

### 1. Зачем нужен GIT и какие задачи решают с его помощью?

GIT - это система контроля версий ( система версионирования, командная утилита). Вот некоторые из многочисленных возможностей данной системы:

- Возврат к любой версии кода из прошлого.
- Просмотр истории изменений.
- Совместная работа без боязни потерять данные или затереть чужую работу.

### 2. Что такое репозиторий GIT и где он находится?

Репозиторий — это все файлы, находящиеся под контролем версий, вместе с историей их изменения и другой служебной информацией. Репозиторий Git можно создать, либо выбрав любую папку на компьютере, либо клонировать себе уже существующий репозиторий, например у работодателя.

Команда `git log` является основным инструментом Git для изучения истории репозитория. С помощью этой команды можно найти конкретную версию проекта или определить изменения, которые будут внесены при выполнении слияния в функциональной ветке.

Папка Git — это место, где Git хранит метаданные и объектную базу данных проекта. Это наиболее важная часть Git, которая копируется при дублировании репозитория (хранилища) с другого компьютера.

### 3. Для чего используется файл `.gitignore`?

`gitignore` нужен для скрытия файлов и папок от системы контроля версий Git. Обычно скрывают конфигурационные файлы (особенно с паролями), временные файлы и папки.

### 4. Что такое "коммит"?

Простыми словами, коммит — это снимок ваших локальных файлов, записанный в локальный репозиторий. Вопреки распространенному мнению, git хранит не только изменения между файлами, но и в целом полную версию всех файлов.

### 5. Укажите базовые команды в рамках работы с GIT:

`git init` - git включается или запускается для данного репозитория.

`rm -rf .git` - удаление файла из рабочей директории.

`git add (index.html)` - добавить содержимое рабочего каталога в индекс для последующего коммита.

`git add .` - добавить все файлы из каталога в репозиторий.

`git status` - показывает состояния файлов в рабочем каталоге и индексе: какие файлы изменены, но не добавлены в индекс; какие ожидают коммита в индексе. Вдобавок к этому выводятся подсказки о том, как изменить состояние файлов.

`git commit` - берет все данные, добавленные в индекс с помощью `git add`, и сохраняет их слепок во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот слепок.

`git reset` - используется в основном для отмены изменений. Она изменяет указатель HEAD и, опционально, состояние индекса.

`git log` - журнал коммитов

`git checkout master` - откатиться назад в файлы.

`git checkout -f (HEAD)` вернуться на последний head коммит.

6. Настройка аккаунта локального пользователя:

`git config --global user.name 'Madina Mel'`

`git config --global user.email 'wolfi251199@gmail.com'`

`cat ~/.config` посмотреть данные пользователя

`git config --global -l` тоже самое

7. Создание репозитория.

Для создания нового репозитория используется команда `git init`. Команду `git init` выполняют только один раз для первоначальной настройки нового репозитория. Выполнение команды приведет к созданию нового подкаталога.

8. Добавление файлов в индекс и выполнение коммита:

`git add (index.html)`

`git commit`

9. Просмотр журнала коммитов:

`git log`