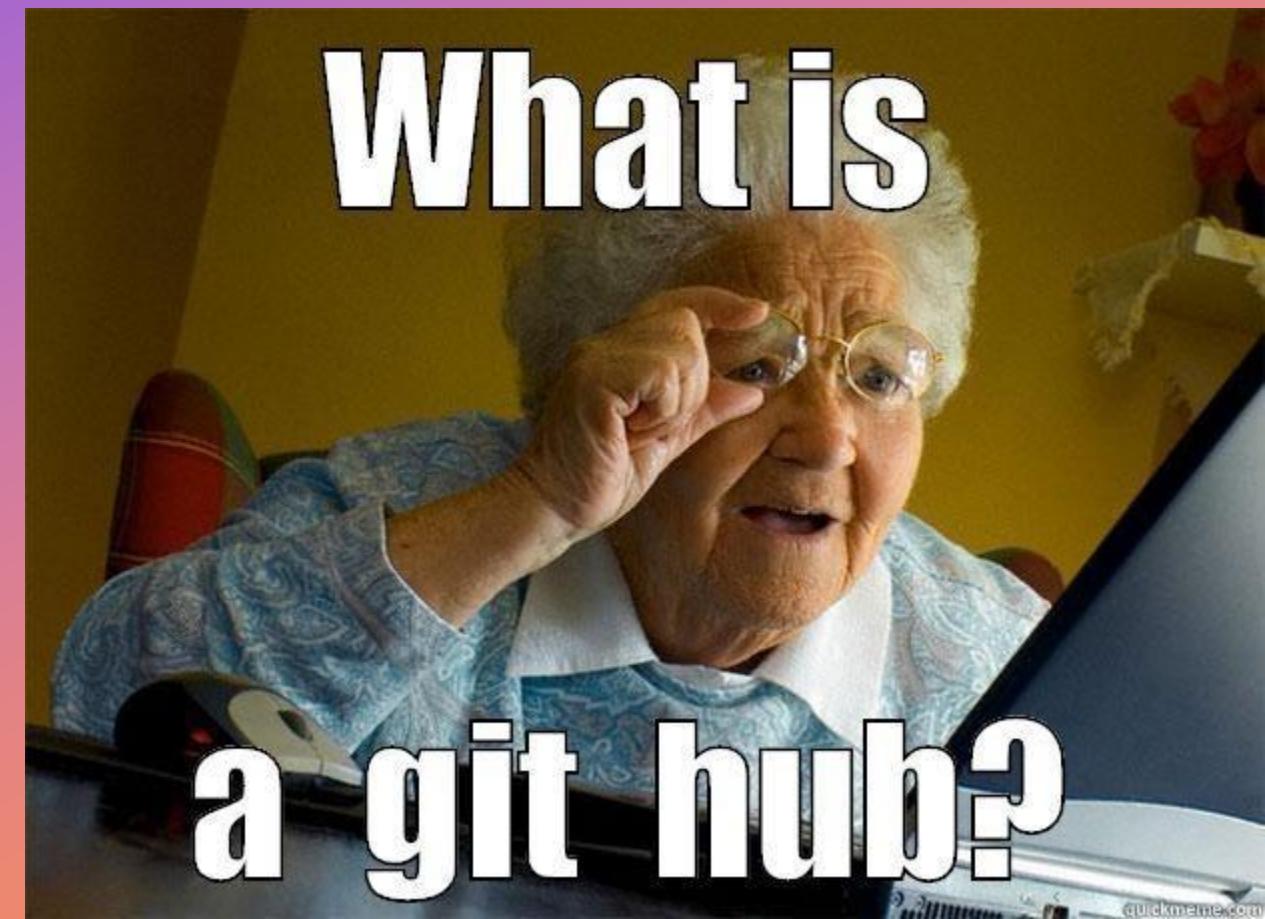


GIT & GITHUB

**What is
a git hub?**



ЛИНУС ТОРВАЛЬДС

инженер-программист, наиболее известный как создатель и ведущий разработчик ядра операционной системы **Linux** и распределенной системы управления версиями **Git**.

Ключевые достижения:

- Linux: В 1991 году
- Git: В 2005





+

.

ЧТО ТАКОЙ GIT?

Git – это система контроля версий

Это «машина времени» для ваших проектов.

Она сохраняет «снимки» (версии) ваших файлов с течением времени, поэтому вы никогда не потеряете свою работу и сможете вернуться к предыдущим версиям.



Проект_финал.docx



Проект_финал_v2.docx



Проект_Точно_финал.docx



+ .

ИСПОЛЬЗУЯ GIT

- Возможность вернуться к старым версиям, если что-то сломалось
- Видно, кто, что и когда изменил
- Возможность совместной работы без хаоса.



Проект_финал.docx



GIT И GITHUB



- **Инструмент** на вашем компьютере
- Программа командной строки, которая выполняет фактическое отслеживание версий



- **Веб-сайт**, на котором размещаются ваши проекты в интернете.
- Платформа для обмена и совместной работы



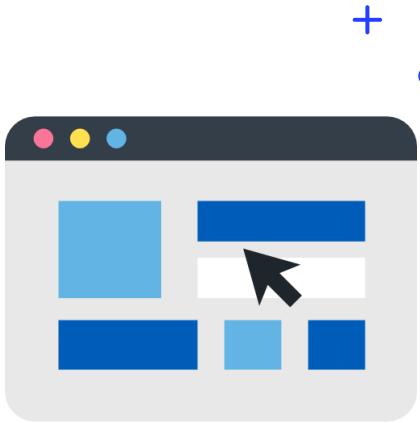
КАК ИСПОЛЬЗОВАТЬ GIT?



BASH CLI

это одна из самых популярных **оболочек командной строки (CLI)** в мире Linux и macOS.

Она служит переводчиком между вами и операционной системой, позволяя управлять компьютером, вводя текстовые команды.



GUI (GRAPHICAL USER INTERFACE)

это графический пользовательский интерфейс.

В отличие от командной строки, где вы вводите текст, GUI позволяет вам взаимодействовать с компьютером с помощью визуальных элементов: **окон, иконок, кнопок, меню и курсора мыши**.



СЛОВАРЬ НАЧИНАЮЩЕГО ГИТАРИСТА

Репозиторий — это, по сути, папка проекта, за которой присматривает Git.

Коммит (commit) — это «сохранение» проекта в определённый момент.

Ветка (branch) — это такая параллельная реальность проекта.

Пуш (push) и Пул (pull) — это команды для синхронизации с удалённым репозиторием.

- **push** — отправить свои коммиты,
- **pull** — забрать свежие изменения.



КОМАНДЫ CLI BASH

- **ls** — показать содержимое текущей папки (список файлов и папок).
- **ls -a** — показать все файлы, включая скрытые (те, что начинаются с точки).
- **cd <путь_к_папке>** — сменить текущую папку (перейти в другую).
- **cd ..** — перейти на один уровень вверх.
- **pwd** — показать полный путь к текущей папке.
- **mkdir <имя_папки>** — создать новую папку.



КОМАНДЫ CLI GIT BASH (2)

- **touch <имя_файла>** — создать новый пустой файл.
- **cp <что_копировать> <куда_копировать>** — скопировать файл или папку.
- **rm <имя_файла>** — удалить файл. **Осторожно, это действие необратимо!**
- **rm -r <имя_папки>** — удалить папку со всем её содержимым.
- **cat <имя_файла>** — вывести всё содержимое файла прямо в терминал.
- **less <имя_файла>** — открыть файл для постраничного просмотра (удобно для больших файлов). Выход — клавиша q.
- **clear** — очистить экран терминала.



НАСТРОЙКА GIT

- `git config --global user.name "Ваше Имя"`
- `git config --global user.email "vash.email@example.com"`

git init

Создание локального git для отслеживание проекта



git add filename

Эта команда помещает файлы в так называемую «область подготовки» (staging area)



```
git restore --staged filename
```

Это команда отменяет добавление файлов из области подготовки



```
git commit -m "first commit"
```

Это "**снимок**" или точка сохранения состояния проекта, ключ **-m** это сообщение снимка.



```
git remote add origin url
```

Это команда для связывания вашего локального репозитория с удалённым (например, на GitHub)



```
git push -u origin main
```

Отправляем изменения на GitHub



git push

Отправляем изменения на GitHub



git pull

Обновляем основную ветку