# Инструкция по рабое с git

## Что такое git

\*\*\*Git\*\*\* -самая популярная реализация распределенной системы контроля версий (версионность поддерживается и на сервере и у каждого клиента). Самой распространенной реализацией \*\*\*Git\*\*\* является (\*GitHub\*)[https://github.com]

## Подготовка репозитория

Для создания репозитория используется команда \*git init\*. Для этого необходимо открыть в терминале папку с будущим репозиторием и написать \*git init\*.

## Создание коммитов

### Добавление файлов к коммиту

Для добавления файла к новому коммиту используется команда \*git add\* Используется она следующим образом: в терминале с папкой-репозиторием пишем \*git add<название файла>.

### Создание коммита

Для создания новой фиксации (коммита) используется команда \*git commit\*. Для этого в терминале с папкой-репозиторием пишем \*git commit -m "<сообщение к коммиту>". Сообщение к коммиту писать \*\*\*ОБЯЗАТЕЛЬНО!!!\*\*\*

## Перемещение между коммитами

Для перемещения на другую фиксацию (коммит) используется команда \*git checkout\*. Для этого необходимо, как показано в прошлом пункте, в журнале изменений найти необходимый коммит и его хеш (номер), после чего в терминале с папкой-репозиторием надо написать \*git checkout <хеш коммита>. После выполнения этой команды мы попадаем в состояние \*\*detached head\*\* в котором никакие следующие коммиты сохраняться не будут. Для выхода из этого состояния необходимо написать \*git checkout master\*.

## Сравнение изменений

Для просмотра изменений используется команда \*git diff\*. Для этого в терминале с папкой-репозиторием пишем \*git diff\*, в терминале будут выведены сделанные изменения.

## Журнал изменений

Для просмотра истории изменений используется команда \*git log\*. Для этого в терминале с папкой-репозиторием необходимо написать \*git log\*.

## Ветки в git

### Создание ветки в git

Для создания новой ветки используется команда "git branch". Для этого в терминале с папкой-репозиторием необходимо написать "git branch <название ветки>.

### Просмотр списка веток

Для просмотра списка веток используетс я команда "git branch". Для этого в терминале с папкой-репозиторием необходимо написать \*git branch\*. Выделенная зеленым со звездочкой ветка - это ветка на которой мы находимся в данный момент.

## Перемещение между ветками.

Для перехода на другую ветку используется команда "gut chekout" Для этого в терминале с папкой-репозиторием необходимо написать "git chekout <название ветки>. Такая ветка должна \*\*\*существовать\*\*\*.

### Слияние веток и разрешение конфликтов

Перед слиянием переходим на ветку мастер, введением команды \*git checkout master\*. Для слияния веток используем команду \*git merge\*. Для этого в терминале папки-репозитория необходимо написать \*git merge <название ветки>. Фиксируем изменения с помощью команды \*git add\* и далее \*git commit\*

### Удаление веток

Для удаления ненужной ветки после слияния используется команда \*git branch -d <название ветки>.

## Работа с удаленным репозиторием

### Подключение к удаленному репозиторию

Чтобы загрузить что-нибудь в удаленный репозиторий, сначала нужно к нему подключиться. Регистрация и установка может занять время, но все подобные сервисы предоставляют хорошую документацию.

Чтобы связать наш локальный репозиторий с репозиторием на GitHub, выполним следующую команду в терминале "git remote add origin <ссылка на репозиторий в GitHub>.

### Отправка изменений на сервер

Пересылаем наш локальный коммит на сервер. Этот процесс происходит каждый раз, когда мы хотим обновить данные в удаленном репозитории.

Команда, предназначенная для этого "git push origin master".