

ОСНОВНЫЕ КОМАНДЫ GIT

git config	
Предназначение	Команда предназначена для настройки параметров <i>Git</i> на вашем компьютере. Она устанавливает значение соответствующего параметра в конфигурации <i>Git</i> .
Формат	<code>git config <ключ> <параметр> <значение></code>
Основные ключи/флаги	<code>--global</code> Отвечает за изменение параметров конфигурации на уровне пользователя. Без указания данного ключа настройки будут изменены только на уровне текущего репозитория. <code>--system</code> Изменение настроек на уровне системы (то есть сразу для всех пользователей).
Параметры	<code>user.name</code> — имя пользователя. <code>user.email</code> — электронная почта пользователя.
Пример	<pre># Задаём значение имени на уровне репозитория \$ git config user.name Skillfactory # Просматриваем значение параметра \$ git config user.name Skillfactory # Задаём значение email глобально \$ git config --global user.email data_science@skillfactory.ru</pre>
git init	
Предназначение	Инициализирует локальный репозиторий. По сути, она создаёт пустой репозиторий на вашем компьютере. Инициализация репозитория — это создание в текущей директории новой поддиректории с именем <code>.git</code> , содержащей все необходимые файлы репозитория — структуру <i>git</i> -репозитория.

Формат	<code>git init <параметры></code>
Пример	<pre># Создаём пустой репозиторий в папке проекта \$ cd projects/project_1/ \$ git init Initialized empty Git repository in projects/project_1/</pre>
git add	
Предназначение	Добавляет файл (папку с файлами) в индекс (иногда говорят «индексирует»), то есть добавляет его в список отслеживаемых для системы контроля версий. Нужно указать в аргументах, какой файл или папку мы хотим добавить.
Формат	<code>git add <имя файла> <ключи></code>
Основные ключи/флаги	-A Указав этот ключ, мы сообщаем <i>Git</i> , что под действие команды попадут все файлы в репозитории.
Пример	<pre># Добавляем файл README.md в индекс \$ git add README.md # Добавляем папку data и всё её содержимое в индекс \$ git add data/ # Добавляем все файлы из текущей директории в индекс \$ git add . # Добавляем все изменённые файлы в индекс \$ git add -A</pre>
git commit	
Предназначение	Создаёт новый коммит с файлами из индекса.
Формат	<code>git commit <флаги></code>

Основные ключи/флаги	<p><code>-m <описание></code> Позволяет написать сообщение вместе с командой, не открывая текстовый редактор.</p> <p><code>-a</code> Переносит все отслеживаемые файлы в область подготовленных файлов и включает их в коммит.</p> <p><code>--amend</code> Заменяет последний коммит новым изменённым коммитом, что бывает полезно, если вы неправильно набрали сообщение последнего коммита или забыли включить в него какие-то файлы.</p>
Пример	<pre># Создаём коммит с комментарием "fixed bug in function clean_data" \$ git commit -m "fixed bug in function clean_data" # Ранее внесли изменения в отслеживаемые файлы; # создаём коммит, не добавляя изменения в индекс \$ git commit -a -m "fixed bug in function clean_data"</pre>
git status	
Предназначение	Позволяет отследить состояние файлов в репозитории и узнать, какие изменения необходимо зарегистрировать Git (при необходимости — отменить).
Формат	<code>git status <флаги></code>
Основные ключи/флаги	<p><code>-s</code> Выводит информацию в сокращённом формате (<i>short-format</i>).</p>
Пример	<pre># Выводим информацию о текущем состоянии файлов в # сокращённом виде \$ git status -s</pre>
git log	
Предназначение	<p>Показывает список последних коммитов и их хеши.</p> <p>Список выводится, начиная с последнего коммита.</p>
Формат	<code>git log <ключи></code>

Основные ключи/флаги	<p>-<число> или -n <число></p> <p>Вывести только последние несколько коммитов. Их число вы указываете в параметре к этому ключу.</p> <p>--pretty=<значение></p> <p>Красивый вывод истории. Доступные значения: <code>oneline</code>, <code>short</code>, <code>medium</code>, <code>full</code> и др.</p>
Пример	<p>#Выводим информацию о двух последних коммитах</p> <pre>\$ git log -n 2</pre>
<code>git revert</code>	
Предназначение	Создаёт новый коммит, который отменяет изменения, внесённые в переданном коммите (последовательности коммитов). То есть для отмены изменений создаётся новый коммит, который «стирает» тот, который мы укажем в аргументах.
Формат	<code>git revert <ключи> <адрес коммита> <флаги></code>
Основные ключи/флаги	<p>--no-edit</p> <p>Позволяет не открывать текстовый редактор для сообщений коммита. Комментарий для перезаписывающего коммита формируется автоматически.</p>
Пример	<p>#Отменяем изменения в коммите 62aa</p> <pre>\$ git revert 62aa --no-edit</pre> <p>#Отменяем изменения в текущем коммите</p> <pre>\$ git revert HEAD --no-edit</pre> <p>#Отменяем изменения в предыдущем коммите</p> <pre>\$ git revert HEAD~1 --no-edit</pre>
<code>git branch</code>	
Предназначение	<p>Многофункциональная команда. Основные варианты применения:</p> <ul style="list-style-type: none"> → создаёт новую ветку репозитория (по умолчанию — без использования ключей); → удаляет существующую ветку; → выводит список веток.

Формат	<code>git branch <ключи> <ИМЯ ВЕТКИ></code>
Основные ключи/флаги	<p><code>-D</code> Этот ключ используется для удаления указанной ветки.</p> <p><code>-r</code> От англ. remote — «удалённый». С этим ключом будут выведены только удалённые ветки.</p> <p><code>-a</code> От англ. all — «все». С этим ключом будут выведены и локальные, и удалённые ветки.</p>
Пример	<pre># Создаём новую ветку с именем develop git branch develop # Удаляем ветку с именем develop git branch -D develop # Выводим локальные и удалённые ветки \$ git branch -a # Выводим только локальные ветки \$ git branch # Выводим только удалённые ветки \$ git branch -r</pre>
git checkout	
Предназначение	Перемещает указатель <i>HEAD</i> на указанную позицию. Позволяет переключаться между ветками. Вместо имени ветки можно передать хеш коммита из текущей ветки, чтобы перемещаться коммитами.
Формат	<code>git checkout <имя ветки (хеш коммита)></code>
Основные ключи/флаги	<p><code>-b</code> При передаче этого ключа команда автоматически создаст новую ветку, а затем переключится на неё.</p>

Пример	<pre># Создаём ветку develop и переключаемся на неё \$ git branch develop \$ git checkout develop # Создаём ветку develop и переключаемся на неё в одно действие \$ git checkout -b develop # Переключаемся на два коммита назад в текущей ветке \$ git checkout HEAD~2 # Переключаемся на коммит с хешем 62ab \$ git checkout 62ab</pre>
<u>git merge</u>	
Предназначение	Сливает изменения (вносит коммиты) из указанной ветки в текущую.
Формат	<code>git merge <сливаемая ветка></code>
Основные ключи/флаги	<pre>--ff</pre> <p>Слияние в режиме <i>fast-forward</i> (неявное слияние), используется по умолчанию.</p> <p>В случае НЕЯВНОГО СЛИЯНИЯ не создаётся никаких новых коммитов — используются только уже существующие. Идея такого слияния заключается в том, что из вливаемой ветки извлекается несколько коммитов, а затем они применяются к последнему коммиту целевой ветки.</p> <pre>--no-ff</pre> <p>Отключить режим слияния <i>fast-forward</i>.</p> <p>В случае ЯВНОГО СЛИЯНИЯ всегда создаётся новый так называемый <i>merge</i>-коммит, который «объединяет» изменения двух веток.</p>
Пример	<pre># Переключаемся на основную ветку \$ git checkout main # Сливаем изменения с ветки develop в ветку main \$ git merge develop</pre>

<u>git remote add</u>	
Предназначение	Подключает удалённый репозиторий к вашему под переданным именем.
Формат	<code>git remote add <имя удалённого репозитория> <ссылка на удалённый репозиторий></code>
Пример	<pre># Подключаем удалённый репозиторий с именем origin \$ git remote add origin https://github.com/SkillfactoryDS/DataCleaningProject.git</pre>
<u>git remote rename</u>	
Предназначение	Переименовывает удалённый репозиторий.
Формат	<code>git remote rename <старое имя удалённого репозитория> <новое имя удалённого репозитория></code>
Пример	<pre>#Переименовываем репозиторий origin в feature \$ git remote rename origin feature</pre>
<u>git remote remove</u>	
Предназначение	Отключает переданный удалённый репозиторий от вашего.
Формат	<code>git remote remove <имя удалённого репозитория></code>
Пример	<pre># Отключаем удаленный репозиторий с именем origin \$ git remote remove origin</pre>
<u>git remote show</u>	
Предназначение	Выводит список всех подключенных удалённых репозиториев. Если передано имя репозитория, то выводит информацию об этом репозитории.
Формат	<code>git remote show <имя удалённого репозитория></code>
Пример	<pre># Выводим список всех удалённых репозиториев \$ git remote show # Выводим информацию про репозиторий origin \$ git remote show origin</pre>

git fetch	
Предназначение	<p>Получает изменения из переданного удалённого репозитория.</p> <p>Если не был передан ни удалённый репозиторий, ни ключ <code>--all</code>, команда пытается получить изменения из репозитория с именем <code>origin</code>.</p>
Формат	<code>git fetch <ключи> <имя удалённого репозитория></code>
Основные ключи/флаги	<p><code>--all</code></p> <p>Получает изменения из всех подключенных удалённых репозиториях.</p>
Пример	<pre># Получаем изменения из удалённого репозитория с именем origin \$ git fetch origin</pre>
git pull	
Предназначение	<p>Получает изменения из переданного удалённого репозитория и обновляет рабочую копию в соответствии с удалённым репозиторием.</p> <p>По умолчанию слияние удалённой ветки с локальной происходит именно в <i>fast-forward</i>-режиме, так что включать его специально не требуется.</p> <p>Является аналогом объединения команд <code>git fetch</code> и <code>git merge</code>.</p>
Формат	<code>git pull <ключи> <имя удалённого репозитория></code>
Основные ключи/флаги	<p><code>--ff</code></p> <p>Слияние в режиме <i>fast-forward</i>, используется по умолчанию.</p> <p><code>--no-ff</code></p> <p>Отключить режим слияния <i>fast-forward</i>.</p>
Пример	<pre># Обновляем нашу рабочую копию в соответствии с удалённым репозиторием origin \$ git pull origin</pre>

git push	
Предназначение	Производит отправку (пуш) локальных изменений из текущей ветки в удалённый репозиторий.
Формат	<code>git push <ключи> <имя удалённого репозитория> <имя ветки></code>
Основные ключи/флаги	<code>--all</code> Пушит все имеющиеся ветки.
Пример	<code># Загружаем все изменения с текущей ветки в удалённый репозиторий</code> <code>\$ git push origin develop</code>