

Справочник основных команд GIT.

- **git init**

Эта команда используется для создания GIT репозитория.

Пример использования:

```
git init
```

- **git config**

Одна из самых популярных команд, используется для указания пользовательских настроек, эл.почта, имя пользователя и т.д.

Например, данная команда используется для установки имени пользователя:

```
git config --global user.name имя пользователя
```

- **git clone**

Используется для клонирования репозитория

Пример:

```
git clone <URL репозитория> (если находится на удаленном сервере)
```

```
git clone /путь/до/репозитория (если репозиторий локальный)
```

- **git add**

Используется для добавления файла в индекс

Например, следующая команда добавит файл под названием file.txt присутствующий в локальном каталоге в индекс:

```
git add file.txt
```

- **git status**

Отображает список измененный файлов + еще не добавленные или ожидающие коммита

Пример:

```
git status
```

- **git push**

Одна из главных команд. Позволяет поместить изменения в главную ветку удаленного хранилища связанного с рабочим каталогом. Например:

```
git push origin master
```

- **git branch**

Может быть использована для отображения, создания или удаления веток. Для отображения всех существующих веток в репозитории введите:

```
git branch
```

Для удаления ветки:

```
git branch -d
```

- **git checkout**

Может быть использована для создания веток или переключения между ними.

К примеру, следующий код создаст новую ветку и переключится на нее:

```
git checkout -b
```

Чтобы просто переключиться между ветками используйте:

git checkout

- **git commit**

Используется для коммита изменений в файлах проекта.

Применение:

git commit -m "Сообщение идущее вместе с коммитом"

- **git pull**

Используется для объединения изменений, присутствующих в удаленном репозитории, в локальный рабочий каталог.

Применение:

git pull

- **git merge**

Используется для объединения ветки в активную ветвь.

Применение:

git merge

- **git log**

Отобразит список всех коммитов в ветке вместе с соответствующими сведениями.

Пример использования:

git log

commit 15f4b6c44b3c8344caasdac9e4be13246e21saw

Author: Iven Ivanov <iveni@gmail.com>

Date: Tue Sep 12 11:00:00 2022 -0600

- **git tag**

Используется для маркировки определенных коммитов с помощью простых меток. Примером может быть эта команда:

git tag 1.1.0 <вставьте ключевое слово>

- **git remote**

Позволяет пользователю подключиться к удаленному репозиторию.

Данная команда отобразит список удаленных репозиториях, настроенных в данный момент:

```
git remote -v
```

Позволит пользователю подключить локальный репозиторий к удаленному серверу:

```
git remote add origin <адрес удаленного сервера>
```

- **git diff**

Используется для выявления различий между ветками.

Для выявления различий с базовыми файлами, используйте

```
git diff --base
```

Следующая команда используется для просмотра различий между ветками, которые должны быть объединены, до их объединения:

```
git diff <ветвь – источник> <ветвь – цель>
```

Для простого отображения существующих различий, используйте:

```
git diff
```

- **git reset**

Используется для сброса индекса и рабочего каталога до последнего состояния коммита. Применение:

```
git reset --hard HEAD
```

- **git rm**

Используется для удаления файлов из индекса и рабочего каталога.

Применение:

git rm имяфайла.txt

- **git stash**

Одна из самых малоизвестных команд git. Она помогает в сохранении изменений на временной основе, эти изменения не попадут в коммит сразу.

Применение:

git stash

- **git show**

Используется для просмотра информации о любом git объекте. Для примера:

git show

- **git fetch**

Позволяет пользователю доставить все объекты из удаленного репозитория, которые не присутствуют в локальном рабочем каталоге.

Пример:

git fetch origin

- **git gc**

Используется для оптимизации репозитория, удаляет и оптимизирует ненужные файлы

Пример:

```
git gc
```