# Обзор базовых операций GIT

## Инициализация git-репозитория

Для инициализации git-репозитория необходимо перейти в директорию папки, в которой вы хотите создать репозиторий и написать в cmd следующее: ***«git init»***



Рисунок 1 Инициалация git-репозитория

В директории появится скрытая папка .git:



Рисунок 2 Скрытая папка .git, появляющаяся в директории после инициализации репозитория

## Получение статуса git-репозитория

Получения статуса репозитория на стандартной ветке master осуществляется при помощи команды ***«git status»***

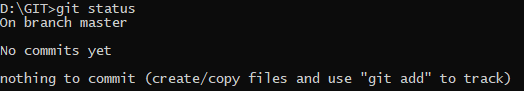


Рисунок 3 Получение статуса пустого репозитория

## Взаимодействие с файлами и изменение их статуса

Создание новой папки readme.txt с текстом: “Hello!”



Git автоматически обнаруживает новые файлы в директории репозитория:

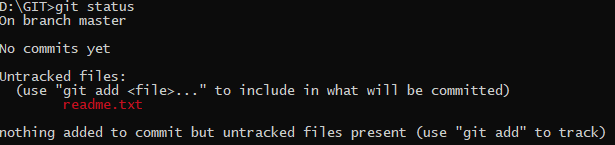


Рисунок 4 Состояние untracked созданного файла readme.txt

Для отслеживания файла необходимо ввести команду: ***«git add (название файла)»***



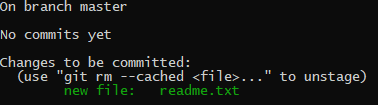


Рисунок 5 Новый статус readme.txt – отслеживаемый

Для фиксации состояния репозитория используется команда: ***«git commit -m (комментарий)»***



Рисунок 6 Фиксация состояния репозитория с комментарием First file



Рисунок 7 Состояние репозитория

При модификации файла git это определить и присвоит файлу статус modified:



Рисунок 8 Изменение содержания файла readme.txt

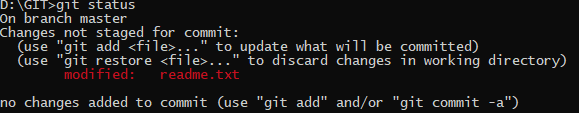


Рисунок 9 Состояние репозитория: файл readme.txt был модифицирован

Для отслеживания нескольких файлов одинакового формата (например txt) можно использовать команду: ***«git add „\*.формат” »***

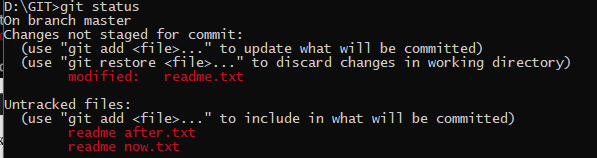


Рисунок 10 Созданные копии файла readme.txt файлы readme now.txt и readme after.txt



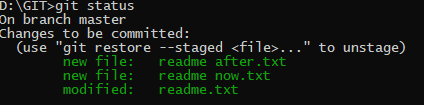


Рисунок 11 Состояние "отслеживание" для всех txt файлов после git add “\*.txt"

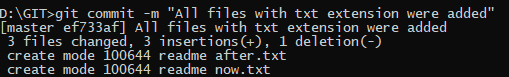
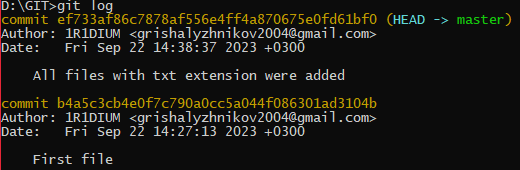
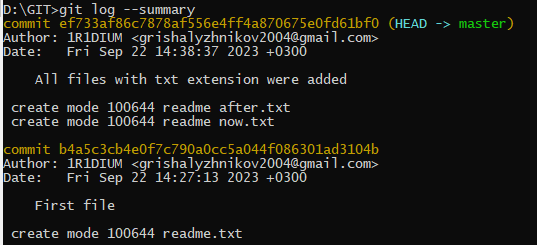


Рисунок 12 Фиксация состояния репозитория после добавления новых файлов

Просмотр историй фиксаций репозитория можно при помощи команд: ***«git log»*** и ***«git log –summary»*** (более подробно). **История фиксаций отражена в обратном порядке, т.е. самая первая фиксация будет в самом низу.**





## Взаимодействие с удаленным репозиторием

Для того, чтобы «привязать» локальный репозиторий к удалённому, необходимо ввести команду: ***«git remote add (Произвольное название удаленного репозитория) (ссылка на удаленный репозиторий на github)»***



Рисунок 13 «Привязка» локального репозитория к удаленному

Для загрузки файлов из локального репозитория в удаленный, необходимо ввести команду: ***«git push -u (название удаленного репозитория) (ветку, которую надо загрузить)»***

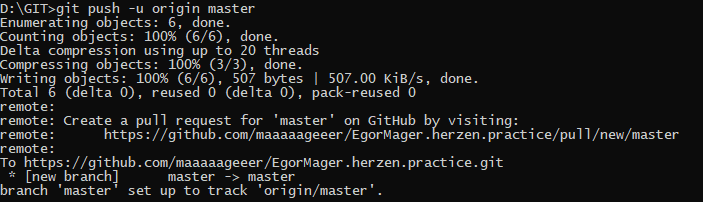


Рисунок 14 Загрузка ветки master в удаленный репозиторий



Рисунок 15 Загруженные файлы из локального репозитория на сайте github.com

Для загрузки с удаленного репозитория на локальный репозиторий, необходимо ввести команду: ***«git pull (название удаленного репозитория) (ветка, куда загрузить)»***

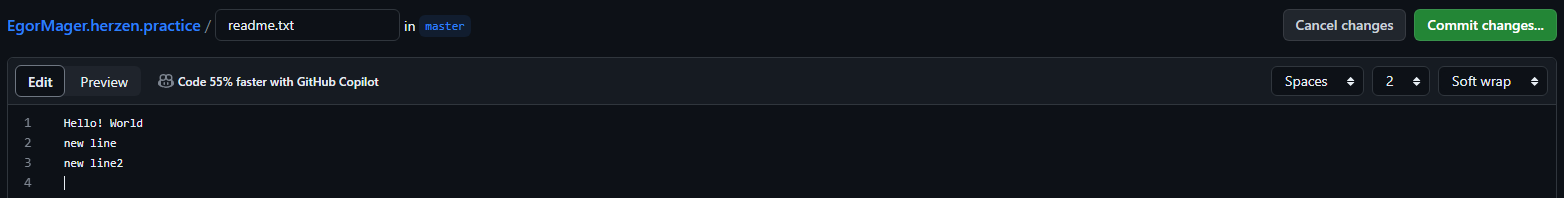


Рисунок 16 Изменение информации в файле readme.txt на удаленном репозитории

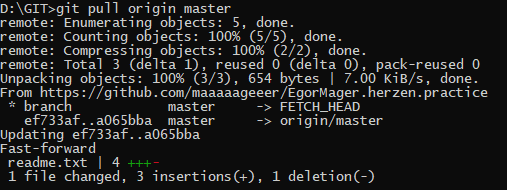


Рисунок 17 Загрузка новой версии ветки master с удаленного репозитория

## Просмотр изменений в файле

Чтобы посмотреть изменения в файле, нужно ввести команду: ***«git diff HEAD»*** , где HEAD – последнее состояние текущей ветки (master)

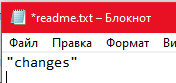


Рисунок 18 Изменение файла readme.txt

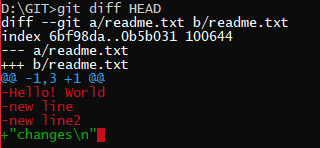


Рисунок 19 Просмотр изменений в ветке master

## Взаимодействие с каталогами

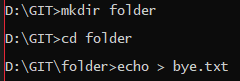


Рисунок 20 Создание каталога c файлом bye.txt

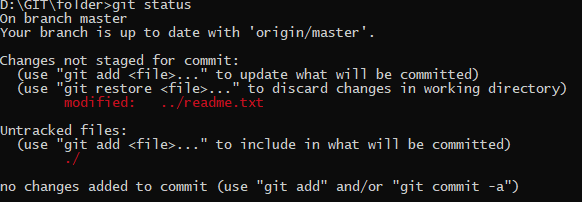


Рисунок 21 Статус репозитория: каталог не отслеживается



Рисунок 22 Отслеживание файла в каталоге

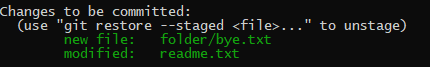


Рисунок 23 Состояние репозитория после отслеживания

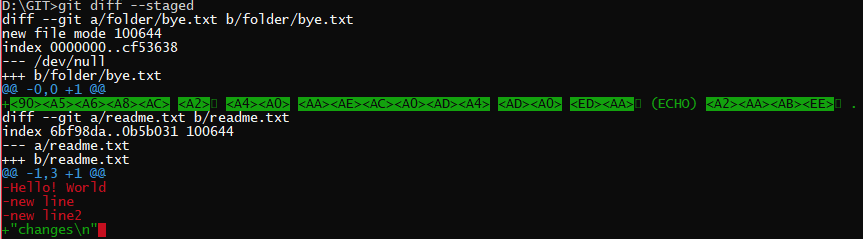


Рисунок 24 Более расширенное сравнение состояний репозитория при помощи «git diff --staged»

## Удаление файла из ветки репозитория

Для удаления файла из ветки нужно ввести команду: ***«git reset (файл)»***



Рисунок 25 Удаление файла bye.txt из ветки репозитория master

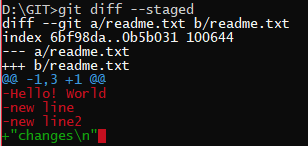


Рисунок 26 Сравнение состояний репозитория

## Отмена изменений

При случайном удалении или изменении файла можно использовать команду: ***«git checkout -- (имя файла)»***

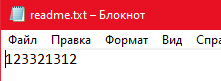


Рисунок 27 «Случайное» изменение содержимого файла



Рисунок 28 Отмена изменений в файле

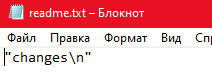


Рисунок 29 Восстановленное содержимое файла

## Создание и взаимодействие с ветками

Для того, чтобы создать новую ветку, необходимо ввести: ***«git branch (название новой ветки)»***



Рисунок 30 Создание новой ветки «clean\_up»

Для просмотра текущей ветки нужно ввести: ***«git branch»***



Рисунок 31 Просмотр текущей ветки (master)

Для переключения ветки введите: ***«git checkout (название существующей ветки)»***

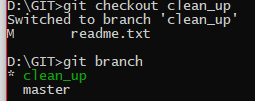


Рисунок 32 Переключение ветки (текущая теперь clean\_up)

Для удаления файлов из отслеживаемых нужно использовать команду: ***«git rm (название файла)»***

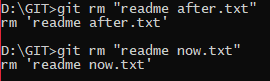


Рисунок 33 Исключение из отслеживаемых файлов readme after.txt и readme now.txt

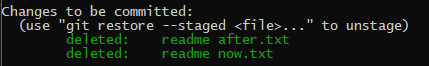


Рисунок 34 Статус репозитория после исключения двух файлов из отслеживаемых

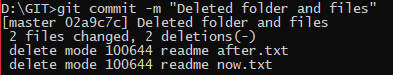


Рисунок 35 Фиксация изменений в ветке clean\_up

Для слияния веток нужно ввести: ***«git merge (название ветки)»***

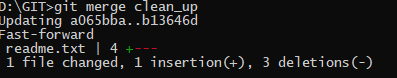


Рисунок 36 Слияние веток master и clean\_up

Удаление ветки происходит при помощи команды: ***«git branch -d (название ветки)»***



Рисунок 37 Удаление ветки clean\_up

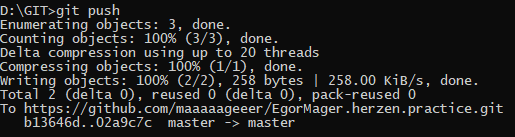


Рисунок 38 Загрузка изменнённой ветки master, которая содержит один файл readme.txt



Рисунок 39 Обновленное состояние удаленного репозитория