






## **Скиллы в Claude: инструкции для стабильного результата**

---

### **Что такое скиллы в Claude и зачем они нужны**

Ключевые тезисы:

-  **Скилл** — это текстовый файл-инструкция, который даёт Claude новый навык и обеспечивает стабильный, предсказуемый результат.
  -  **Промпт** — это разовая просьба, качество выполнения которой может «плавать».
  -  **Скилл** — это «чертёж», по которому Claude работает одинаково хорошо каждый раз.
  -  Скилл — это архив (папка) с обязательным файлом `skill.md` внутри.
  -  **Безопасность:** 36% скиллов на маркетплейсах содержат уязвимости (prompt injection).
- 

### **Создание и суть скиллов**

#### **Плавающее качество промптов**

Одна и та же задача (например, написать пост) выполняется Claude по-разному в разных чатах. Память Claude запоминает контекст (имя, нишу), но не гарантирует идентичное выполнение конкретных задач.

**Аналогия:** Архитектор нанимает строителя.

- **Без чертежа (промт):** «Построй хороший дом». Результат будет хорошим, но непредсказуемым (стены не там, окна не там).

- **С чертежом (скилл):** Детальный план с размерами, материалами, чеклистом. Результат — качественный и абсолютно предсказуемый.

Скилл — это и есть такой «чертёж» для Claude.

## Создание первого простого скилла

**Задача:** Всегда писать поздравления для друга заглавными буквами (CAPS).

1. **Без скилла:** Новый чат → «Придумай поздравление для друга» → обычный ответ.
2. **Со скиллом:**
  - Заходим в **Customize Skills** → **+** → **Write skill instructions**.
  - **Skill name:** **CAPS**.
  - **Description:** Подробно описываем, когда применять скилл (например, «Используй эту инструкцию всегда, когда я прошу написать поздравление, пост или приветствие для моего друга»).
  - **Instructions:** «Всегда, когда прошу написать поздравление для друга, пиши его заглавными буквами».
  - Нажимаем **Create**.

**Результат:** В новом чате при запросе «напиши поздравление для моего друга» Claude автоматически применяет скилл и выводит текст заглавными буквами.

---

## Формат и структура скиллов

### Что такое файл **.skill** ?

- Это архив (аналогичный **.zip** или **.rar**), созданный компанией Anthropic для удобного обмена.
- Файл с расширением **.skill** при двойном клике автоматически устанавливается в Claude.

- Если изменить расширение `.skill` на `.zip` и открыть, появится папка с содержимым.

## Внутренняя структура

Внутри папки (архива) **обязательно** должен находиться файл с именем `skill.md`.

- Этот файл содержит все инструкции в формате **Markdown** (`.md`).
- *Markdown* — это текстовый формат с простой разметкой (например, `#` для заголовка, `-` для списка), созданный Джоном Грубером (сооснователем Reddit) в 2004 году.
- Claude использует структуру Markdown, чтобы понять, где название скилла, описание и сами инструкции.

Пример содержимого `skill.md`:



```
---  
name: CAPS  
description: Используется для написания поздравлений другу  
---  
Всегда, когда прошу написать поздравление для друга, пиши его заглавными буквами.
```

- Три тире (`---`) отделяют метаданные (название, описание) от самой инструкции.
- Размер такого файла может быть крошечным (например, 370 байт).



## Способы установки скиллов

1. **Загрузка готового файла:** `+` → **Upload skill** → выбрать файл с расширением `.skill`, `.zip` или `.md`.
2. **Написание вручную:** `+` → **Write skill instructions** → ввести имя, описание и инструкцию. Подходит для простых скиллов.

3. **Создание с помощью Claude:**  → **Create with Claude** → объяснить боту в чате, какой навык нужен. Claude задаст уточняющие вопросы и в итоге создаст скилл, который можно добавить в библиотеку одним кликом.
  -  **Полезный приём:** После успешной работы в чате можно попросить Claude: «Сделай скилл на основании нашего чата», чтобы сохранить результат как навык.

---

## Работа со сложными скиллами (например, с GitHub)

Часто на GitHub скиллы выложены в виде репозитория с множеством файлов.

**Типичная проблема:** Не удаётся загрузить большой `.zip` архив всего репозитория.

**Правильный алгоритм:**

1. Скачать репозиторий и распаковать его.
2. Найти нужную папку со скиллом (например, `pdf`).
3. Убедиться, что внутри есть `skill.md`.
4. Проверить и изменить (при необходимости) поле `name:` внутри файла `skill.md`, чтобы оно было уникальным в вашей библиотеке.
5. Заpackовать только эту конкретную папку в новый ZIP-архив.
6. Загрузить этот новый ZIP-файл через **Upload skill**.

**Итог по форматам:**

- `.md` — один файл с инструкциями.
- `.zip` / `.skill` — папка (архив), содержащая файл `skill.md` и, возможно, другие ресурсы (скрипты, шаблоны).
- Содержимое во всех случаях одинаково — папка с файлом `skill.md` внутри. Меняется только «обёртка».



## Вопрос безопасности

Исследование 2026 года: 36% скилов на маркетплейсах содержат уязвимости.

- **Prompt Injection** — это когда в обычный текст инструкции встроены невидимые команды (специальные Unicode-символы). Вы их не видите в редакторе, но Claude видит и выполняет.
- **Риск:** Запуск вредоносных инструкций.

Рекомендации:

1. **Лучший скилл — созданный вами** (через Skill Creator, текстовый редактор или с помощью Claude).
2. Если скачиваете чужой скилл — **откройте файл `skill.md` и внимательно прочитайте каждую строку**. Если что-то выглядит странно или непонятно — не устанавливайте.

---

## Пример полезного скилла: YouTube Video Analyzer

**Что делает:** Берёт ссылку на любое видео YouTube, автоматически (через связку Quark + Claude Chrome connector) копирует и анализирует текст, делая разбор (реверс-инженеринг) контента.

**Ценность:** Избавляет от ручного копирования транскрипта. На создание такого сложного скилла может уйти около часа отладки.

💡 **Главный принцип:** Как только получили желаемый результат в чате — сразу сохраняйте его как скилл, чтобы не повторять одни и те же шаги в будущем.

---

# Выводы

1. **Скилл** — это навык-инструкция для Claude, обеспечивающий стабильное качество.
2. **Основа скилла** — файл `skill.md` в формате Markdown.
3. **Устанавливать** можно файлы `.skill`, `.zip` или `.md`.
4. **Сложные скиллы** с GitHub нужно правильно перепаковывать, следя за уникальностью имени.
5. **Будьте осторожны** со сторонними скиллами — проверяйте содержимое `skill.md` на предмет скрытых команд (prompt injection).
6. **Самый безопасный и точный скилл** — тот, который вы создали сами под свои конкретные задачи.