**Основы Git (основные команды)**

**Конспект уроков Нетологии.**

1. **Работа с историей изменения файлов.**

Просмотр изменений:

* **git diff**

Отображает изменения локальных файлов, которые находятся в рабочей директории, с тем, что уже закоммичено.

* **git diff --staged** и **git diff --cached**

Отображает разницу того, что добавлено в индекс но ещё не заммичено с тем, что уже закоммичено.

Cравнивает проиндексированные изменения с последним коммитом.

* **git difftool –** графическая утилита для оценки изменений.

1. **Фиксация изменений**

перед коммитом обязательно добавляем файлы в индекс (**git add .)**

* **git commit** - без параметров вызов редактора vi и последущее сохранение
* **git commit -m “ferst commit”** - быстрый коммит с комментариями.
* **git commit -v** - добавить в комментарии разницу (diff)
* **git commit --all** - автомотически добавит в индекс все файлы и сделает откроет редактор коммита.

1. **Удаление файлов из Git**

Чтобы удаление файлов попало в индекс надо использовать (**git rm**)

* **git rm file.txt**
* **git rm -f file.txt** (force) пропустить проверку актуальности (удалит файлы даже если они проиндексированы)
* **git rm --cached file.txt** - удалить файл только из индекса (перестать отслеживать), но оставить на HDD
* **git rm log/\\*.log** возможно использовать шаблоны для удаления (чтобы не прописывать каждый файл)
* **git rm -n, --dry-run file.txt** пробный запуск. Позволяет посмотреть, что будет, но не выполнять это.

1. **Перемещение файлов**

**git mv test.txt test.md**

1. **История коммитов**

* **git log -** просмотр всей истории изменений.
* **git log -3** просмотр трёх последних коммитов.
* **git log -3 --patch** - сразу показывает изменения в коммите вместе с логом
* **git log -3 --stat -** выводит статистику изменений (полезен для код ревью, что конкретно было изменено в коммитах)
* **git log --pretty=oneline, full, fuller -** форматированный вывод
* **git log --pretty=format:"%h - %an, %ar: - %s"** удобочитабельный вывод
* **git log --grep** “строка”

1. **Операции отмены**

Отмена правок в коммитах которые не нужны, случайны, ошибочны. Можно заменить предыдущий коммит на новый коммит.

Операция отмены - это одна из редких областей GIT, в которой не верными действиями можно необратимо утерять результат своей работы.

когда можно использовать: Забыли добавить файлы в последний коммит, добавили лишние файлы, указали неверный комментарий к коммиту.

1. **Работа с удалёнными репозиториями**