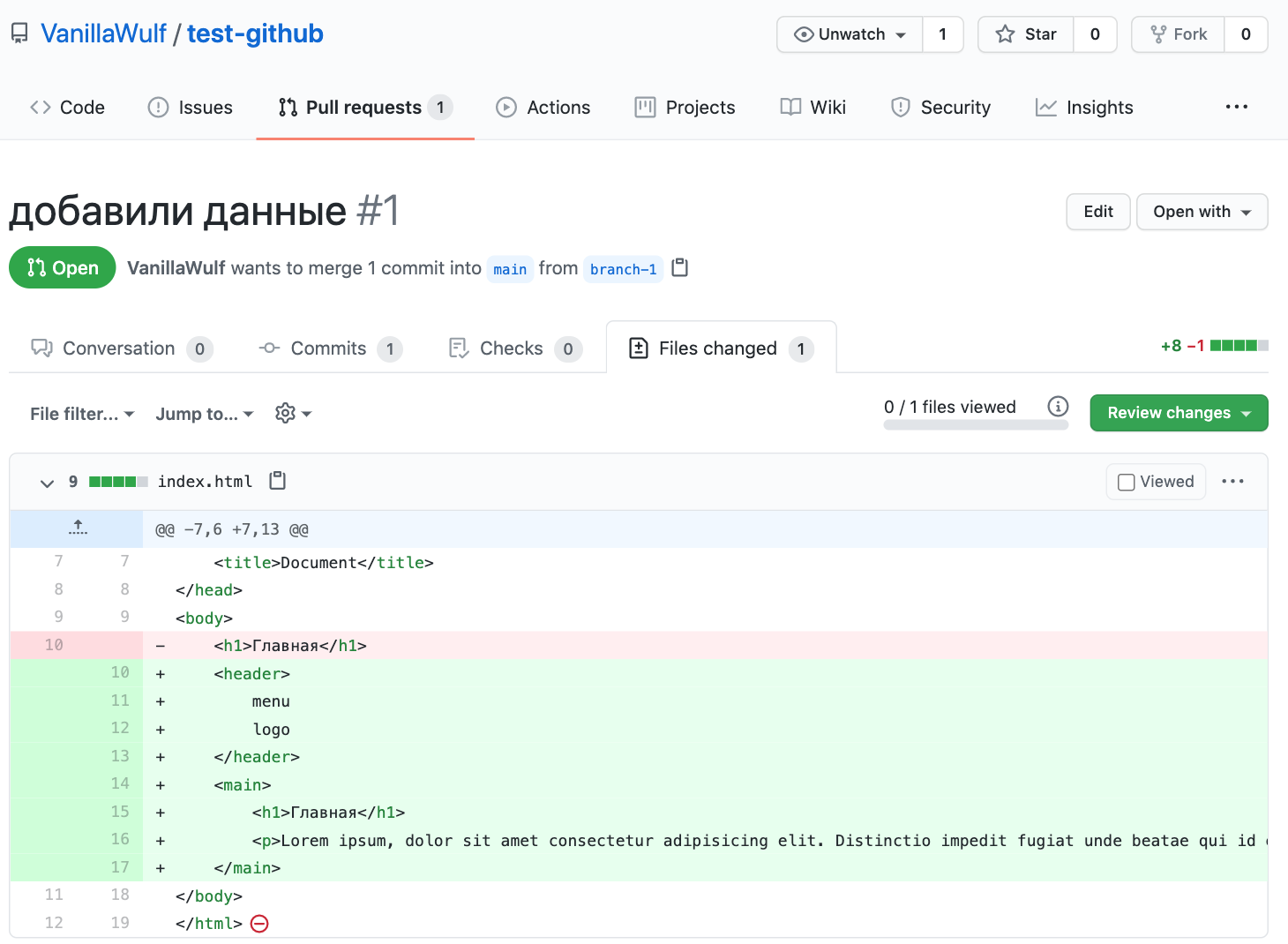
Слияние веток

Ветки позволяют просматривать изменения. А чтобы сделать их слияние, понадобится функция pull request (pr).

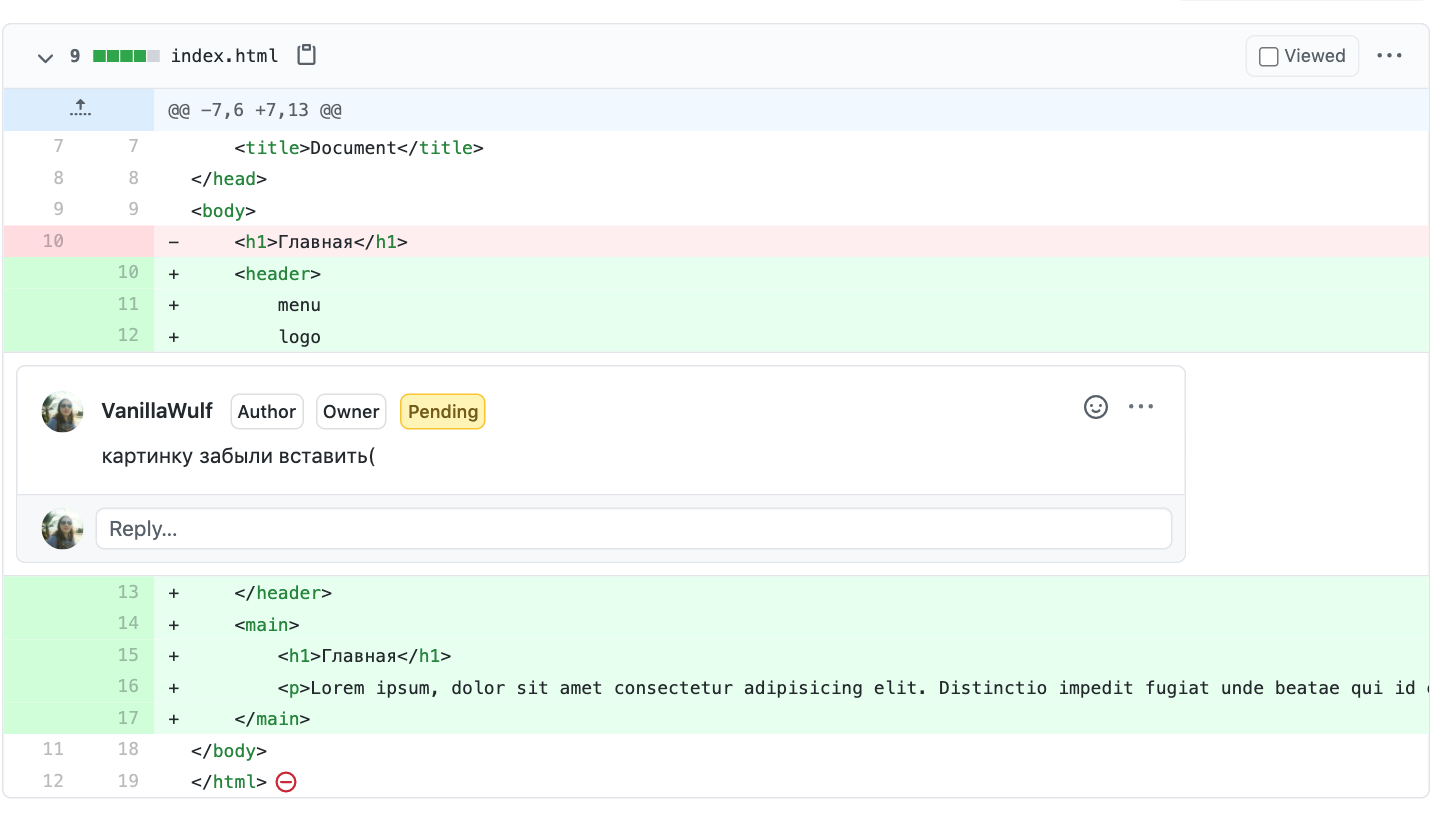
**Пример.** Создаём ветку branch-1 в нашем репозитории и делаем в неё пуш коммита «Добавили данные»:



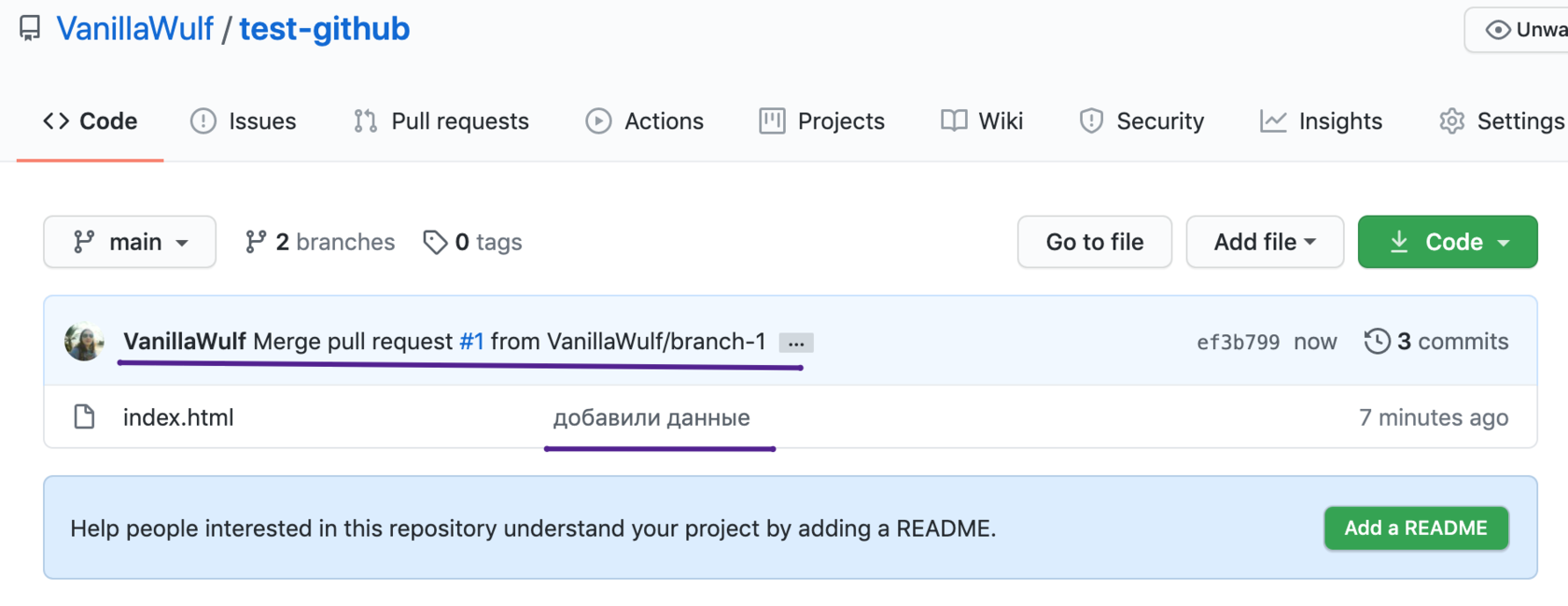
После этого мы можем сделать pull request в любую ветку, то есть заявку на слияние веток. GitHub сам подсветит изменённые строки, что очень удобно для ревью кода. Красным подсвечиваются удалённые строки, зелёным — добавленные.



Ревью кода

Когда над проектом ведут работу несколько разработчиков, очень важно, чтобы каждый понимал, что происходит в проекте, уточнял спорные моменты и предлагал более оптимальные решения. Этот процесс называется code review (ревью кода).  


После того как ревью проведено, все спорные моменты решены, ошибки поправлены и есть подтверждения от других разработчиков (approve), происходит слияние веток (merge). Все коммиты из ветки branch-1 попадают в main, а в истории коммитов появляется отметка о слиянии.



Это вводная статья по GitHub, которая поверхностно описывает основные процессы. О создании веток и управлении ими мы обязательно расскажем в следующих материалах.