

## 1. git log — Просмотр коммитов

Команда для отображения истории коммитов, включая хэши, авторов и сообщения.

Пример:

```
git log
```

---

## 2. git reset — Возврат к начальному коммиту

Позволяет откатить изменения и вернуться к указанному коммиту, отменяя все последующие изменения.

Пример:

```
git reset <хэш_коммита>
```

---

## 3. git restore — Восстановление файлов

Используется для восстановления файлов из рабочего каталога. Если файлы уже добавлены в индекс (через `git add`), используйте флаг `--staged`, чтобы убрать их оттуда.

Примеры:

- Убрать файлы из индекса:  
`git restore --staged .`
  - Восстановить файл из рабочего каталога:  
`git restore <путь_к_файлу>`
- 

## 4. git add — Добавление изменений в индекс

Команда добавляет все изменения (или указанные файлы) в индекс для последующего коммита.

Пример:

```
git add .
```

---

## 5. Conventional Commits — Примеры оформления коммитов

Стандарт именования коммитов для упорядочивания:

- `feat:` добавлен новый функционал
  - `fix:` исправлен баг
  - `chore:` обновление конфигурации
- 

## 6. git rm --cached — Удаление файла из отслеживания

Удаляет файл или папку из индекса, не удаляя его физически из рабочей директории. Git перестает отслеживать эти файлы, но они остаются локально.

Пример:

```
git rm --cached <имя_файла>
```

---

## 7. git stash — Сохранение изменений и очистка рабочей директории

Команда временно сохраняет изменения в рабочей директории и индексируемых файлах, позволяя вернуться к ним позже.

Примеры:

- Сохранить изменения:  
`git stash`
  - Показать список сохранённых состояний:  
`git stash list`
  - Извлечь последнее состояние и применить его:  
`git stash pop`
  - Удалить последнее сохранённое состояние:  
`git stash drop`
  - Применить сохранённое состояние:  
`git stash apply stash@{0}`
- 

## 8. git commit --amend — Изменение последнего коммита

Позволяет изменить последний коммит (например, добавить новые изменения или исправить сообщение).

Примеры:

- Изменить последний коммит:  
`git commit --amend`
  - Изменить последний коммит без изменения сообщения:  
`git commit --amend --no-edit`
- 

## 9. git patch — Перенос изменений из другого репозитория

Создание патча для применения в другом репозитории.

Пример создания патча:

```
git diff <хэш> > my_patch.patch
```

Пример применения патча:

```
git apply my_patch.patch
```

---

## 10. git am — Применение нескольких патчей

Команда используется для применения всех патчей из указанной директории.

Пример:

```
git am ./patches/*
```