

## Skills в Claude Code: полный гайд по автоматизации

---

# Полный гайд по Skills в Claude Code и Vibe Coding






## Ключевые тезисы

- **Skills** — это открытый стандарт для автоматизации рутины и расширения возможностей AI-инструментов.
- **Claude Code** обладает наиболее продвинутой реализацией skills.
- Основная ценность — автоматизация **конкретных рабочих процессов (воркфлоу)**, а не абстрактных задач.
- Skills частично заменяют **MCP (Model Context Protocol)**, экономя контекст.
- Skills можно создавать самостоятельно, скачивать из открытых источников или использовать официальные от Anthropic.

## Что такое Skills?

*Skills* — это стандартизированные инструкции и скрипты, которые позволяют AI-агенту выполнять конкретные, повторяющиеся задачи. Это **конкретные автоматизации**, привязанные к вашей работе.

### Примеры использования:

-  Монтаж видео (нарезка, рендеринг)
-  Создание презентаций в Excalidraw
-  Автоматический ресерч новостей
-  Публикация контента в Telegram
-  Генерация коммерческих предложений

## Сравнение инструментов

- **Claude Code:** Самые продвинутые skills (выбор модели, уровень "думания", изолированный контекст, хуки).
- **Antigravity/Codex:** Базовая функциональность (имя, описание, скрипт).
- **OpenAI Hermes Agent:** Поддержка стандарта.

**⚠ Важно:** При скачивании skills из неофициальных источников есть риск *prompt injection* — вредоносных инструкций, которые могут слить ваши данные.

## Анатомия Skill в Claude Code

Базовая структура (работает везде):


- `name` — название skill
- `description` — описание
- Основное тело skill (инструкции)

Расширенные возможности (только Claude Code):

- `disable_model_invocation` — запрет авто-вызова, только через `/`
- Выбор модели (можно использовать дешевую для простых tasks)
- `effort` — уровень "думания" модели
- `context_fork` — запуск в изолированном контексте
- `paths` — авто-активация при работе с определенными файлами
- `hooks` — скрипты, срабатывающие при событиях
- Аргументы через слэш-команды

## Организация Skills

- **Локальные:** Лежат в папке `.cursor` проекта, работают только в нем.
- **Глобальные:** Лежат в `~/.cursor`, подтягиваются во все проекты.

 **Рекомендация:** Глобальные skills — для расширения функционала Claude.  
Локальные — для конкретных рабочих процессов.

## Практическое использование

Как создавать skills:

1. **Вручную:** Прописать последовательность действий, которую успешно выполнил AI.
2. **Через плагин `skill-creator`:** Официальный инструмент Anthropic для создания и улучшения skills.

Плагин `skill-creator` умеет:

- Создавать новые skills с правильной структурой
- Запускать **evaluations** — тестирование эффективности
- Сравнить две версии skill (benchmark)
- Улучшать существующие skills на основе тестов

Процесс работы с `skill-creator`:

Запрос на создание skill → Уточняющие вопросы → Создание skill →  
→ Авто-тестирование → Предложения по улучшению





## От Skill к Автоматизации

Любой skill можно преобразовать в полноценную автоматизацию:





1. Вынести на сервер
2. Подключить к LLM через API
3. Создать удобный UI
4. Интегрировать в бизнес-процессы

**Пример:** Skill для генерации презентаций → Веб-интерфейс с полем ввода и кнопкой "Сгенерировать".

## Практические советы


-  **Экономьте контекст:** Skills видят только `name` и `description` до вызова.
-  **Длина skill:** Рекомендуется не более 500 строк.
-  **Бюджет описаний:** На все `name` и `description` выделяется ~1% контекста (10К токенов из 1М).
-  **Используйте MCP с умом:** Только для критически важного функционала (например, Playwright для браузера).

## Важные предупреждения

-  Не используйте skills из непроверенных источников
-  Избегайте множественных MCP — они "жрут" контекст
-  Проверяйте skills через evaluations перед использованием
-  Начинать с автоматизации самой рутинной работы (80-90% задач часто автоматизируемы)

## Выводы

- Skills в Claude Code — **мощнейший инструмент** для автоматизации рутины.
- **Самые ценные skills** — те, что описывают ваши конкретные рабочие процессы.
- **Не нужно быть программистом** — достаточно описать последовательность действий.
- **Экономия времени** — до 10-15 раз за счет автоматизации повторяющихся задач.
- **Будущее** — преобразование локальных skills в облачные автоматизации с UI.

 **Главный совет:** Начните с анализа своей ежедневной рутины и автоматизируйте самое времязатратное через skills.