

Claude Code Cheat Sheet

Models

Claude Code:

```
/model [haiku|sonnet|opus]
```

Switch models on-the-fly without losing context.

Model	Cost	Best For
Haiku	cheapest	Fast screening, data extraction, classification, simple coding tasks
Sonnet	balanced	Balanced coding, writing, code review, refactoring
Opus	advanced	Complex debugging, deep research, architectural decisions, advanced problem-solving

Claude Code:

```
/fast
```

Toggles fast mode (Opus 4.6 only): ~2.5x faster output at a higher cost per token. Use for urgent tasks where speed matters more than cost.

Cost & Usage Control

Claude Code:

```
/usage # Session cost, plan usage limits, activity stats  
/cost # Alias for /usage  
/stats # Alias for /usage (opens on Stats tab)
```

The three main cost levers: model, effort and context (see further).

Tip: `HIGH/XHIGH/MAX` effort bills thinking tokens separately — they can exceed the response cost on complex tasks.

Thinking Effort

Claude Code:

```
/effort [low|medium|high|xhigh|max|auto]
```

Controls the token budget. Available levels depend on the model.

Level	Examples
<code>LOW</code>	<i>"What's the node version?", "Read this config file"</i>
<code>MEDIUM</code>	<i>"Add a helper function", "Rename this variable across files"</i>
<code>HIGH</code>	<i>"Implement this feature", "Debug this failing test"</i>
<code>XHIGH</code>	<i>"Refactor the entire auth module"</i>
<code>MAX</code>	<i>"Design the caching strategy" (session-only)</i>

Headless Mode

Run Claude Code without an interactive terminal:

```
claude -p "Write a commit message for these changes" < <(git diff --cached)
claude -p "Convert this CSV to JSON" < data.csv > data.json
claude -p "Check this PR for security issues" < <(gh pr diff 42)
cat src/**/*.*.ts | claude -p "Find any TODO comments and list them"
```

Structured output — use `--output-format json` for scripting:

```
claude -p "List imports in src/index.ts" --output-format json | jq '.result'
```