



17 фишек для эффективной работы с Claude Code

17 фишек для эффективной работы с Claude Code

Ключевые тезисы:

.###  Управление диалогом и контекстом

- `/clear conversation` — быстрая очистка текущего диалога без удаления истории (история сохраняется в боковой панели).
- `/compact` — сжатие диалога (создание саммари) для освобождения токенов.  При многократном использовании может потерять детали исходной задачи.
- `/context` — проверка заполненности контекстного окна (например, 3% из 1 млн токенов).

Настройки и лимиты

- `/usage` — просмотр лимитов: пятичасовые, недельные и лимиты для конкретных моделей (Opus, Sonnet).
- **Extra Usage** — платная опция для продолжения работы при исчерпании лимитов или активации **Fast Mode**.
- **Fast Mode** — ускорение работы в 2.5 раза. Активируется через настройки VS Code (`"fastMode": true`) после оплаты Extra Usage.
- `@` (символ "собака") — привлечение внимания Claude к конкретному файлу в контексте.

Голосовой ввод и режимы работы

- **Голосовой ввод в терминале** — активируется командой `/voice`. Бесплатен в подписке Claude Code.
- **Голосовой ввод в системе** — можно использовать сторонние платные приложения (например, Aqua) для ввода текста в любом месте.

- **Режимы работы (в настройках VS Code):**
 - **ask before edit** — запрос разрешения на любое изменение.
 - **automatic edit** — запрос разрешения на основные изменения.
 - **bypass permissions** — почти полная автономность (кроме редактирования папок `.claude`).
 - **Auto Mode** (скоро появится) — модель сама решает, когда запрашивать разрешение.

Повышение эффективности

- **Plan Mode** — обязательное планирование перед выполнением сложных задач. Сильно повышает эффективность и снижает количество ошибок.
- **Ultra Think** — уровни "думанья" (от low до max). Чем выше уровень, тем больше токенов тратится, но тем лучше Claude справляется со сложными задачами.
- **/bw (byway) в терминале** — возможность задать уточняющий вопрос Claude, пока он выполняет основную задачу, не прерывая её. В расширении аналог — уточнение через историю диалогов в отдельной вкладке.

Автоматизация и расширенные возможности

- **CLAUDE.md** — системный промпт-файл с информацией о проекте, чтобы не повторять её в каждом диалоге.
- **Skills (Скиллы)** — стандартные операционные процедуры для автоматизации рутинных задач (например, анализ Telegram-канала через браузер с помощью Playwright).
- **Саб-агенты:**
 - **Для поиска в документации Claude** — чтобы получать точные ответы по настройке, не выдумывая лишнего.
 - **Для ресерча в интернете** — на модели Sonnet, чтобы не захламлять контекст главного агента технической информацией (HTML-разметкой и т.д.).
- **Hooks (хуки)** — настройка звуковых уведомлений, например, когда Claude завершает работу или создаёт новый план.

- **Git Worktrees (через отдельный skill)** — работа над одним проектом в нескольких параллельных сессиях без конфликтов. Каждый инстанс работает со своей копией проекта, что ускоряет разработку в 4-5 раз.

Полезные детали по подпискам

- **Подписка Pro (\$20)** — можно апгрейдить до Max (\$100), доплатив разницу (\$80), а не полную стоимость.
- **Подписка Max (\$200)** — даёт в 20 раз больше лимитов, чем Pro.

Выводы:

Claude Code — мощный инструмент, эффективность которого можно значительно повысить за счёт правильного управления контекстом (`/compact` , `@`), использования планирования (`Plan Mode`), автоматизации рутины (`Skills`) и параллельной работы (`Git Worktrees`). Ключевые навыки — умение настраивать поведение Claude через промпты и саб-агентов, а также контроль над лимитами и режимами работы.