Using Claude Code for Research and Business Tasks

Video: Using Claude Code for non-coding tasks

Overview

This guide demonstrates how to use Claude Code as an AI agent for research and business tasks beyond traditional coding—including market research, document analysis, and competitive intelligence.

Prerequisites

1. Claude Account

You'll need a Claude account (a paid Pro account provides more capacity). Access Claude Code settings through: **Settings** → **Code**

2. Install Claude Code

Follow the installation instructions in your Claude settings.

Windows users: A community-created installation guide is available for Windows 10 users (link provided in the video comments).

3. Visual Studio Code (VS Code)

Download and install VS Code from code.visualstudio.com. Available for Windows, Linux, and Mac. VS Code serves as your text editor and provides the interface for working with Claude Code.

4. Understanding Markdown (Optional but Helpful)

Markdown is a simple markup language used for formatting text. Most AI outputs use Markdown for structure (headings, bold text, lists, etc.). A Markdown cheat sheet is useful for reference.

How It Works

Project-Based Approach

Claude Code operates on a project basis within specific folders. Each project maintains its own memory and context, completely isolated from other projects. This allows you to:

- Organize multiple concurrent projects
- Keep research contexts separate
- Focus Claude on specific document sets

The Memory System

When you initialize Claude Code in a folder, it creates a `claude.md` file that serves as its memory for that project. This file is automatically generated and maintained by Claude Code.

Practical Example 1: EU Tender Analysis

Project Structure:

```
None
tender-folder/
├── tender-guide.pdf
├── tender-text.md (the actual tender in Markdown format)
└── tender-preparation.md (your instructions to Claude)
```

The Instruction File (`tender-preparation.md`):

- Clear title and structure
- Description of what's in the folder
- Your job definition for Claude (e.g., "Review this tender and assess fit for my company")
- Background context about your company and team

Running the Analysis:

1. Open the folder in VS Code
2. Open Terminal (within VS Code)
3. Type `code` to start Claude Code
4. Ask Claude to summarize or analyze the folder contents

Claude will read all documents and provide analysis based on your instructions and the tender content.

Practical Example 2: Market Research

Use Case: Researching competing platforms for an intra-logistics equipment marketplace.

Project Structure:

```
None
market-research/
├── instructions.md (brief overview of the project)
├── project-context.md (detailed background)
└── market-landscape.md (list of potential competitors)
```

Process:

1. **Initialize the project:**

```
None
code
```

2. Claude Code will read all files and create its memory file.

3. **Approval settings:**

    - Option 1: Approve each action individually
    - Option 2: Approve all (don't ask again)
    - Option 3: Change the plan
4. **Give clear instructions via prompt:** Example: "Go to the market-landscape file, extract the first three companies, research them online, and create a research dossier file with your findings. Ask me if you have questions before proceeding."

5. **Claude Code will:**

    - Parse your request
    - Identify the companies
    - Conduct web searches
    - Compile findings
    - Create a new markdown file with the research results

Key Benefits

- **Automated research**: Claude handles repetitive research tasks while you focus on other work
- **Structured output**: Results are saved in organized, readable formats
- **Context isolation**: Each project maintains separate context and memory

- **Scalability**: Process multiple items (e.g., dozens of companies) by letting Claude work in the background

Tips for Success

1. **Be clear and specific** in your instruction files
2. **Provide context**: The more background you give Claude, the better its analysis
3. **Use repetition** in prompts when needed to ensure understanding
4. **Ask Claude to ask questions** before proceeding if you want to ensure alignment
5. **Organize by project**: Keep related documents in dedicated folders

Workflow Summary

1. Create a project folder with your documents
2. Write an instruction file explaining the task
3. Open the folder in VS Code
4. Open Terminal and run `code`
5. Interact with Claude Code via text prompts
6. Review and use the generated outputs

Some useful links:

1) installing Claude Code (Mac and Linux):
https://docs.claude.com/en/docs/claude-code/overview

2) Installing Claude Code in Windows:
https://claude.ai/public/artifacts/8ad38c46-8b37-4f75-99ab-6f695ccac6f4

3) Visual Studio: https://code.visualstudio.com/Download

4) Tiny basics of Markdown (I show this in the video):
https://www.markdownguide.org/cheat-sheet/

5) An excellent course for anyone who wants to go deeper into Claude Code:
https://learn.deeplearning.ai/courses/claude-code-a-highly-agentic-coding-assistant/lesson/6b35/introduction

---

This guide accompanies the video demonstration of Claude Code for research and business applications.