T02 Technical Intermediate

Non-Interactive & Headless Mode

Using Claude Code without human interaction: scripts, CI, pipes

Base flag: `-p`

The `-p` flag (or `--print`) activates non-interactive mode. Claude returns the response in the terminal and exits immediately without waiting for user input.

```
# Direct prompt
claude -p "Analyze this file and list the bugs"

# With pipe stdin (content becomes the input)
git diff | claude -p "Explain these changes"
cat error.log | claude -p "Identify the root
cause"
```

Output formats

`--output-format` controls the response structure, which is essential for integration into scripts.

Format	Usage
`text`	Human-readable, default
`json`	Parseable by `jq`
`stream-json`	Real-time streaming

```
# JSON output for automatic parsing
git status --short | claude -p "Categorize the
changes" \
  --output-format json | jq '.categories'

# Stream JSON for long operations
cat report.txt | claude -p "Summarize" --output-
format stream-json
```

`--no-stream` and flow control

By default, Claude Code streams the response character by character. `--no-stream` waits for the complete response before displaying anything, which simplifies pipes with tools that expect complete input.

```
claude -p "Generate a report" --no-stream >
report.md
```

CI/CD usage

In an isolated container, `--dangerously-skip-permissions` suppresses all confirmation requests. Reserve exclusively for sandboxed environments where Claude cannot access sensitive resources.

```
# GitHub Actions (isolated container)
claude -p "Run the tests and fix failures" \
  --dangerously-skip-permissions \
  --output-format json
```

Common patterns

```
# Automated log analysis
tail -n 200 app.log | claude -p "Critical alerts
only" \
  --output-format json

# PR review (package.json integration)
git diff main...HEAD | claude -p "Security
review, JSON format" \
  --output-format json > review.json
```

Interactive vs Non-Interactive

Aspect	Interactive	Non-Interactive (`-p`)
Output	Stream UI	Raw stdout
Permissions	Prompts	Auto or skip
Usage	Development	CI, scripts, pipes
Session	Persistent	Single, stateless

Best practice: limit pipe size to avoid exceeding the context window. Filter with `head`, `grep` or `--name-only` before sending to Claude.