Заставь Claude смотреть видео за тебя: чек-лист из 3 бесплатных пакетов

В Claude появится команда /watch: даёшь ссылку на YouTube (или любой видеофайл, скринкаст, Zoom-запись) → он скачивает видео, режет на кадры по сменам сцен, привязывает их к таймкодам, читает каждый скриншот и выдаёт полную транскрипцию с описанием происходящего.

> **Главное:** Claude перестаёт галлюцинировать «как сделать», потому что реально видит экран — кадр за кадром, а не угадывает по тексту.

Зачем это нужно

Скринкасты с багами

записал созвон или обзор продукта → Claude сам смотрит, составляет план правок и чинит баги в проекте (тебе смотреть не нужно)

YouTube-инструкции

даёшь ссылку на туториал → Claude по кадрам точно понимает, как собрать нужную систему, и не угадывает

Конкурентная разведка

массово выгружай заголовки, описания, каналы и субтитры чужих видео

Как это работает: руки + уши + глаза

Уши — Whisper

превращает звук в текст

Руки — yt-dlp

качает видео, субтитры и метаданные

Глаза — FFmpeg

нарезает кадры, чтобы Claude видел картинку

Вместе = Claude, который **смотрит**, а не только читает

Промт-установщик (скопируй целиком и вставь в Claude Code / Codex)

Ты — Claude Code. Создай мне с нуля рабочий skill «/watch», который смотрит любое видео по ссылке (YouTube, Loom, запись Zoom, прямой mp4): не просто читает субтитры, а ВИДИТ, что на экране — кадры берёт по сменам сцен, а не по таймеру. Сделай всё сам, по шагам, и в конце отчитайся.

ШАГ 1. Проверь и при необходимости установи зависимости:
- yt-dlp (скачивание видео и субтитров)
- ffmpeg (извлечение кадров по сменам сцен)
- openai-whisper (транскрипция, если у видео нет субтитров)
Определи мою ОС сам. macOS: "brew install yt-dlp ffmpeg" и "pip install -U openai-whisper". Linux: "pipx install yt-dlp", "sudo apt install -y ffmpeg", "pip install -U openai-whisper". Windows: "winget install yt-dlp.yt-dlp" и "winget install Gyan.FFmpeg", затем "pip install -U openai-whisper". Если что-то уже стоит — пропусти. Перед установкой покажи мне команды и спроси подтверждение.

ШАГ 2. Создай папку ~/.claude/skills/watch/ (на Windows — соответствующий путь профиля .claude). Запиши в неё файл SKILL.md ровно с содержимым между маркерами <<<SKILL_START>>> и <<<SKILL_END>>> (сами маркеры в файл не клади):
<<<SKILL_START>>>
---
name: watch
description: |
  Смотрит любое видео по ссылке (YouTube, Loom, запись Zoom, прямой mp4) — не только
  читает транскрипт, но и ВИДИТ, что на экране. Кадры берёт по сменам сцен (ffmpeg scene
  detection), а не по таймеру, поэтому не пропускает графику, демо, монтаж и b-roll.
  Звук: субтитры через yt-dlp, fallback — Whisper. Используй, когда пользователь пишет
  /watch <url> или просит «посмотри это видео», «разбери ролик/созвон/лекцию», «что показано
  на экране». Триггеры: /watch, посмотри видео, разбери ролик, проанализируй видео, transcribe.
---
 watch — Claude смотрит видео целиком

Зависимости (проверить при первом запуске)
- yt-dlp, ffmpeg, whisper. Если чего-то нет — установить и сообщить пользователю.

watch — Claude смотрит видео целиком
Зависимости (проверить при первом запуске)
- yt-dlp, ffmpeg, whisper. Если чего-то нет — установить и сообщить пользователю.

Workflow

1. Подготовка
- Сделай рабочую папку: mkdir -p /tmp/watch/<slug>
- Распарси URL (YouTube / Loom / Zoom-share / прямой видеофайл).

2. Транскрипт — сначала субтитры (бесплатно, если есть)
yt-dlp --write-subs --write-auto-subs --sub-langs "ru,en" --skip-download --convert-subs srt -o "/tmp/watch/<slug>/subs.%(ext)s" "<URL>"
Если .srt появился — это транскрипт с тайм-кодами, используй его.

3. Транскрипт — fallback Whisper (если субтитров нет)
yt-dlp -x --audio-format mp3 -o "/tmp/watch/<slug>/audio.mp3" "<URL>"
whisper "/tmp/watch/<slug>/audio.mp3" --model small --language ru --output_format srt --output_dir "/tmp/watch/<slug>/"

4. Кадры ПО СМЕНАМ СЦЕН (ключевой шаг — не по таймеру)
yt-dlp -f "bv*[height<=720]+ba/b[height<=720]" -o "/tmp/watch/<slug>/video.mp4" "<URL>"
ffmpeg -i "/tmp/watch/<slug>/video.mp4" -vf "select='gt(scene,0.3)',showinfo" -vsync vfr "/tmp/watch/<slug>/frames/frame_%04d.png" 2> "/tmp/watch/<slug>/scenes.log"
Порог 0.3 — крути 0.2–0.4 по плотности сцен. Тайм-коды кадров бери из showinfo-лога (pts_time), чтобы привязать кадр к моменту. Если кадров слишком много — оставь до ~150 самых информативных, прореживая близкие по времени.

5. Анализ
- Прочитай транскрипт (.srt).
- Просмотри извлечённые кадры (vision): слайды, код, демо, графика.
- Свяжи визуал с транскриптом по тайм-кодам.

6. Отчёт
Выдай: TL;DR (3–5 строк); ключевые концепции с тайм-кодами; что показано НА ЭКРАНЕ, чего нет в тексте; заметные моменты/цитаты с тайм-кодами.

7. Ингест в базу знаний (спросить)
Спроси: «Сохранить разбор в твою базу знаний (Obsidian)?». Если да — запиши markdown-заметку в указанную папку со ссылками на связанные заметки. Без подтверждения — не сохраняй.

Правила
- Рабочие файлы — во временную папку, проект не засоряй.
- Ничего не публикуй и не отправляй наружу без явной просьбы.
- Приватное видео с логином не обходи — сообщи об этом.
<<<SKILL_END>>>

ШАГ 3. Проверь результат: убедись, что файл ~/.claude/skills/watch/SKILL.md создан, yt-dlp/ffmpeg/whisper доступны (выведи их версии). Сделай тестовый сухой прогон: возьми любую короткую публичную ссылку и забери ТОЛЬКО субтитры (yt-dlp ... --skip-download), чтобы подтвердить, что цепочка работает — видео целиком не качай.

ШАГ 4. Отчитайся: что установил, где лежит skill, и напиши мне ровно одну строку-инструкцию, как пользоваться: «/watch <ссылка> — посмотри видео целиком и дай разбор».

Команда /watch: как пользоваться

Запуск

в новом диалоге пиши /watch + ссылка на YouTube (или путь к видеофайлу / MP3 / Zoom-записи)

Что происходит

Claude качает видео, режет на кадры по сменам сцен, привязывает к таймкодам, берёт субтитры (или транскрибирует Whisper'ом) и читает каждый скриншот

Скорость

7-минутное видео — меньше минуты: около **63 кадров**, привязанных к таймкодам

На выходе

полная транскрипция + описание происходящего в кадре; Claude сам предложит сохранить разбор в твою базу знаний (Obsidian)