Команды Claude Code и Codex

Каталог команд: что делает каждая, чем отличается в двух CLI, как вызвать.

КАК ЧИТАТЬ КАТАЛОГ

Команда правит сессию, промпт — задаёт работу

/

Наберите **/** — откроется список; печатайте дальше для фильтра.

⇥

Задача уже идёт — команда + **Tab** ставит её в очередь.

● — блок Claude Code, ● — Codex.

md

Память проекта: **CLAUDE.md** · Claude Code, **AGENTS.md** · Codex.

Состав команд зависит от версии CLI, модели и экспериментальных флагов.

01

РАЗДЕЛ 01

Настройка проекта

/init · /permissions · /model · /status

НАСТРОЙКА ПРОЕКТА

/init

Claude Code · Codex

- **Claude Code**

Создаёт **CLAUDE.md** в корне — постоянный бриф проекта.

- **Codex**

Создаёт каркас **AGENTS.md** ; отредактируйте и закоммитьте.

```
# в корне проекта
/init
→ CLAUDE.md (Claude Code)
→ AGENTS.md (Codex)
```

НАСТРОЙКА ПРОЕКТА

/permissions

Claude Code · Codex

- **Claude Code**

  Режим разрешений и недавние отказы.

- **Codex**

  Пресет одобрений: Auto · Read Only · custom. **/approve** — повтор отклонённого действия.

```
# Codex
/permissions → Auto · Read Only
/approve → повтор отказа
# Claude Code
/permissions
```

НАСТРОЙКА ПРОЕКТА

/model

Claude Code · Codex

- **Claude Code**

Смена модели. **/effort** low…xhigh|max|auto — глубина рассуждений.

- **Codex**

Модель + уровень reasoning. **/fast on|off** — Fast service tier.

```
# Claude Code
/model opus
/effort xhigh
# Codex
/model · /fast on
```

НАСТРОЙКА ПРОЕКТА

/status

Claude Code · Codex

- **Claude Code**

  Модель, режим разрешений, подключённые MCP-серверы.

- **Codex**

  Модель, политика одобрений, writable roots, токены, remote.

```
/status
→ модель и режим разрешений
→ MCP / инструменты
→ расход токенов
```

02

РАЗДЕЛ 02

Контекст и сессии

/context · /compact · /clear · /resume · /rewind

КОНТЕКСТ И СЕССИИ

/context · /compact

- **Claude Code**

/context — заполнение окна. **/compact [фокус]** сжимает историю: `keep decisions only`.

- **Codex**

/compact сжимает ранние ходы. Токены — в **/status**.

```
# Claude Code
/context
/compact keep decisions only
# Codex
/compact
```

КОНТЕКСТ И СЕССИИ

/clear · /new

Claude Code

/clear очищает историю. Правки файлов остаются.

Codex

/clear — терминал + чат. **/new** — новый чат без очистки терминала.

```
# сменить задачу начисто
/clear
# Codex: новый чат
/new
```

КОНТЕКСТ И СЕССИИ

/resume

Claude Code · Codex

- **Claude Code**

  Именованные и фоновые сессии. **Ctrl+A** — все проекты.

- **Codex**

  Пикер сессий. **/archive** + `codex unarchive`.

```
# Claude Code
/resume feature-auth
claude -r "feature-auth"
# Codex
/resume · /archive
```

КОНТЕКСТ И СЕССИИ

/rewind

Claude Code · alias /undo

Claude Code

Откат разговора и правок файлов к чекпоинту. Alias — **/undo**.

Codex

Аналога нет. Ветка — **/fork**; откат правок — через git.

```
# Claude Code
/rewind
→ выбрать точку отката
/undo
```

03

РАЗДЕЛ 03

Планирование и цели

/plan · /goal

ПЛАНИРОВАНИЕ И ЦЕЛИ

/plan

Claude Code · Codex

- **Claude Code**

Plan-режим: предлагает каждое действие и ждёт одобрения.

- **Codex**

/plan [запрос] — план + сразу задача. Недоступна во время задачи.

```
# Claude Code
/plan
# Codex
/plan предложи план миграции
```

ПЛАНИРОВАНИЕ И ЦЕЛИ

/goal

Claude Code · Codex

- **Claude Code**

Условие завершения; работает до него. Оверлей: время, ходы, токены. Интерактив, **-p**, Remote Control.

- **Codex**

Цель ≤4000 символов. **/goal** — показать; **pause · resume · clear** .

```
# Claude Code
/goal починии тест checkout
# Codex
/goal все тесты проходят
/goal pause · resume · clear
```

04

РАЗДЕЛ 04

Ревью и изменения

/review · /code-review · /security-review · /diff

РЕВЬЮ И ИЗМЕНЕНИЯ

/review

Claude Code · Codex

- **Claude Code**

  Ревью изменений. **/ultrareview** — облачное, параллельные агенты; с номером — конкретный PR.

- **Codex**

  Фокус на изменениях поведения и тестах. Модель — `review_model`.

```
# обе CLI
/review
# Claude Code: облачное
/ultrareview 482
```

Специализированные ревью

/code-review

Ревью на корректность. **--fix** применяет правки, **--comment** — инлайн в GitHub PR.

/simplify

Только чистка: упрощение, переиспользование, структура.

/security-review

Безопасность: инъекции, XSS, утечки секретов. Перед merge или релизом.

```
/code-review --fix
/security-review
/simplify
# Codex: всё закрывает /review
```

РЕВЬЮ И ИЗМЕНЕНИЯ

/diff

Claude Code · Codex

- **Claude Code**

  Интерактивный просмотрщик; стрелки, **j/k**, PageUp/Down.

- **Codex**

  Git diff: staged, unstaged и **untracked**. Запускайте после **/review**.

```
# перед коммитом
/review
/diff
```

05

РАЗДЕЛ 05

Параллельная работа

/agent · /fork · /side · /btw · /ps · /stop

ПАРАЛЛЕЛЬНАЯ РАБОТА

/agent · /fork · /side

- **Claude Code**

  /agents — субагенты; `claude agents` — agent view. Фон — **/tasks**, **/workflows**.

- **Codex**

  /agent — субагент; **/fork** — ветка; **/side** — детур; **/ps · /stop** — фоновые терминалы.

```
# Codex
/fork · /side
/ps · /stop
# Claude Code
/agents
```

ПАРАЛЛЕЛЬНАЯ РАБОТА

/btw · /side

- **Claude Code**

/btw <вопрос> — быстрый лукап без добавления в основной тред.

- **Codex**

/side · /btw — эфемерный детур; статус родителя виден. Нельзя вложить.

```
# Claude Code
/btw какой флаг для json-вывода?
# Codex
/side · /btw
```

06

РАЗДЕЛ 06

MCP, Skills, Hooks

/mcp · /skills · /hooks

/mcp

Claude Code · Codex

- **Claude Code**

Управляет серверами; даёт динамические `/mcp__server__prompt`. Reconnect без рестарта.

- **Codex**

Список доступных инструментов. **/mcp verbose** — диагностика серверов.

```
# Claude Code
/mcp
/mcp__github__list_prs
# Codex
/mcp verbose
```

MCP, SKILLS, HOOKS

/skills

Claude Code · Codex

- **Claude Code**

Скиллы в `.claude/skills/name/SKILL.md`. **/reload-skills** — пересканировать без рестарта.

- **Codex**

Обзор и подключение локальных скиллов для следующего запроса.

```
# обе CLI
/skills
# Claude Code: перечитать
/reload-skills
```

/hooks

Claude Code · Codex

- **Claude Code**

SessionStart — перезагрузить навыки, заголовок.

MessageDisplay — трансформировать ответ.

- **Codex**

Просмотр хуков: доверять, отключать или инспектировать.

```
/hooks
# события Claude Code
SessionStart · MessageDisplay
```

07

РАЗДЕЛ 07

Глубина рассуждений

ultrathink · ultracode

ГЛУБИНА РАССУЖДЕНИЙ

СЛОВО В ПРОМПТЕ

ultrathink

Claude Code · v2.1.68

Добавьте слово **ultrathink** в любое место промпта — агент думает глубже **на этот ход**, затем возврат к medium.

Работает только в терминале Claude Code — не в вебе и не через API. Шкала рассуждений: **think** → **think hard** → **think harder** → **ultrathink**.

Codex: магического слова нет; глубина — через **/model**.

```
# добавить куда угодно
Разбери этот race condition ultrathink
ultrathink Спроектируй новую схему БД
→ высокий effort · 1 ход
```

ГЛУБИНА РАССУЖДЕНИЙ

РЕЖИМ СЕССИИ

ultracode

Claude Code · v2.1.154

Включается через `/effort ultracode`. Закрепляет **effort = xhigh** для *всех* запросов сессии и включает авто-оркестрацию динамических воркфлоу.

Появился в **v2.1.154** (28 мая 2026) вместе с Opus 4.8. Открытый расход токенов — включайте осознанно.

Codex: аналога нет; параллелизм — вручную через /agent, /fork + /model.

```
# включить на всю сессию
/effort ultracode
→ effort = xhigh для всех
→ авто-оркестрация воркфлоу
⚠ открытый расход токенов
```

08

РАЗДЕЛ 08

Популярные MCP

context7 · github · sequential-thinking · playwright · memory · fetch

ПОПУЛЯРНЫЕ MCP

MCP-СЕРВЕР

context7

Актуальная документация библиотек прямо в контексте — без устаревших знаний модели.

Что даёт

Добавьте **use context7** в промпт — сервер вставляет свежую документацию нужной библиотеки. Работает с React, Next.js, Tailwind, Supabase и 1000+ пакетами.

Установка

```
claude mcp add context7 npx @upstash/context7-mcp
```

```
# добавить в промпт
Auth в Next.js App Router? use context7
→ подтянул Next.js 15 + Auth.js v5
→ актуальные примеры кода
```

ПОПУЛЯРНЫЕ MCP

MCP-СЕРВЕР

github

PR, issues, ревью, ветки — полноценная работа с GitHub прямо в сессии агента.

Что даёт

Динамические команды `/mcp__github__*` : создавать PR, листать issues, читать ревью, работать с ветками — без выхода из терминала.

Установка

```
/install-github-app (Claude Code) или через /mcp add-json
```

```
/mcp__github__list_prs
/mcp__github__create_pr
/mcp__github__get_issue 42
→ работает в обеих CLI
```

ПОПУЛЯРНЫЕ MCP

MCP-СЕРВЕР

playwright

Браузерная автоматизация прямо из агента: скриншоты, клики, заполнение форм, e2e-тесты.

Что даёт

Агент управляет браузером: открывает страницы, делает скриншоты, кликает элементы, заполняет формы и пишет e2e-тесты по увиденному.

Установка

```
claude mcp add playwright npx @playwright/mcp@latest
```

```
# агент сам откроет браузер
Зайди на localhost:3000,
сделай скриншот checkout-формы
и напиши тест на заполнение
→ screenshot + test.ts готов
```

ПОПУЛЯРНЫЕ MCP

MCP-СЕРВЕР

memory

Постоянная векторная память между сессиями — агент помнит решения, контекст и предпочтения.

Что даёт

Хранит сущности и факты в локальном векторном хранилище. Агент сохраняет ключевые решения и подтягивает их в следующих сессиях.

Установка

```
claude mcp add memory npx @modelcontextprotocol/server-memory
```

```
# агент сохраняет в память
Решили: auth через JWT, не sessions
# следующая сессия
Как сделать refresh token?
→ помнит про JWT из прошлого
```

Шпаргалка: задача → команда

Codex ↔ Claude Code

ЗАДАЧА	● CODEX	● CLAUDE CODE
Память проекта	`/init → AGENTS.md`	`/init → CLAUDE.md`
Цель / план	`/goal · /plan`	`/goal · /plan`
Контекст	`/compact · /status`	`/context · /compact`
Модель / effort	`/model · /fast`	`/model · /effort`
Макс. рассуждение	—	`ultrathink · /effort ultracode`
Ревью / дифф	`/review · /diff`	`/review · /code-review`
Субагенты	`/agent · /fork · /side`	`/agents · /branch`
Внешние инструменты	`/mcp`	`/mcp`
Откат	`/resume · /fork`	`/rewind · /undo`

Набирайте / — и сверяйтесь с источником

Состав команд зависит от версии CLI, модели и флагов. Источник истины — `/help`, официальная документация и changelog.