

Claude Code + VS Code Cheat Sheet

Purpose

Use this cheat sheet as a practical setup and workflow reference for Claude Code inside Visual Studio Code on Windows and Mac.

The goal is not only installation. The goal is a safer AI-assisted development workflow built around project memory, approvals, Git review, and repeatable implementation plans.

Setup Checklist

- Install VS Code and open the actual project folder, not a broad parent directory.
- Install Node.js LTS when using npm-based tooling or JavaScript and TypeScript projects.
- Install Claude Code using Anthropic's current recommended installer or approved team package path.
- Install the official Claude Code VS Code extension.
- Authenticate with the correct Anthropic account and organization workspace.
- Verify the terminal, extension panel, project files, and Git view are all accessible in VS Code.

Recommended VS Code Layout

- Explorer on the left for file review.
- Editor in the center for code, CLAUDE.md, and diffs.
- Claude panel on the right for analysis, plans, and implementation discussion.
- Integrated terminal at the bottom for approved commands.
- Git/source control view available before and after every AI-assisted edit.

Claude Code + VS Code Cheat Sheet

CLAUDE.md Essentials

- Document the tech stack, framework, package manager, and build commands.
- State coding conventions: TypeScript, component patterns, styling system, and naming rules.
- List guardrails: no new dependencies without approval, no secrets, no broad rewrites.
- Explain architecture boundaries and files that should be handled carefully.
- Require analyze, plan, approve, implement, review, and commit as the default workflow.
- Keep CLAUDE.md committed to the repo when it contains shared project standards.

Analyze -> Plan -> Approve -> Implement

- Analyze: ask Claude Code to inspect the repository and explain relevant files before editing.
- Plan: require files to touch, risks, expected behavior, tests, and rollback considerations.
- Approve: review file edits, file creation, terminal commands, dependency changes, and context access.
- Implement: apply the smallest useful change and keep diffs reviewable.
- Review: run lint, type checks, tests, and builds before treating work as complete.
- Commit: use Git only after human review confirms the change is correct.

Approval Rules

- Read terminal commands before approving them.
- Be careful with commands that install packages, alter Git state, delete files, or contact external services.
- Do not approve broad folder access unless the task requires it.
- Do not disable review processes for production repositories.
- Approvals are governance controls, not setup friction.

Claude Code + VS Code Cheat Sheet

Beginner Mistakes to Avoid

- Starting with 'build me an app' before Claude understands the repository.
- Skipping CLAUDE.md and expecting consistent behavior.
- Approving commands without reading them.
- Installing new dependencies before checking existing project tools.
- Working in a dirty Git tree without reviewing current changes.
- Deploying AI-generated code without tests, builds, and human review.

Troubleshooting Quick Fixes

- Claude command not recognized: restart the terminal and confirm the install path is on PATH.
- Node not found: install Node.js LTS and reopen VS Code.
- Authentication fails: verify the correct Anthropic account, workspace, and plan access.
- Browser login does not open: copy the login URL manually if provided and use the approved browser.
- Extension not responding: reload VS Code, trust the workspace, and confirm authentication.
- Permission errors: move the repo to a writable development folder and retry with scoped access.

Enterprise Recommendation

Teams should standardize Claude Code with shared CLAUDE.md files, approved install paths, branch rules, dependency policy, review expectations, and CI checks.

Claude Code should accelerate developers while preserving engineering ownership. Use it to analyze, plan, implement, and document, but keep humans responsible for architecture, security, maintainability, and release quality.