

# Лабораторная работа №1. Работа с git

**Выполнил: Попов Дмитрий Павлович НФИбд-03-19 1032191646**

## Задание

Необходимо ознакомиться с основными возможностями git, а также научиться работать с markdown

## Команды git

- Чтобы создать git репозиторий из пустого каталога, выполнили команду `git init`.
- Использовали команду `git status`, чтобы проверить текущее состояние репозитория.
- Выполнили команду `git add`, чтобы проиндексировать изменения. Это означает, что git теперь знает об изменении, но изменение пока не записано в репозиторий.
- Чтобы сделать коммит воспользовались командой `git commit -m`. Метка `-m` для комментария в командной строке.
- Получили список произведенных изменений командой `git log`.
- Для возвращения назад в историю использовали команду `checkout`, которая копирует любой снимок из репозитория в рабочий каталог.
- Создал тег первой версии командой `git tag v1`, для проверки доступных тегов использовал команду `git tag`.
- Для отмены проиндексированных изменений (перед коммитом) команда `git reset`, которая сбрасывает буферную зону к HEAD. Это очищает буферную зону от изменений, которые мы только что проиндексировали.
- Для отмены коммита, можно создать новый коммит, который удаляет изменения, сохраненные нежелательным коммитом. Воспользовались командой `git revert HEAD`.
- При получении ссылки на коммит (т.е. хэш, ветка или имя тега), команда `git reset`:
  - перепишет текущую ветку, чтобы она указывала на нужный коммит;
  - опционально сбросит буферную зону для соответствия с указанным коммитом;
  - опционально сбросит рабочий каталог для соответствия с указанным коммитом.
- Мы использовали команду `git reset --hard v1`. Параметр `--hard` указывает, что рабочий каталог должен быть обновлен в соответствии с новым head ветки.
- Чтобы удалить тег и коммиты, на которые он ссылался, воспользуемся сборщиком мусора `git tag -d oops`.
- Чтобы изменить предудущий коммит воспользовались командой `git commit --amend`.
- Перемещали файлы командой `git mv`. Она удаляет файл из начального каталога, создает этот файл в новом каталоге и эти факты сразу проиндексированы и готовы к коммиту.
- Вся информация git находится в каталоге `.git`. Просмотреть можно командой `dir -C .git`.
- С помощью команды `git cat-file` можно выводить определенные коммиты или каталоги по их хэшу.
- Для создания новой ветки использовали команду `git checkout -b style`, где `style` название новой ветки.
- Для слияния, то есть переноса изменений из двух веток в одну, использовали `git merge`.

- При возникновении конфликтов в изменениях, разрешали конфликты вручную. Также использовали команду перебазирования `gebase` вместо `merge`. Результат выполнения похож на результат слияния, но отличается дерево коммитов.
- Для создания клонированного репозитория воспользовались командой `git clone`.
- Чтобы посмотреть ветки в репозитории воспользовались командой `git branch`. Для отображения удаленных веток добавили метку `-a`.
- Для извлечения новых коммитов из удаленного репозитория использовали команду `git fetch`. Но она не сливает их с наработками в локальных ветках.
- Для извлечения новых коммитов из удаленного репозитория и сливания изменения, воспользовались командой `git pull`.
- Для отслеживания удаленной ветки, можем добавить локальную ветку командой `git branch --track`.
- Для создания чистого репозитория, то есть репозитория, в котором нет рабочих каталогов, воспользовались командой `git clone --bare`.
- Чтобы добавить репозиторий к оригинальному репозиторию, воспользовались командой `git remote add shared`.
- Для отправки изменений в общий репозиторий использовали команду `git push shared master`.

## Вывод

Ознакомился с основными возможностями `git`, а также научился работать с `markdown`