

# КЛЮЧЕВЫЕ КОМАНДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ

### СТРУКТУРА РЕПОЗИТОРИЯ

### Папка .git

Скрытая папка в корне проекта, содержащая данные для работы Git

#### Объекты

Каждый файл и его версия хранятся как объекты с уникальными SHA-1-хешами

#### **HEAD**

• • •

Указатель на текущую ветку и последний коммит в ней

# ВЗАИМОДЕЙСТВИЕ С ФАЙЛАМИ В GIT

Добавление файлов, подготовка изменений для следующего коммита:

\_\_\_\_\_\_ git add <имя\_файла>

Чтобы удалить файл из рабочего каталога и индекса:

git rm <имя\_файла>

Перемещение, переименование файлов

git mv <старый\_путь/старое\_имя> <новый\_путь/новое\_имя>

Команда git pull получает изменения из удалённого репозитория и объединяет их с текущей веткой:

• • •

git pull origin main

Команда git push отправляет ваши локальные изменения в удалённый репозиторий:

• • •

git push origin main

В Git файлы могут находиться в нескольких состояниях, которые определяют, как они отслеживаются и обрабатываются системой контроля версий. Понимание этих состояний является ключевым для эффективного управления изменениями в проекте

# ОСНОВНЫЕ СОСТОЯНИЯ ФАЙЛОВ

#### Неотслеживаемые

Untracked

Файлы, которые присутствуют в рабочем каталоге, но еще не находятся под контролем версий. Git не отслеживает изменения в этих файлах до тех пор, пока они не будут добавлены в индекс

Пример: новый файл, который вы только что создали

#### Отслеживаемые

**Tracked** 

- Файлы, которые находятся под контролем версий. Эти файлы могут находиться в следующих подкатегориях
- Неизменённые (Unmodified): Файлы, которые не были изменены с момента последнего коммита
- Изменённые (Modified): Файлы, которые были изменены с момента последнего коммита, но еще не добавлены в индекс
- Индексированные (Staged): Файлы, которые были добавлены в индекс с помощью команды git add и готовы к коммиту

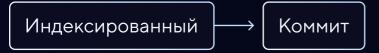
# ПЕРЕХОДЫ МЕЖДУ СОСТОЯНИЯМИ



Чтобы перевести файл из состояния "неотслеживаемый" в "отслеживаемый", используйте команду git add <имя\_файла>

Отслеживаемый (Изменённый) —— Индексированный

Чтобы перевести файл из состояния "изменённый" в "индексированный", используйте команду git add <имя\_файла>



Для фиксации изменений в истории используйте команду git commit

## ОТСЛЕЖИВАНИЕ ИЗМЕНЕНИЙ

git add <файл>

добавляет файл в индекс

Отслеживание позволяет подготовить изменения для фиксации в коммите

# ФИКСАЦИЯ ИЗМЕНЕНИЙ (КОММИТ)

git commit -m "сообщение"

создает новый коммит с заданным сообщением

Коммит сохраняет текущие изменения и добавляет их в историю изменений

# ПОЛЕЗНЫЕ КОМАНДЫ

git status

показывает текущий статус рабочей директории

git diff

помогает увидеть различия

git log

позволяет посмотреть историю коммитов и их содержание

### ИТОГ

- Понимание принципов размещения файлов в Git является основополагающим для эффективной работы с системой контроля версий. Знание о том, как организованы рабочий каталог, индекс и локальный репозиторий, позволяет разработчикам уверенно управлять своими проектами
- Понимание состояний файлов и умение управлять ими с помощью команд Git позволяет эффективно контролировать процесс разработки, обеспечивая надежное сохранение изменений и их фиксацию в репозитории. Это основа для успешной работы с системой контроля версий, обеспечивающей чистоту и согласованность вашей кодовой базы
- Отслеживание и фиксация ключевые действия для работы с Git
- Понимание процесса индексирования и фиксации изменений помогает эффективно управлять проектом