12 функций Codex, которые стоит включить в проекте

1. Инструкции внутри проекта

Codex лучше работает, если у него есть не просто общий промпт, а понятная система инструкций. Что сделать:

- Создать отдельные инструкции на уровне аккаунта
- Добавить правила для конкретного проекта
- Указать, где Codex должен искать доп. регламенты
- Разделить общие инструкции и инструкции под код
- Подключить файлы с правилами, референсами и примерами проектов

> Зачем это нужно: Codex получает не короткую заметку, а маршрут: где брать контекст, какие правила учитывать и как работать именно в этом проекте.

2. Параллельные субагенты

Codex может запускать несколько агентов для одной задачи, но запрос должен быть явным. Что сделать:

- Попросить Codex запустить несколько субагентов
- Разделить задачу по ролям: исследование, тесты, аудит
- Указать, что каждый агент работает в своей ветке
- Смотреть статус каждого агента в боковом чате
- Проверять, что сделал каждый агент отдельно

> Пример логики: Один агент изучает структуру проекта, второй запускает тесты, третий проверяет потенциальные проблемы. Так задача выполняется быстрее и качественнее.

3. Облачные задачи Codex Cloud

Codex может выполнять задачу не только локально, но и в облаке. Что сделать:

Подключить Codex Cloud к нужному репозиторию	Выбрать проект, где есть GitHub-ветка	Отправить задачу в облако
Закрыть ноутбук или переключиться на другие дела	Вернуться позже и проверить результат	

Зачем это нужно: Задача продолжает выполняться без постоянного открытого локального чата.

4. Команда /review

После выполнения задачи Codex можно попросить сделать аудит созданного кода. Что сделать:

Завершить задачу в ветке	Написать команду /review	Получить отчёт по проблемам
Исправить найденные баги и слабые места	Использовать ревью как обязательный этап перед финальным результатом	

Зачем это нужно: Кодовая гигиена не даёт проекту быстро превратиться в набор костылей.

5. Подключения по MCP

В Codex можно подключать MCP и плагины, чтобы расширять возможности агента.

Что сделать:

- Открыть настройки Codex
- Перейти в раздел плагинов
- Проверить, какие подключения уже включены
- Добавить нужные MCP под свои задачи
- Настроить каждый MCP отдельно

Что можно подключать:

- генерацию изображений
- работу с GitHub
- инструменты для дизайна и иконок
- сервисы для работы с данными
- свои API через MCP-доступ

Зачем это нужно: Codex получает доступ к инструментам, которые помогают ему выполнять задачи не только внутри текста или кода.

6. Skills под конкретный проект

Навыки удобнее хранить не в общей корневой папке, а внутри конкретных репозиториев. Что сделать:

Разделить skills по проектам	Не складывать все навыки в одну общую папку	Хранить skills рядом с тем репозиторием, где они реально нужны
Загружать их в GitHub вместе с проектом	Использовать разные наборы skills для кода, контента, исследований и маркетплейсов	

Зачем это нужно: Навыки не путаются между проектами, а команда сразу получает нужный набор правил и сценариев работы.

7. Веб-исследования

Codex хорошо работает с актуальным поиском и может сам искать данные, если задача сформулирована правильно. Что сделать:

Просить Codex не гадать, если он не уверен

Прямо указывать, что нужны актуальные данные

Проверять источники, которые он использовал

Давать задачу не просто «найти», а собрать вывод, план или решение

Использовать веб-исследование не только для кода, но и для рабочих и бытовых задач

> Пример запроса: «Не гадай, если не уверен. Найди актуальные данные сам и объясни, что изменилось».

8. Генерация изображений

Codex может генерировать изображения внутри проекта, если эта возможность подключена.

Что сделать:

- Использовать команду или навык для генерации изображения
- Дать задачу на конкретный элемент: иконку, картинку, визуал для блока
- Указать формат и назначение изображения
- Проверить, подходит ли результат для проекта
- При необходимости доработать через повторный запрос

Где полезно:

- иконки для сайтов
- изображения для лендингов
- визуалы для интерфейса
- тестовые картинки для продукта

9. Несколько репозиториев в одном проекте

Codex можно дать доступ сразу к нескольким папкам или репозиториям. Что сделать:

Открыть настройки проекта	Добавить основную рабочую папку	Добавить доп. папки для контекста
Указать, какая папка является основной	Использовать дополнительные репозитории как базу знаний	

Зачем это нужно: Codex может опираться не только на текущий проект, но и на связанные материалы: прошлые видео, исследования, документы, референсы и базу опыта.

10. Восстановление контекста через новый тред

Если в чат попал лишний контекст, можно продолжить работу с нужного сообщения. Что сделать:

Найти сообщение, с которого нужно продолжить	Нажать «создать тред»	Выбрать продолжение в новом локальном чате или рабочем пространстве
Убрать лишний контекст	Продолжить задачу без старого мусора в диалоге	

Зачем это нужно: Можно не начинать всё с нуля, но и не тащить за собой ошибочные сообщения.

11. Запись навыков и автоматизаций

Codex может записывать действия пользователя и превращать их в навык или автоматизацию.

Что сделать:

- Нажать «записать навык»
- Разрешить запись действий
- Показать процесс на экране
- Дать Codex собрать навык по записанному сценарию
- Использовать его для повторяющихся задач

Для чего подходит:

- выкладка видео
- работа с файлами
- повторяющиеся действия в сервисах
- обучение ассистента или команды
- автоматизация рутины

12. Сервер через терминал

Codex можно использовать так, чтобы задача выполнялась на другом компьютере.
Что сделать:

Поднять сервер Codex на домашнем или рабочем компьютере

Подключиться к нему с другого устройства через терминал

Отправлять задачи с ноутбука

Выполнять их на более стабильной или мощной машине

Использовать это для работы в поездках или при плохом интернете

Зачем это нужно: Можно работать с проектами удалённо, даже если основной компьютер находится в другой стране.

Итог

Codex становится намного сильнее, когда его используют не как обычный чат, а как рабочую агентную систему: с инструкциями, подключениями, навыками, ревью, несколькими репозиториями и понятной логикой выполнения задач.