Вот основные команды Git, разделённые по категориям:

**Настройка Git**

* git config --global user.name "Your Name" — указание имени пользователя.
* git config --global user.email "your.email@example.com" — указание email пользователя.

**Основные команды**

* git init — инициализация нового репозитория.
* git clone [url] — клонирование удалённого репозитория.
* git status — отображение статуса изменений.
* git add [file] — добавление файла в индекс (staging area).
* git commit -m "message" — создание коммита с сообщением.
* git push [remote] [branch] — отправка изменений на удалённый репозиторий.
* git pull — получение и слияние изменений из удалённого репозитория.

**Ветвление и работа с ветками**

* git branch — список веток.
* git branch [branch-name] — создание новой ветки.
* git checkout [branch-name] — переключение на другую ветку.
* git merge [branch-name] — слияние указанной ветки в текущую.
* git branch -d [branch-name] — удаление ветки.

**История и просмотр изменений**

* git log — журнал изменений.
* git log --oneline — сокращённый журнал изменений.
* git diff — просмотр изменений в несохранённых файлах.
* git diff [commit1] [commit2] — сравнение изменений между двумя коммитами.

**Работа с удалёнными репозиториями**

* git remote — список удалённых репозиториев.
* git remote add [name] [url] — добавление удалённого репозитория.
* git fetch — получение изменений без слияния.
* git push -u origin [branch] — отправка новой ветки на удалённый репозиторий и установка её как отслеживаемой.

**Откат изменений**

* git reset [file] — отмена добавления файла в индекс.
* git reset --hard [commit] — откат всех изменений до указанного коммита.
* git revert [commit] — создание нового коммита, отменяющего изменения предыдущего.

**Работа с тегами**

* git tag [tag-name] — создание тега на текущем коммите.
* git push origin [tag-name] — отправка тега в удалённый репозиторий.

Это основные команды, которые часто используются в работе с Git.

Чтобы создать файлы или папки с содержимым в Git Bash, используются стандартные команды Linux. Вот основные шаги:

**1. Создание папки**

* mkdir [folder-name] — создаёт новую папку.

Пример:

bash

Копировать код

mkdir my\_folder

**2. Переход в папку**

* cd [folder-name] — переход в папку.

Пример:

bash

Копировать код

cd my\_folder

**3. Создание файла**

* touch [file-name] — создаёт новый пустой файл.

Пример:

bash

Копировать код

touch file.txt

**4. Создание файла с содержимым**

* Используйте команду echo для создания файла с содержимым.

Пример:

bash

Копировать код

echo "Hello, world!" > file.txt

Это создаст файл file.txt с текстом "Hello, world!".

**5. Просмотр содержимого файла**

* cat [file-name] — отображение содержимого файла.

Пример:

bash

Копировать код

cat file.txt

**6. Редактирование файла**

* Используйте встроенный текстовый редактор, например, nano.

Пример:

bash

Копировать код

nano file.txt

После редактирования нажмите Ctrl + X, затем Y, чтобы сохранить изменения.

Теперь ты можешь добавлять эти файлы в Git с помощью стандартных команд, таких как git add, git commit, и так далее.

Чтобы создать папку в репозитории через Git Bash, выполните следующие шаги:

1. **Откройте Git Bash** и перейдите в корневую директорию вашего репозитория с помощью команды cd:

bash

Копировать код

cd /path/to/your/repository

1. **Создайте папку** в репозитории с помощью команды mkdir:

bash

Копировать код

mkdir new\_folder

1. **Добавьте в эту папку файл**, так как Git отслеживает только файлы, а не пустые папки. Создайте файл с помощью команды touch:

bash

Копировать код

touch new\_folder/example.txt

1. **Добавьте папку и файл в индекс Git**:

bash

Копировать код

git add new\_folder/example.txt

1. **Создайте коммит** для фиксации изменений:

bash

Копировать код

git commit -m "Добавлена новая папка с файлом"

1. **Отправьте изменения** в удалённый репозиторий (если нужно):

bash

Копировать код

git push origin main

Теперь новая папка будет отслеживаться Git и отправлена в удалённый репозиторий.