

Claude Code Cheat Sheet

Quick Reference - AIA Copilot Training 2026

Installation & Starting

```
npm install -g @anthropic-ai/claude-code
claude # Start interactive REPL
claude "explain project" # Start with prompt
claude -c # Continue recent conversation
claude --version # Check version
```

Essential Slash Commands

Command	Description
/help	Show all commands
/exit	End session
/clear	Clear history
/compact	Summarize conversation
/init	Create CLAUDE.md
/commit	Commit with AI message
/pr	Create pull request
/permissions	View/modify tool permissions
/cos	Show session cost & duration

Model Selection

```
claude --model sonnet # Balanced (default)
claude --model opus   # More capable
claude --model haiku  # Faster, cheaper
```

File References

```
@filename + Tab      # Autocomplete file reference
> Explain @src/auth.js # Reference specific file
> @package.json       # Reference config files
```

Thinking Modes

Phrase	Level	Use Case
"think"	Low	Simple analysis
"think hard"	Medium	Design decisions
"think harder"	High	Complex problems
"ultrathink"	Max	Critical architecture

Permission Management

```
/permissions          # View permissions
--allowedTools "Read,Write,Bash" # Allow specific tools
--disallowedTools "Bash(rm:*)"  # Block commands
--dangerously-skip-permissions  # ⚠ Skip all prompts
```

CLAUDE.md Template

```
# Project Name

## Commands
- npm run dev: Start dev server
- npm test: Run tests

## Tech Stack
- Node.js 20 + TypeScript
- PostgreSQL + Prisma

## Code Style
- Use async/await, never callbacks
- Prefer named exports
- JSDoc for all functions

## Workflow
- Run tests after changes
- Conventional commit messages
```

Subagents

```
@explore # Fast read-only search (Haiku)
@plan    # Research and planning (Sonnet)
@custom  # Your custom agents
```

Creating Custom Subagent

`.claude/agents/reviewer.md`:

```
---
name: reviewer
description: Code reviewer
```

```
tools: Read, Grep, Glob
model: sonnet
---
Review code for quality and security.
```

Custom Commands

Create `.claude/commands/test-gen.md` :

```
---
name: test-gen
description: Generate unit tests
---
Generate comprehensive tests for $ARGUMENTS
using Jest framework.
```

Usage: `/test-gen src/utils/validator.js`

Hooks (Security)

`.claude/hooks/pre-tool/no-secrets.sh` :

```
#!/bin/bash
if git diff --cached | grep -iE "(api_key|secret|token)"; then
  echo "❌ Secret detected!"
  exit 2 # Blocks the operation
fi
```

Make executable: `chmod +x .claude/hooks/pre-tool/no-secrets.sh`

Skills Marketplace (NEW)

```
/plugin marketplace add anthropics/skills
/plugins # Browse & install
```

Using skills:

- > Use document-skills to generate a project brief
- > Use frontend-design to create a landing page

Planning Mode ⚠️ CRITICAL

Always use for: live data, bulk ops, destructive actions

- > Plan this first, show steps, wait for approval

Why: 5 min planning saves 2 hours recovery

MCP Configuration

`.mcp.json` :

```
{
  "mcpServers": {
    "github": {
      "command": "mcp-server-github",
      "args": ["--token", "${GITHUB_TOKEN}"]
    }
  }
}
```

Common Workflows

Explore → Plan → Code → Commit:

1. > Overview of this codebase
2. > Plan how to add user preferences
3. > Implement the plan

```
4. > /commit
5. > /pr
```

TDD:

```
1. > Write failing tests for email validation
2. > /commit
3. > Implement to pass tests
4. > /commit
```

Output Formats

```
--output-format json          # For scripting
--output-format stream-json   # Real-time
--output-format text          # Default
```

Keyboard Shortcuts

Shortcut	Action
Tab	Autocomplete
Ctrl+C	Cancel
Ctrl+D	Exit
↑/↓	Command history

Troubleshooting

Issue	Fix
Command not found	Add npm bin to PATH
Auth fails	<code>claude login</code>

Issue	Fix
Slow responses	<code>/compact</code>
MCP errors	<code>claude --mcp-debug</code>

Best Practices

- ☐ Be specific in prompts
- ☐ Use @ mentions for files
- ☐ Review diffs before approving
- ☐ Use Planning Mode for live data
- ☐ Commit often (small commits)
- ☐ `/compact` for long sessions
- ☒ Don't skip reviews
- ☒ Don't commit secrets
- ☒ Don't use `--dangerously-skip-permissions`

Quick Tips

1. **Three-Layer Memory:** Global (`~/claude/CLAUDE.md`), Project (`./CLAUDE.md`), Reference (`@files`)
2. **Model Choice:** Haiku (explore), Sonnet (build), Opus (decide)
3. **Context Management:** `/clear` between tasks, `/compact` for long sessions
4. **Safety:** Planning Mode + Hooks + Git branches

Resources

- Docs: <https://code.claude.com/docs>
- MCP: <https://modelcontextprotocol.io>
- Community: <https://github.com/hesreallyhim/awesome-claude-code>