Зайдем в любую папку, например /tmp/, используя терминал:

cd /tmp/

Создадим там папку hello, в которой будет наш проект с помощью команды:

mkdir hello

Перейдем в нее:

cd hello

Инициализируем git-репозиторий:

git init

По итогу получим сообщение:

Initialized empty Git repository in /tmp/hello/.git/

которое означает, что мы успешно создали репозиторий. Теперь мы можем сохранять информацию об изменениях в файлах. Ура!

Проверим. Создадим файл **README.md** с помощью команды:

touch README.md

Зайдем в файл **README.md**:

nano README.md

И добавим строчку Hello world!.

Три состояния

Прежде чем перейти к следующей команде поговорим о состояниях, в которых могут быть файлы.

Каждый файл в git может быть в двух состояниях:

* *Отслеживаемые* - под версионным контролем, которые хранятся в последнем коммите. Это состояние делится на три подсостояния:
  + Зафиксированные (**committed**) - сохраненные в локальной базе;
  + Подготовленные (**staged**) - отмеченные для включения в следующий коммит;
  + Измененные (**modified**) - файлы, изменения которых не были зафиксированы;
* *Неотслеживаемые* - все остальные файлы.

Три секции

Git состоит их трех секций:

* *Git-директория (.git directory)* - та часть git'а, в которой хранятся все снимки, все метаданные проекта;
* *Рабочая директория (working directory)* - то, что вы видите, зайдя в свой проект. По сути рабочая директория - это текущий выбранный вами снимок, файлы которого вы можете менять.
* *Область подготовленных файлов (Staging Area, Индекс)* - тут хранится информация о файлах, которые попадут в следующий коммит.

Теперь введем команду, которая проверяет состояние файлов в репозитории:

git status

И получим следующие строчки:

Untracked files:

(use "git add <file>..." to include in what will be committed)

README.md

Git показал нам, что **README.md** находится в секции Untracked files. То есть **README.md** - **неотслеживаемый** файл (см. рисунок выше).

Чтоб добавить данный файл в следующий снимок, введите команду git add README.md.

Теперь при вызове git status мы увидим, что **README.md** добавлен в **Staging Area** - область подготовленных файлов для последующего коммита (см. рисунок выше):

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: README.md

Итак, мы добавили все необходимые изменения. Теперь мы хотим создать снимок. Вводим команду:

git commit

У вас откроется редактор, в котором вы можете ввести комментарий к коммиту. Старайтесь писать то, что было сделано в рамках данного коммита. В данном случае: Мы добавили файл README.md.

В итоге изменения внесенные нами стали зафиксированными и файл был добавлен в git-директорию (см. рисунок выше).

Отлично. Теперь давайте попробуем изменить наш файл **README.md**.

nano README.md

И добавим или изменим предыдущую строку: Привет Мир!.

В итоге при вызове git status мы получим следующий вывод:

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: README.md

Наш файл находится в измененном состоянии (см. рисунок выше).

Проделаем ту же последовательность команд: git add, git commit и добавим наши изменения в снимок.

Работа с Github

Перейдем к работе с github. Итак, чтобы создать репозиторий на github переходим github.com/new Вводим название, описание.

Чтобы инициализировать наш удаленный репозиторий, введем команду:

git remote add origin https://github.com/[ваш\_логин]/[имя\_репозитория]

Затем:

git push -u origin master

Команда **push** позволит вам отправить изменения, которые вы внесли в своем локальном репозитории, на удаленный репозиторий. В результате обновив страницу репозитория, мы увидим там наш файл **README.md**.

Давайте изменим наш файл в github. Для этого:

* Нажимаем на файл **README.md**;
* Затем на карандашик.
* Меняем текст на любой.
* Снизу пишем комментарий к коммиту и нажимаем Commit changes.

Чтобы получить эти изменения на вашем компьютере, выполним команду:

git pull origin master

С помощью данной команды можно получить последние изменения с ветки (в данном случае master) и автоматически слить их. Если вам необходимо просто получить последние изменения на репозитории, используйте git fetch.

Если вы захотите получить доступ к репозиторию другого проекта (к примеру open source), то вы можете выполнить команду git clone:

git clone https://github.com/vectree/resources/

По итогу, рассмотрим весь цикл работы с git, который мы проделали.

Мы инициализировали репозиторий git init, добавили файл README.md в рабочую директорию. После выполнили следующий ряд действий:

* git add - добавили файл в индекс (область подготовленных файлов);
* git commit - создали снимок с данным файлом;
* git push - отправили изменения на удаленный сервер;
* Изменили файл README.md на стороне **github**;
* git pull - получили изменения и слили их.