Теперь, когда мы начали работать над сценарием, давайте превратим каталог колдунов в код проекта Git. Мы делаем это с помощью:

git init

Слово init означает инициализацию. Команда устанавливает все инструменты, необходимые Git для начала отслеживания изменений, внесенных в проект.

In the terminal, initialize a new Git project.

Notice the output:

Initalized an empty git repository in /home/ccuser/workspace/sorcerers-code/.git/

The Git project was created. Click Next to continue.

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

checkout Switch branches or restore working tree files

commit Record changes to the repository

diff Show changes between commits, commit and working tree, etc

merge Join two or more development histories together

rebase Forward-port local commits to the updated upstream head

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some

concept guides. See 'git help <command>' or 'git help <concept>'

to read about a specific subcommand or concept.

rm Удалить файлы из рабочего дерева и из индекса изучите историю и состояние (см. также: git help revisions)    bisect Используйте бинарный поиск, чтобы найти фиксацию, введшую ошибку    grep Печатать строки, соответствующие шаблону    log Показать журналы фиксации    show Показать различные типы объектов    status Показывать статус рабочего дерева расти, маркировать и настраивать свою общую историю    ветвь Список, создание или удаление ветвей    checkout Переключение ветвей или восстановление рабочих файлов дерева    commit Запись изменений в репозиторий    diff Показать изменения между коммитами, фиксацией и рабочим деревом и т. д.    merge Присоединяйтесь к двум или более историям разработки вместе    переадресация Переадресация в прямом порту локальная для обновленной восходящей головки    tag Создайте, перечислите, удалите или проверьте объект тега, подписанный с GPG сотрудничать (см. также: рабочие процессы git help)    fetch Загрузка объектов и ссылок из другого репозитория    pull Извлечь и интегрировать с другим репозиторием или локальной ветвью    push Обновить удаленные ссылки вместе со связанными объектами «git help -a» и «git help -g» доступны доступные подкоманды и некоторые концептуальные руководства. См. 'Git help <command>' или 'git help <concept>' для чтения о конкретной подкоманде или концепции.

**Внесение и оформление изменений в локальном репозитории**

1. Воспользовавшись командой git status можно узнать, на какой ветке (branch) репозитория вы сейчас находитесь, какие изменения присутствуют в вашей рабочей копии и другую информацию.  
   *Рабочей копией называется совокупность файлов в локальной папке репозитория за исключением служебных файлов.*
2. После внесения каких-либо изменений в рабочую копию их можно закоммитить в локальный репозиторий:
3. сначала нужная часть изменений подготавливается к коммиту с использованием команды git add %file\_path%
4. после чего производится коммит командой git commit  
   Использование команды без аргументов откроет текстовый редактор, где надо будет написать комментарий для коммита, коммит обязательно должен иметь комментарий. Другим вариантом задания комментария к коммиту является использование команды git commit -m "%commit\_message%"
5. Историю изменений можно посмотреть командой git log или git log --name-only. Если вся история изменений не умещается на экране, то можно пользоваться клавишами прокрутки на клавиатуре ("стрелочки", PgUp, PgDown), выход из режима просмотра изменений осуществляется нажатием клавиши "q".

**Загрузка локальных изменений в удаленный репозиторий**

После того, как были выполнены нужные локальные коммиты, изменения можно загрузить в удаленный репозиторий с помощью команды git push origin master. GIT клиент при этом запросит имя пользователя и пароль для доступа к github.  
Выполнение этой команды может закончиться с ошибкой, если в локально репозитории отсутствуют последние изменения, имеющиеся в удаленном репозитории. Для решения этой проблемы надо выполнить команду git pull, которая скачает последние изменения из удаленного репозитория и смержит их с вашими локальными правками, после чего можно повторить команду git push.