

GIT

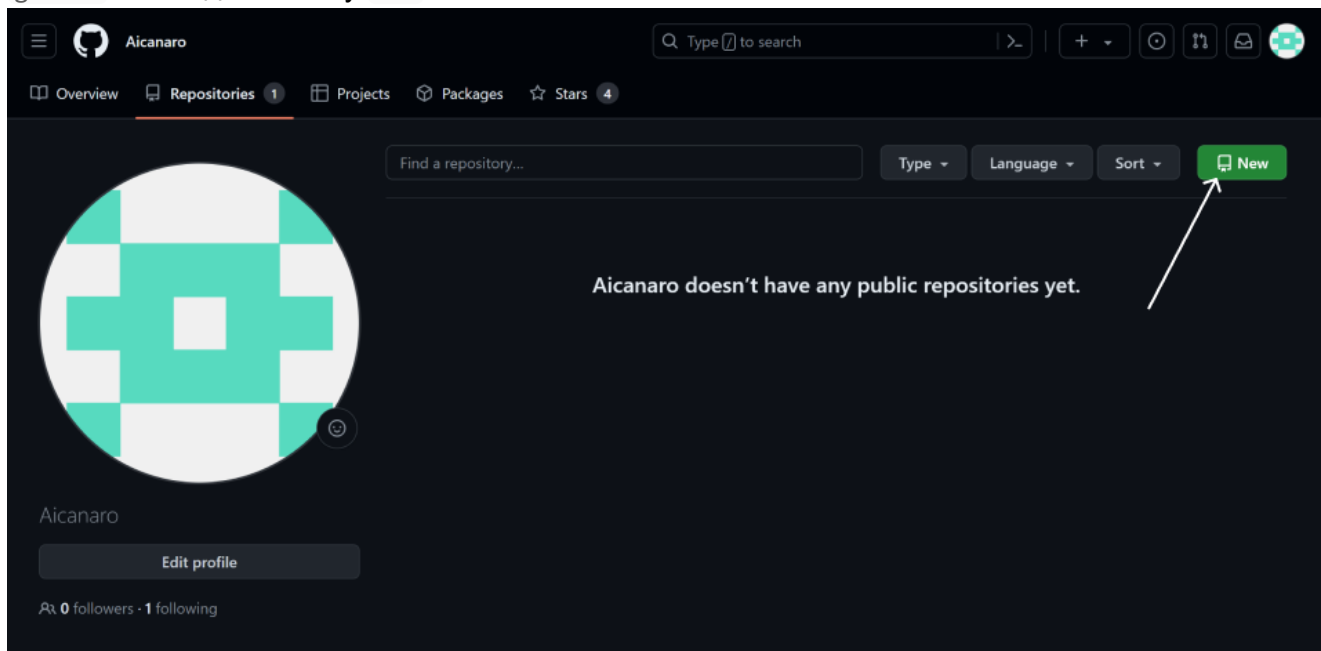
Git или система контроля версий, была сделана для разработчиков чтобы можно было бы сделать некие checkpoint'ы для работы программ, и сейчас я попробую простым языком рассказать что вообще нужно сделать для корректной работы гита.

Подготовка

Прежде чем сделать репозиторий и начать работу в нем необходимо скачать сам гит с их [официального сайта](#), после нужно зарегистрироваться на сайте [гитхаба](#), ведь именно с ним мы и будем работать, и будем там просматривать все изменения.

Начало работы

После успешной установки `git` и регистрации на `github` мы заходим в наш аккаунт на `github` и находим кнопку `new`




Далее мы вводим имя нашего репозитория в котором будем работать, для ознакомления мой репозиторий будет назван `rep` и сделаю его `публичным`

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Required fields are marked with an asterisk ().*


Owner * Repository name * Имя Репозитория

 Alcanaro /

Great repository names are short and memorable. Need inspiration? How about [automatic-octo-guide](#) ?

Description (optional)

☒  **Public** Публичный
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private** Приватный
You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: **None** ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: **None** ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

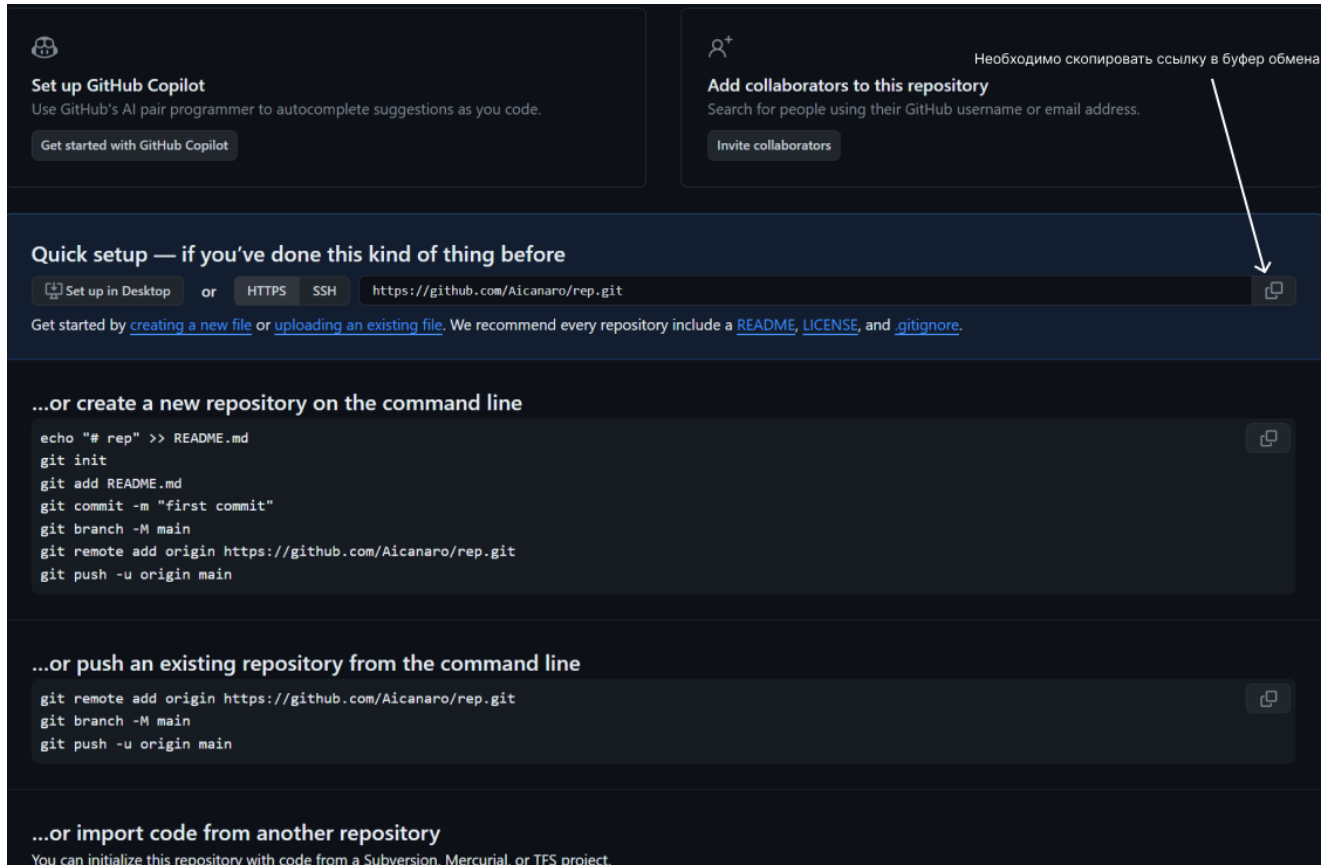
[Create repository](#)

Публичный - весь интернет сможет ознакомиться с вашей работой, доработать ваш проект или внести в него свои коррективы.

Приватный - ты сам выбираешь людей которые могут видеть и дополнять твой репозиторий.

После уже нажимаем `create repository` и мы попадаем на страницу которая очень страшная и там много чего непонятного но все что нам нужно сделать это скопировать

ссылку



Начало настоящей работы

Для удобства я сделаю на рабочем столе папку которую назову `rep` - НАЗВАНИЕ ПАПКИ НЕ РАВНО НАЗВАНИЮ ДИРЕКТОРИИ Я ЭТО СДЕЛАЛ ДЛЯ УДОБСТВА ЧТЕНИЯ. И в ней о нажатую ПКМ (Правой Кнопкой Мыши) в появившемся окошке выбираем `open Git Bash here` и сразу пишем 2 команды:


1. `git init`
2. `git clone` (скопированная ссылка без скобочек)

И все, гит настроен и готов к работе, но чтобы все точно заработало, необходимо закрыть окно консоли и попасть в появившуюся папку (папка будет иметь тоже самое название что и ваше название репозитория на `github`) и там для примера сделать текстовый документ, неважно какой и будет ли в нем что либо, главное чтобы мы сделали файл, я сделал файл `Млем.txt`


И снова в этом окне мы щелкаем ПКМ и выбираем `open Git Bash here` после мы сделаем 4 команды чтобы отправить изменение на наш сервер (да, репозиторий что мы делали в `github` можно назвать сервером):

1. `git status`
2. `git add -A`
3. `git commit -m "Млем добавлено в репозиторий"`
4. `git push`

И все, наши изменения видны и на сервере

 **Aicanaro** Млем

ebed607 · 26 minutes ago 1 Commits

 Млем.txt

Млем

26 minutes ago

А теперь разберем что за команды я использовал

`git status` - отображает статус нашего репозитория и показывает какие файлы подверглись изменениям

`git add -A` - мы создаем новые копии наших файлов, в последствии именно они будут отправлены на сервер модификатор `-A` / `-all` это одно и тоже, но мне привычнее писать `-A`

`git commit -m "Твои слова"` - создает некий checkpoint, а метод `-m ""` дает дать коммиту название, или написать там то что было изменено чтобы было проще ориентироваться среди коммитов

`git push` - отправить последний коммит на сервер

В целом это основные команды для работы с гитом

push (толкнуть) `Local -> Server`

pull (притянуть) `Local <- Server`

add (добавить) `git add -A`

commit (не придумал аналогии) `git commit -m "..."`

status (отобрази статус репозитория) `git status`