

# Лабораторной работе №1

---

Выполнил: Терентьев Егор Дмитриевич 1032192875 НФИбд-03-19

## Задание: Познакомиться с основными возможностями Git и Markdown

**Git** — это распределенная система управления версиями. Это означает, что локальный клон проекта является полным репозиторием управления версиями. Полнофункциональные локальные репозитории упрощают работу в автономном режиме или в удаленном расположении. Разработчики фиксируют свою работу локально, а затем синхронизируют свою копию репозитория с копией на сервере.

### Основные команды **Git**:

- Для начала стоит установить имя и электронную почту с помощью команд:
  - `git config --global user.name "Your Name"`
  - `git config --global user.email "your_email@whatever.com"`
- Также установить оканчание строк:
  - `git config --global core.autocrlf true`
  - `git config --global core.safecrlf true`
- Для установки unicode воспользуемся:
  - `git config --global core.quotePath off`
- Чтобы создать репозиторий пишем команду:
  - `git init`
- Чтобы проверить текущее состояние репозитория, воспользуйтесь:
  - `git status`
- Чтобы проиндексировать изменения:
  - `git add "file_name"`
- Чтобы зафиксировать добавленные в индекс изменения:
  - `git commit -m "comment"`
- Чтобы получить список произведенных изменений:
  - `git log`
  - `git log --pretty=oneline`
  - `git log --pretty=oneline --max-count=2`
  - `git log --pretty=oneline --since='5 minutes ago'`
  - `git log --pretty=oneline --until='5 minutes ago'`
  - `git log --pretty=oneline --author=`
  - `git log --pretty=oneline --all`
- Чтобы переключиться между ветками или извлечь файлы:
  - `git checkout "name"`
- Чтобы создать тег:
  - `git tag "name"`
- Чтобы выполнить сброс:
  - `git reset`
- Чтобы отменить коммит:
  - `git revert`

- Для перемещения:
  - `git mv "название файла" "путь куда переместить"`
- Чтобы получать содержимое git объекта:
  - `git cat file`
  - При помощи флага "-t" можем узнать тип (type), а "-p" покажет содержимое
- Чтобы создать ветку:
  - `git branch "file_name" or git checkout -b "file_name"`
- Чтобы слить ветки:
  - `git merge "branch_name"`
- Чтобы сделать перебазирование:
  - `git rebase "branch_name"`
- Чтобы создать клон репозитория:
  - `git clone "name_repos_to_clone" "name_cloned_repos"`
- Чтобы посмотреть на ветки, доступные в нашем репозитории:
  - `git branch`
- Чтобы добавить ветку наблюдения:
  - `git branch --track`
- Для создания чистого репозитория, то есть репозитория, в котором нет рабочих каталогов, воспользовались командой
  - `git clone --bare.`

## Заключение:

Git — это мощная система управления версиями и после выполнения данной лабораторной работы, я смог познакомиться с ней поближе и освоить основные возможности работы с ней