

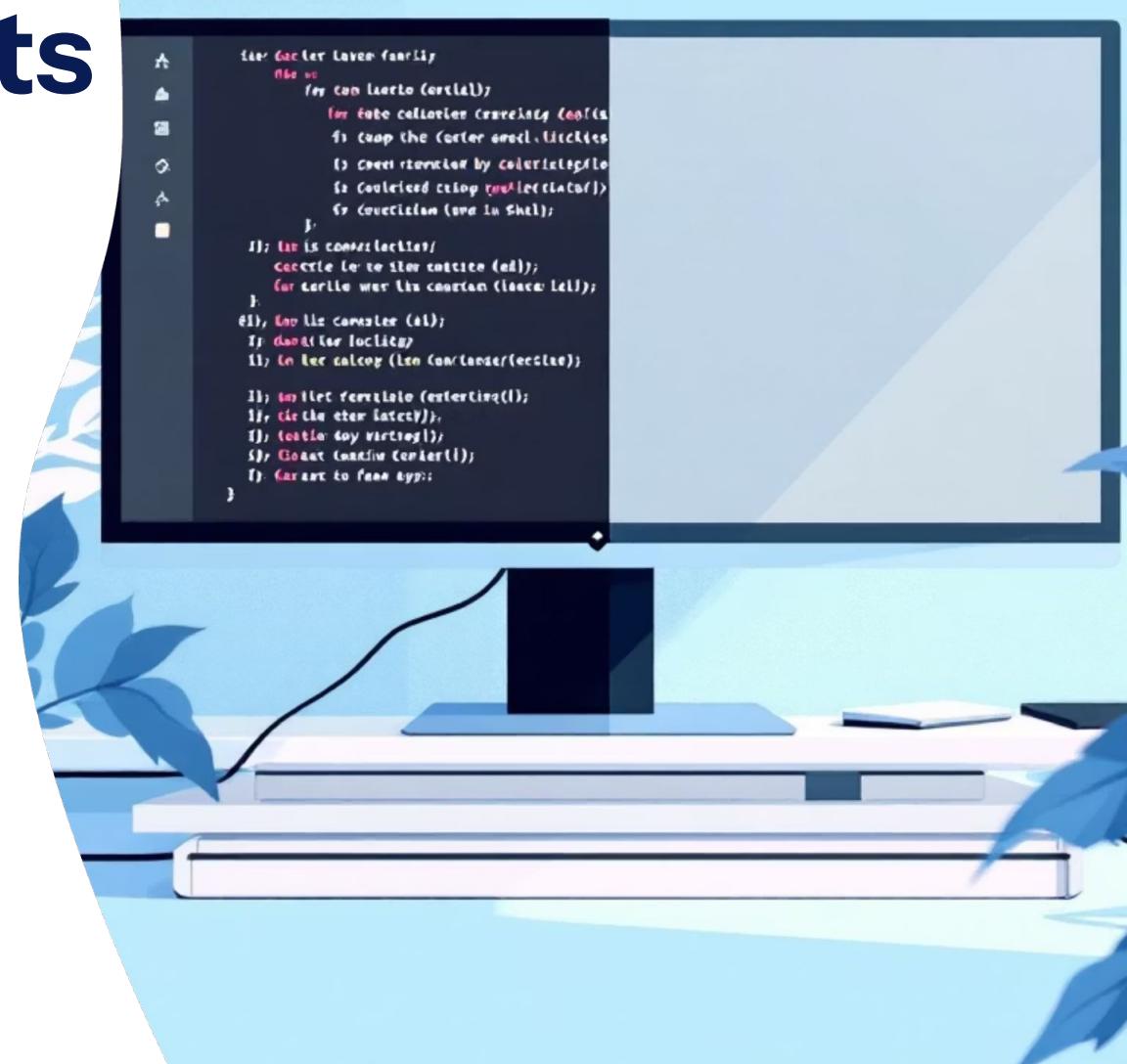
Slash Commands vs Subagents

Understanding two powerful approaches to workflow automation in Claude: when to use simple shortcuts versus independent processing

```
for Doctor's office family
  if &lt; user &gt;
    for can doctor (certified)
      for todo collection (traveling (dest))
        for trip the doctor abroad (tickets)
          for cover insurance by creditcard (file)
          for collected expense (travel (status))
          for connection (done in shell)
        }
      }
    }
  }
}

if &lt; user is connected /&gt;
  connect to the user connection (id);
  for certif who has the connection (name: id);
}

End the connection (id);
For doctor (for locality)
  In the location (lon) (lat) (radius)
    for outlet (outlet (extinguish (l)));
    for the item (status));
    for outlet day (working);
    for (agent (name (center (l)));
    for agent to move app;
```



Slash Commands: Your Quick Shortcuts

The Basics

Pre-written prompt templates stored as markdown files that provide instant access to common tasks. Think of them as saved snippets that execute with a single command.

Storage location: .claude/commands/ directory

File format: .md (markdown)

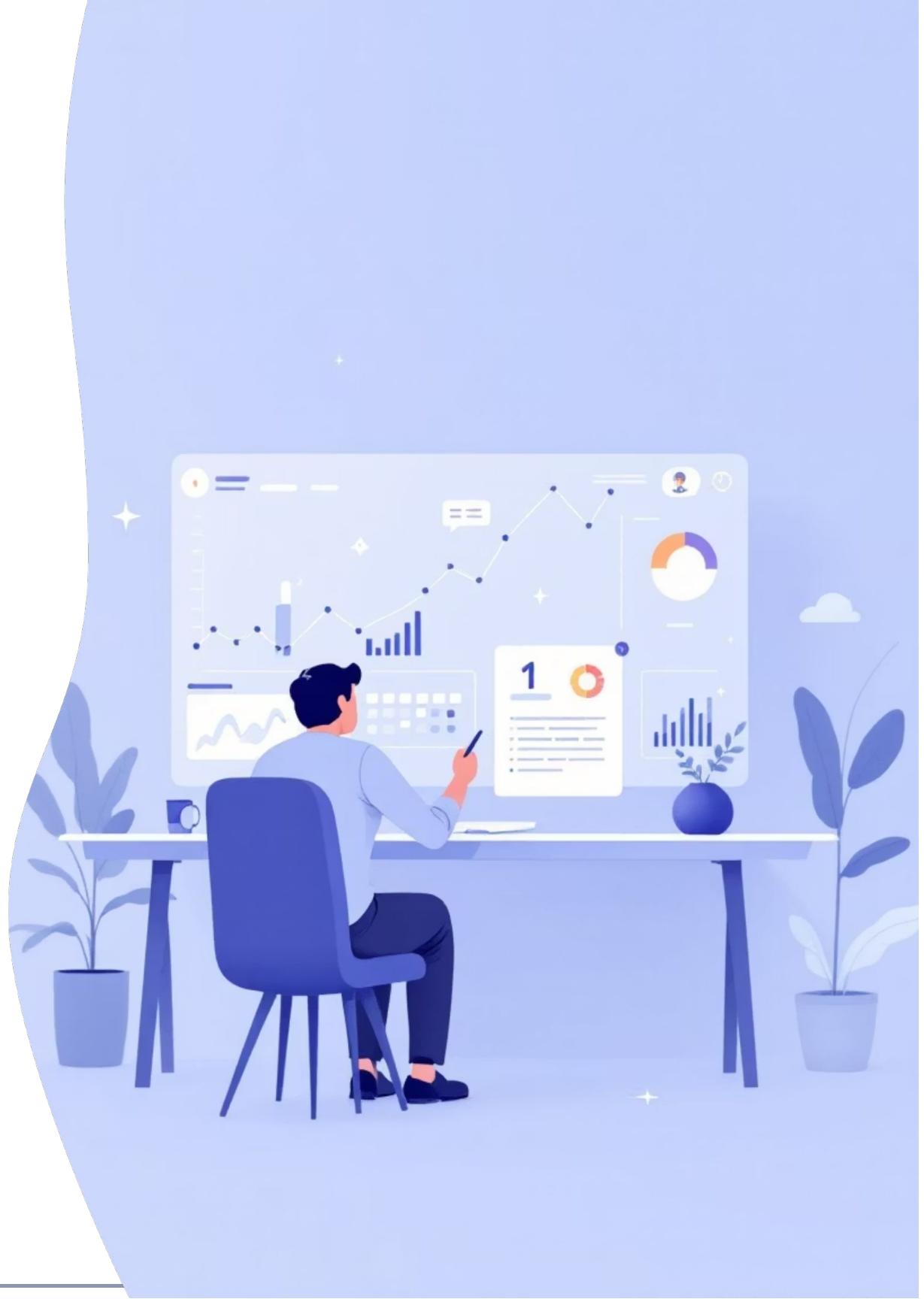
How They Work

- Stored as simple text templates in your project
- Sent directly to the main Claude instance
- Share the same conversation context
- Support dynamic inputs via \$ARGUMENTS placeholder

Example: /fix-lint

Please fix all linting errors in the codebase:

1. Run the linter
2. Fix any issues found
3. Commit the changes



Slash Commands: Strengths & Limitations



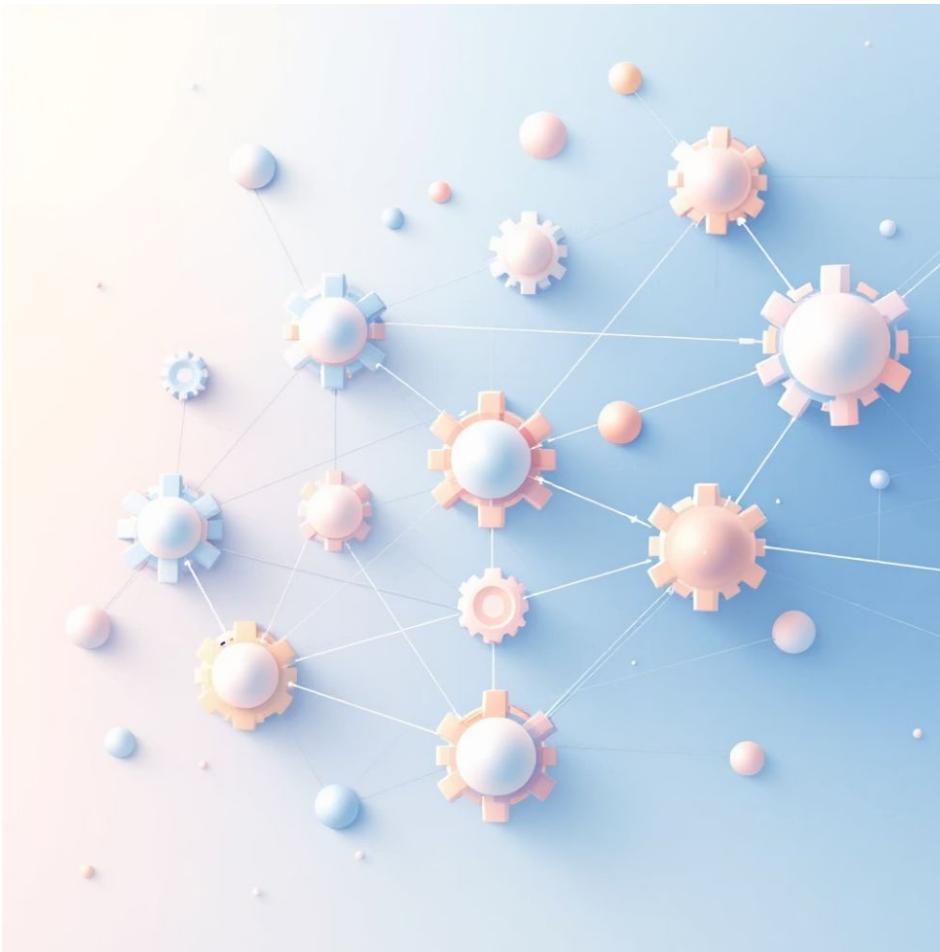
✓ Advantages

- Simple text-based prompts—easy to create and maintain
- Fast execution with no additional overhead
- No extra API calls required
- Perfect for repetitive workflows

✗ Trade-offs

- No isolated context—everything shares main conversation
- Can't run tasks in parallel
- Consumes your main conversation tokens
- Limited to template-based execution

Subagents: Independent Processing Power



What Are Subagents?

Separate Claude instances that run independently from your main conversation, allowing for parallel processing and context isolation.

Core Purpose

Delegate specific research or verification tasks while preserving your main conversation context and token budget.

The Workflow

1. Main Claude

"Let me use a subagent to verify the API documentation..."

3. Research & Analysis

Subagent reads docs, analyzes patterns, investigates details

2. Subagent Spawns

New Claude session starts with isolated context

4. Results Return

Main Claude receives findings and continues work

Subagents: Capabilities & Considerations

Key Strengths

Separate Execution Context

Isolated environment prevents context pollution

Parallel Processing

Multiple tasks run simultaneously

Context Preservation

Main conversation stays clean and focused

Deep Research

Ideal for thorough investigation and verification

Important Trade-offs

- ❑ **Additional API Calls:** Each subagent spawns a separate Claude instance, increasing API usage
- ❑ **Higher Cost:** Running parallel instances uses more resources than single-instance commands

