

CLAUDE CODE: FROM COPILOT TO SHIPPING FEATURES.

```
~/DANIC-AI-OS · ZSH  
  
$ npm  
$  
$
```

An agentic coding tool that lives in the terminal — reads, writes, and runs code with you. Goes beyond autocomplete: understands your **full codebase** and executes **multi-step tasks**. Built on Claude's reasoning, designed for engineers who want a **collaborator**, not a copilot.

01 What is Claude Code

FOUNDATION

Copilot → Suggests code

Claude Code → Ships features

Modes: Plan · Investigate · Verify

02 Core Primitives

6 BUILDING BLOCKS

03 Installation & Setup

5 STEPS · ~3 MIN

01 · INSTALL

Install CLI

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

04 · RUN

02 · AUTH

Authenticate

```
claude login
```

03 · CD

Enter Project

```
cd ./your-project
```

05 · MEMORY

Launch Claude

cLaude

Create CLAUDE.md

/init

04

The Claude Code Stack

4 LAYERS

05

Agentic Patterns

6 PATTERNS

06 Building with Claude Code

8-STEP WORKFLOW

01

Define the goal in plain English

State a clear outcome — not a technical pseudo-spec.

02

Write a CLAUDE.md with conventions

Coding standards, tech stack, boundaries as project memory.

03

Let Claude explore the codebase

Read-first: understand before changing.

04

Approve the plan before execution

Plan mode as a gate — review before write access.

05

Add Skills for repeated workflows

Encapsulate recurrence — composable expertise.

06

Add Hooks for automation

Lifecycle automation: format, lint, guardrails.

07

Run, verify, iterate

Feedback loop: test → critique → fix.

08

Commit with co-author tag

Auditable trail: `Co-authored-by: Claude`.

- SECURITY & TRUST

- Permission prompts
- Sandboxed execution
- Audit trail
- Read-before-write

- OBSERVABILITY

- Session transcripts
- Tool-call logs
- Cost tracking
- Token usage per session

```
SKILL.MD

---
name:      pdf-extractor
description: Extract text and tables from PDF files
---

# Body markdown – instructions for Claude
When the user uploads a PDF, follow these steps:
1. Inspect the PDF structure
2. Choose the right extraction tool
3. Validate output schema
4. Return clean, structured data
# Optional scripts & resources
# Loaded only when relevant
```

• ANATOMY OF A SKILL

- YAML frontmatter (name + description)
- Body markdown (instructions for Claude)
- Optional scripts and resources
- Loaded only when relevant

PDF

Excel

Document

Frontend Design

Custom Team

Patterns

COORDINATION MODELS

- Orchestrator — Worker
- Specialist Subagents
- Parallel Exploration
- Critic — Editor pair
- Researcher — Builder

U

Use Cases

WHEN IT PAYS OFF

- Large refactors
- Codebase migrations
- Test-suite generation
- Documentation passes
- Multi-repo coordination

09

Evaluating Claude Code Sessions

METRICS & TOOLING

QUALITY

Task success rate %

PR merge rate %

Test pass rate

%

Code-review feedback

qual

EFFICIENCY

Tokens per task

tok

Wall-clock time

sec

Tool calls per task

n

Cost per shipped feature

EUR

10 Learning Resources

VERIFIED · OFFICIAL + ECOSYSTEM

