# Шпаргалка. Начало работы с Git

Чтобы вам было проще запомнить все команды, о которых шла речь в этом модуле, мы собрали их в одном месте.

### Инициализация репозитория

git init (от англ. ***init***ialize, «инициализировать») — инициализируй репозиторий.

### Синхронизация локального и удалённого репозиториев

git remote add origin https://github.com/YandexPracticum/first-project.git

git remote -v — проверь, что репозитории действительно связались;

git push -u origin main (от англ. push, «толкать») — в первый раз загрузи все коммиты из локального репозитория в удалённый с названием origin.

💡 Ваша ветка может называться master, а не main. Подправьте команду, если это необходимо.

git push (от англ. push, «толкать») — загрузи коммиты в удалённый репозиторий после того, как он был привязан с помощью флага -u.

### Подготовка файла к коммиту

git add todo.txt (от англ. add, «добавить») — подготовь файл todo.txt к коммиту;

git add --all (от англ. add, «добавить» + all, «всё») — подготовь к коммиту сразу все файлы, в которых были изменения, и все новые файлы;

git add . — подготовь к коммиту текущую папку и все файлы в ней.

### Создание и публикация коммита

git commit -m "Комментарий к коммиту."

git push (от англ. push, «толкать») — добавь изменения в удалённый репозиторий.

### Просмотр информации о коммитах

git log (от англ. log, «журнал [записей]») — выведи подробную историю коммитов;

git log --oneline (от англ. log, «журнал [записей]» + oneline, «одной строкой») — покажи краткую информацию о коммитах: сокращённый хеш и сообщение.

### Просмотр состояния файлов

git status (от англ. status, «статус», «состояние») — покажи текущее состояние репозитория.

### Добавление изменений в последний коммит

git commit --amend --no-edit (от англ. amend, «исправить») — добавь изменения к последнему коммиту и оставь сообщение прежним;

git commit --amend -m "Новое сообщение" — измени сообщение к последнему коммиту на Новое сообщение.

💡 Выйти из редактора Vim: нажать Esc, ввести :qa!, нажать Enter.

### «Откат» файлов и коммитов

git restore --staged hello.txt (от англ. restore, «восстановить») — переведи файл hello.txt из состояния staged обратно в untracked или modified;

git restore hello.txt — верни файл hello.txt к последней версии, которая была сохранена через git commit или git add;

git reset --hard b576d89 (от англ. reset, «сброс», «обнуление» + hard, «суровый») — удали все незакоммиченные изменения из staging и «рабочей зоны» вплоть до указанного коммита.

### Просмотр изменений

git diff (от англ. ***diff***erence, «отличие», «разница») — покажи изменения в «рабочей зоне», то есть в modified-файлах;

git diff a9928ab 11bada1 — выведи разницу между двумя коммитами;

git diff --staged — покажи изменения, которые добавлены в staged-файлах.