

# Claude Code & Sonnet 4.5

Latest Features and Usage Guide

October 2025

## Table of Contents

1. Introduction
2. Claude Sonnet 4.5 Overview
3. Claude Sonnet 4.5 Key Features
4. Claude Code Overview
5. Claude Code Installation
6. Claude Code Key Features
7. How to Use Claude Code
8. Best Practices and Tips

# 1. Introduction

This guide provides a comprehensive overview of Claude Sonnet 4.5 and Claude Code, including their latest features, capabilities, and practical usage instructions. Released in September 2025, these tools represent significant advances in AI-assisted development.

## 2. Claude Sonnet 4.5 Overview

Claude Sonnet 4.5 is Anthropic's most intelligent AI model, specifically optimized for coding and building complex agents. It represents a major leap forward in autonomous operation and software development capabilities.

### Key Highlights:

- State-of-the-art coding performance on SWE-bench Verified
- Can run autonomously for up to 30 hours on complex tasks
- 61.4% success rate on OSWorld (real-world computer tasks)
- Same pricing as Sonnet 4: \$3/\$15 per million tokens
- Available on Claude API, Amazon Bedrock, and Google Cloud Vertex AI

## 3. Claude Sonnet 4.5 Key Features

### 3.1 Advanced Coding Capabilities

- Best-in-class code generation and analysis
- Enhanced system design and planning
- Improved security engineering
- Excels at autonomous long-horizon coding tasks
- Superior performance on real-world software coding challenges

### 3.2 Extended Autonomous Operation

- Can work independently for 30 hours vs. 7 hours for Opus 4
- Maintains focus on complex, multi-step tasks throughout extended sessions
- Effective planning and execution for projects spanning hours or days

### 3.3 Agent Building Excellence

- Strongest model for building complex, independent agents
- Best model at using computers autonomously
- Advanced context management across sessions
- More effective parallel tool usage

### **3.4 Memory and Context Features (Beta)**

- New memory capability stores information outside context window
- Remembers preferences and context across conversations
- Automatic cleanup of tool interaction history
- Intelligent context management to prevent token waste

### **3.5 Improved Safety and Alignment**

- Most aligned frontier model from Anthropic
- Reduced sycophancy, deception, and power-seeking behaviors
- Better defense against prompt injection attacks
- Improved behavior to discourage delusional thinking

### **3.6 Specialized Capabilities**

- Substantial gains in reasoning and mathematics
- Excel in cybersecurity applications
- Strong performance in finance and research domains
- Enhanced computer use capabilities

## 4. Claude Code Overview

Claude Code is an agentic coding tool that lives in your terminal and IDE, helping developers code faster through natural language commands. It understands your codebase and executes routine tasks, explains complex code, and handles git workflows seamlessly.

## 5. Claude Code Installation

### 5.1 Terminal Installation

Install Claude Code globally using npm:

```
npm install -g @anthropic-ai/claude-code
```

After installation, navigate to your project directory and run:

```
claude
```

### 5.2 VS Code Extension (Beta)

- Native VS Code extension available in beta
- Brings Claude Code directly into your IDE
- Real-time changes visible through dedicated sidebar panel
- Inline diffs for easy code review

## 6. Claude Code Key Features

### 6.1 Checkpoints System

- Automatically saves code state before each change
- Instant rollback to previous versions
- Tap Esc twice or use /rewind command to revert
- Enables ambitious tasks with safety net

### 6.2 Enhanced Terminal Interface (v2.0)

- Improved status visibility
- Searchable prompt history (Ctrl+r)
- Easy reuse and editing of previous prompts

- Better message rendering with large contexts

## 6.3 Advanced Capabilities

- Queue messages while Claude is working
- Drag-and-drop or copy/paste image files into prompts
- MCP "project" scope for repository-level configuration
- Thinking mode support for complex reasoning
- @-mentions support in slash command arguments

## 6.4 Core Functionality

- Natural language command execution
- Codebase understanding and navigation
- Complex code explanation
- Git workflow automation
- Routine task execution

## 7. How to Use Claude Code

### 7.1 Basic Usage

Claude Code operates through natural language commands. Simply describe what you want to accomplish:

**Examples:**

- "Add a dark mode toggle to the settings page"
- "Refactor the authentication module to use async/await"
- "Explain how the payment processing works"
- "Fix the TypeScript errors in the user service"

### 7.2 Git Workflows

Claude Code can handle complex git operations:

- Create commits with descriptive messages
- Create and manage branches
- Handle pull requests
- Resolve merge conflicts

### 7.3 IDE Integration

- Use the VS Code extension for visual feedback
- Review changes through inline diffs
- Monitor Claude's progress in the sidebar panel
- Tag @claude in GitHub for assistance

### 7.4 Slash Commands

Claude Code supports custom slash commands for common operations. Use the /rewind command to undo recent changes.

## 8. Best Practices and Tips

### 8.1 For Claude Sonnet 4.5

- Enable extended thinking for complex coding tasks
- Be aware that thinking mode impacts prompt caching efficiency
- Use memory features to maintain context across sessions
- Leverage autonomous operation for long-horizon tasks

- Take advantage of improved parallel tool usage

## 8.2 For Claude Code

- Use checkpoints before attempting major refactoring
- Search prompt history (Ctrl+r) to reuse successful commands
- Be specific with natural language instructions
- Review changes through the VS Code extension before committing
- Use @-mentions for targeted assistance
- Queue up multiple tasks when appropriate

## 8.3 General Tips

- Start with smaller tasks to understand capabilities
- Provide context about your codebase architecture
- Use descriptive commit messages for better history
- Leverage thinking mode for complex problem-solving
- Join the Discord community for support and tips