Week 2: Tool Calling - How Claude Takes Action

01. Session Goals

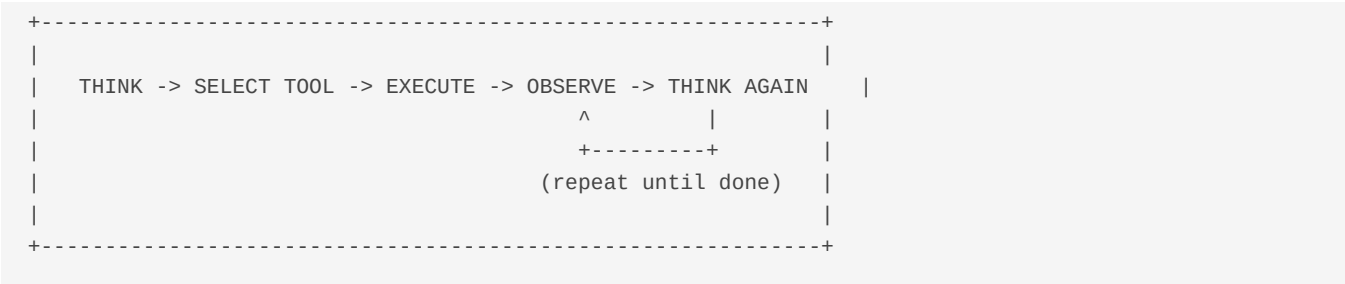
- Understand how Claude uses tools to interact with the world
- Learn the tool calling loop: think, select, execute, observe
- Practice with built-in tools for data analysis and research
- Build a research workflow using web search and data fetching

02. Block 1: Theory - How Tool Calling Works (30 min)

The Key Insight
Last week we learned that Claude Code bets on filesystems and bash. This week we go deeper: how does tool calling actually work?

When you give Claude access to tools, it stops being a chatbot and becomes an agent. It can read files, search the web, and execute commands. But the most important tool is one you might not expect: bash.

The Tool Calling Loop
Every time Claude uses a tool, it follows this loop:



Example: "Analyze the top 10 customers by revenue from this CSV"

1. Think: I need to read the CSV file first
2. Select: Use the `Read` tool
3. Execute: Read the file contents
4. Observe: I see 500 rows with columns: customer_name, revenue, date...
5. Think: Now I need to sort by revenue and take top 10
6. Select: Use `Bash` to sort and filter

7. Execute: Run the command
8. Observe: Here are the results...
9. Think: Task complete, present findings

Why Bash Is the Most Important Tool

Remember Vercel's lesson from Week 1: they removed 80% of their agent's tools and got better results. What did they keep? Bash.

Here's what Vercel removed from their text-to-SQL agent:

- Schema lookup tool
- Query validation tool
- Error recovery tool
- Entity join finder
- Context recall mechanism
- Results formatter
- Data visualization tool
- ...and 5 more specialized tools

What replaced all of them? An agent that can execute bash commands and explore files.

Why this works:

LLMs have seen grep, cat, find, and ls billions of times during training. These commands are native operations, not bolted-on behaviors. When you give Claude bash access, you're not teaching it something new. You're letting it use skills it already has.

How agents use bash:

Command	What It Does	Agent Use Case
`ls`	List files	"What data do I have to work with?"
`cat`	Read file contents	"Let me look at this file"
`grep`	Search for patterns	"Find all mentions of 'pricing objection'"
`find`	Locate files	"Where are the Q4 reports?"
`head/tail`	Preview files	"Show me the first 10 rows"
`wc`	Count lines/words	"How many records are in this file?"
`sort`	Order data	"Sort by revenue descending"
`uniq`	Find unique values	"What industries are represented?"

On-demand context retrieval:

Instead of stuffing everything into the prompt upfront, agents retrieve context as they need it:

```
Agent receives task
    -> Explores filesystem (ls, find)
    -> Searches for relevant content (grep)
    -> Reads specific files (cat)
    -> Sends only what's needed to the model
    -> Returns structured output
```

This is why Vercel's agent dropped from $1.00 to $0.25 per sales call analysis. It loads only what it needs, when it needs it.

The philosophy:

> "Every agent needs filesystem and bash. If you're building an agent, resist the urge to create custom tools. Instead, ask: can I represent this as files?"

Built-in Tools
Claude Code comes with these tools ready to use:

Tool	What It Does	Business Use Cases
Read	Read files	Analyze CSVs, read reports, review documents
Write	Create files	Generate reports, save analysis results
Edit	Modify files	Update data, fix errors in documents
Bash	Run commands	Execute scripts, process data
Glob	Find files	Locate reports, find datasets
Grep	Search content	Find mentions, search across files
WebSearch	Search the web	Research companies, find market data
WebFetch	Fetch web pages	Get company info, pull public data

How Claude Decides Which Tool to Use
Claude reads your request and matches it to available tools:

Your Request	Claude's Thinking	Tool Selected
"What's in this CSV?"	Need to read a file	Read
"Find all mentions of 'revenue'"	Need to search content	Grep (or Bash with grep)
"Research Acme Corp"	Need current web info	WebSearch
"Create a summary report"	Need to write output	Write
"How many rows have status 'closed'?"	Need to filter and count	Bash (`grep "closed" \ wc -l`)
"Sort leads by score"	Need to reorder data	Bash (`sort -t',' -k3 -rn`)

Bash vs. Other Approaches
Why not just use RAG (retrieval augmented generation) or vector search?

Approach	How It Works	Problem
Prompt stuffing	Load everything into context	Hits token limits fast
Vector search	Find semantically similar chunks	Imprecise for structured data
Bash + filesystem	Exact pattern matching, on-demand	Precise, efficient, debuggable

When you need "all deals over $50K in Q4", vector search might return similar-sounding content. Bash gives you exactly what you asked for:

```
grep "2024-Q4" deals.csv | awk -F',' '$3 > 50000'
```

Claude knows how to write these commands. Let it.

Demo: Watch Claude Work
Live demo with this prompt:

```
Look at the sample-leads.csv file. Tell me:
1. How many leads are there?
2. Which industries are represented?
3. Who are the top 3 leads by score?
4. What's the average company size?
```

Watch how Claude:

1. Uses Read to load the CSV
2. Reasons about the data structure
3. Uses Bash commands like `sort`, `grep`, `wc` to analyze
4. Presents structured findings

---

03. Block 2: Lab 1 - Exploring Built-in Tools (30 min)

Task: Data Exploration with Claude
Use Claude Code to explore the sample data in this repo.

Exercise 1: File Discovery (5 min)

```
> What data files are available in this repository? List them with their sizes.
```

Notice: Claude uses Glob to find files.

Exercise 2: Data Profiling (10 min)

```
> Read the sample-leads.csv file and give me a complete profile:
> - Total rows and columns
> - Column names and what they contain
> - Any missing or unusual values
> - Distribution of the 'status' field
```

Notice: Claude uses Read then reasons about the data.

Exercise 3: Data Analysis (10 min)

```
> From sample-leads.csv, answer these business questions:
> 1. What percentage of leads are in Technology vs other industries?
> 2. What's the correlation between company size and lead score?
> 3. Which lead sources produce the highest-scoring leads?
```

Notice: Claude may use Bash to run calculations.

Exercise 4: Cross-file Analysis (5 min)

```
> Compare sample-leads.csv with mock-crm.json.
> Which leads from the CSV also appear in the CRM contacts?
> What additional info does the CRM have about them?
```

Notice: Claude reads multiple files and synthesizes.

Discussion Questions
1. Which tools did Claude use for each task?
2. Did Claude ever use multiple tools in sequence?
3. How did Claude handle the analysis - code or reasoning?

---

04. BREAK (10 min)

05. Block 3: Theory - Web Tools for Research (30 min)

Why Web Tools Matter
Your data lives in files. But context lives on the web:

- Company websites
- News articles
- LinkedIn profiles
- Industry reports
- Competitor information

Claude's web tools bridge the gap between your internal data and external context.

WebSearch: Finding Information
WebSearch queries the web and returns relevant results.

Good prompts for WebSearch:

```
> Search for recent news about Acme Corp funding
> Find information about the CRM software market size 2024
> Look up reviews of competitor product X
```

What you get back:

- Search result titles and snippets
- URLs to sources
- Dates when available

WebFetch: Getting Page Content
WebFetch retrieves and reads a specific URL.

When to use WebFetch:

- You have a specific URL to read
- You need detailed content from a page
- You want to extract structured data from a site

Example flow:

1. WebSearch → find relevant URLs
2. WebFetch → get detailed content from best result
3. Analyze → synthesize findings

Combining Tools for Research
Company Research Workflow:

```
User: "Research TechCorp for our sales call tomorrow"

Claude's process:
1. WebSearch: "TechCorp company news 2024"
2. WebFetch: Company about page
3. WebSearch: "TechCorp leadership team"
4. WebFetch: LinkedIn or team page
5. Synthesize: Compile research brief
6. Write: Save to file
```

Practical Research Patterns

Research Goal	Tool Sequence
Company overview	WebSearch → WebFetch company site
Recent news	WebSearch with date filter
Competitive analysis	WebSearch competitors → WebFetch each
Market sizing	WebSearch industry reports → WebFetch
Contact research	WebSearch person + company → synthesize

Beyond Built-in: External Web Services

Claude's built-in WebSearch and WebFetch cover most needs. But when you need more control, there are specialized services:

Service	What It Does	When to Use
Tavily	AI-native search API	When you need structured search results optimized for agents
Firecrawl	Web scraping and crawling	When you need to extract data from complex sites
Bright Data	Web scraping with anti-bot	When sites block automated access
Google Search Grounding	Official Google search data	When you need Google-quality results

Tavily was built specifically for AI agents. It returns clean, structured data instead of raw HTML. Free tier includes 1,000 API credits per month.

Firecrawl converts websites into LLM-ready markdown. Useful for extracting content from JavaScript-heavy sites that WebFetch can't handle.

Bright Data offers an MCP server that handles anti-bot measures, CAPTCHAs, and rate limiting. Best for large-scale scraping or sites that actively block automation.

These services connect to Claude through MCP servers. We'll cover how to build custom tools in Week 6 (Agent SDK), but here's the key idea: any API can become a Claude tool by wrapping it in an MCP server. You define the tool schema with inputs and outputs, and Claude learns when to use it.

Limitations to Know

- Rate limits: Don't hammer with requests
- Paywalled content: Can't access subscription sites
- Dynamic content: Some sites don't render well
- Accuracy: Web data may be outdated

---

06. Block 4: Lab 2 - Building a Research Workflow (45 min)

Task: Company Research for GTM
Build a research workflow that prepares you for sales calls.

Scenario: You have a call with three companies tomorrow. Use Claude to research each one.

Step 1: Single Company Research (15 min)
Pick a real company (or use "Stripe" as example):

```
> Research Stripe for a sales call. I need:
> 1. What they do (1-2 sentences)
> 2. Company size and headquarters
> 3. Recent news (last 3 months)
> 4. Key products or services
> 5. Potential pain points we could address
```

Watch Claude combine WebSearch and WebFetch.

Step 2: Structured Output (10 min)
Ask Claude to format the research:

```
> Take your research on Stripe and format it as a pre-call brief.
> Use this structure:
> - Company Snapshot (name, size, industry)
> - What They Do
> - Recent Developments
> - Talking Points for Our Call
> Save it to output/stripe-research.md
```

Step 3: Batch Research (15 min)
Now research multiple companies:

```
> Research these 3 companies from our sample-leads.csv:
> 1. Acme Corp
> 2. GlobalTech Inc
> 3. HealthFirst Solutions
>
> For each, provide:
> - What they do
> - Industry and size
> - One interesting fact from recent news
>
> Format as a markdown table.
```

Step 4: Connect to Your Data (5 min)
Combine web research with your lead data:

```
> For the top 5 leads by score in sample-leads.csv:
> - Look up each company briefly
> - Add a "research note" based on what you find
> - Save the enriched list to output/enriched-leads.md
```

Deliverable
By end of lab, you should have:

- [ ] Researched at least one company in depth
- [ ] Created a formatted pre-call brief
- [ ] Batch researched multiple companies
- [ ] Connected web research to your lead data

---

07. Wrap-Up (15 min)

Key Takeaways
1. Claude is an agent, not a chatbot - It uses tools to take action
2. The loop: Think → Select Tool → Execute → Observe → Repeat
3. Built-in tools: Read, Write, Bash, Glob, Grep, WebSearch, WebFetch
4. Web tools unlock context - Bridge your data with external information
5. Combine tools for workflows - Search → Fetch → Analyze → Write

What We Didn't Cover (But You Can Explore)

- Custom tools via MCP - Next week's topic
- Tool schemas - How tools define their inputs
- Error handling - What happens when tools fail
- Permissions - Controlling what Claude can do

Homework

Research Assignment:

1. Pick 5 entities relevant to your project:

Project Domain	Research Targets
GTM/Sales	Prospects, competitors, or partners
Developer Tools	Open source projects, tech companies, tools
Content/Marketing	Influencers, publications, trending topics
Customer Support	Product documentation, competitor FAQs, forums
Operations	Vendors, compliance requirements, industry benchmarks

2. Use Claude to research each one. Document:
   - Which tools Claude used
   - How long each research took
   - Quality of information found
   - Any gaps or limitations
3. Create a "Research Playbook" with your best prompts:
   - What prompt works best for overview/context?
   - What prompt works best for recent updates?
   - What prompt works best for detailed analysis?
4. Bring your playbook to Week 3 - we'll enhance it with MCP integrations.

Next Week Preview

Week 3: MCP Integration

- Connect Claude to your databases, APIs, and services
- Build persistent integrations (not just web searches)
- Access private data sources
- Create connected workflows for your domain

---

08. Facilitator Notes

Common Issues

1. WebSearch returns no results: Check query specificity, try different terms
2. WebFetch fails: Some sites block automated access, try alternatives
3. Slow responses: Web tools take longer than file tools, set expectations
4. Rate limiting: If hitting limits, add delays between requests

Timing Adjustments

- Lab 1 can be shortened if concepts click quickly
- Lab 2 Step 3 (batch research) can become homework if time-constrained
- Focus on ensuring everyone completes Step 1 and Step 2

Discussion Prompts

- "What surprised you about how Claude chose tools?"
- "Where do you see tool calling being useful in your work?"
- "What research tasks do you do manually that Claude could help with?"

Energy Management

- Web tools are slower - keep energy up during waits
- Have participants share interesting findings during labs
- Celebrate good research prompts when you see them