Codex CLI Cheat Sheet

Install & Login

Install Codex CLI:

```
npm i -g @openai/codex
# or
brew install --cask codex
```

Start Codex in the current repository:

```
codex
```

The first run prompts you to sign in with ChatGPT or use an API key. ChatGPT Plus, Pro, Business, Edu, and Enterprise plans include Codex access.

Tip: Keep the CLI current:

```
npm i -g @openai/codex@latest
codex update
codex --version
```

Models & Reasoning

Codex CLI:

```
/model      # Change model and, for supported models, reasoning level
/status     # Confirm active model, reasoning level, context, sandbox, approvals
/fast       # Toggle fast mode on supported models
```

Switch models and reasoning effort mid-session without losing context.

Command line:

```
codex --model gpt-5.5
codex -m gpt-5.4 "Review this repository"
codex -m gpt-5.5 -c model_reasoning_effort='"high"'
codex exec -m gpt-5.4 -c model_reasoning_effort='"low"' "Review this diff"
```

Model	Cost	Best For
GPT-5.5	highest	Hard coding tasks, large refactors, research-heavy work, architecture
GPT-5.4	balanced	Everyday feature work, debugging, reviews, multi-file edits
GPT-5.4 Mini	lower	Fast implementation, smaller fixes, subagents, codebase exploration
GPT-5.3-Codex / GPT-5.2-Codex	coding-specialized	Long-horizon agentic coding when available in your account
codex-mini-latest	fast	Lightweight local tasks and quick CLI automation

Configure the default in `~/.codex/config.toml`:

```
model = "gpt-5.5"
model_reasoning_effort = "medium"
model_reasoning_summary = "auto"
```

Setting	Values	Notes
`model_reasoning_effort`	`minimal`, `low`, `medium`, `high`, `xhigh`	Responses API only; supported models only; `xhigh` is model-dependent
`model_reasoning_summary`	`auto`, `concise`, `detailed`, `none`	Controls visible reasoning summaries, not how much reasoning the model does
`plan_mode_reasoning_effort`	`none`, `minimal`, `low`, `medium`, `high`, `xhigh`	Overrides reasoning only for Plan Mode

Effort	Use For
`minimal`	Fast formatting, small lookups, trivial edits
`low`	Reading config, simple fixes, command output summaries
`medium`	Default everyday coding and review work
`high`	Debugging, multi-file implementation, careful refactors
`xhigh`	Hard design/debugging work when the selected model supports it

Tip: Use `minimal` or `low` for cheap exploration, then raise effort once the target files and constraints are clear. Do not assume every model accepts every level.

Cost & Usage Control

Codex CLI:

```
/status       # Model, context, token usage, sandbox, approvals, writable roots
/fast         # Toggle fast mode on supported models
```

Main cost levers:

Lever	Lower Cost	Higher Capability
Model	Mini / smaller model	GPT-5.5 / Pro / Codex model
Reasoning	`minimal`, `low`, or `medium`	`high` or model-dependent `xhigh`
Context	Compact and reference fewer files	Large repo scans and long history
Subagents	Single-threaded run	Parallel workers and explorers

Tip: Subagents do their own model and tool work. They are useful for parallel review or implementation, but they cost more than a comparable single-agent run.