# Git Commit Messages Cheat Sheet

## Conventional Commits Template

```
<type>(scope): <description>

[optional body]

[optional footer(s)]
```

**Scope** specifies what part of the codebase is affected:

- Feature/module: `auth`, `payments`, `dashboard`, `profile`
- File/folder: `api`, `database`, `config`, `utils`
- Component: `navbar`, `sidebar`, `footer`, `button`
- Layer: `frontend`, `backend`, `ui`, `middleware`

### Examples:

```
feat(auth): Add JWT token refresh functionality
fix(database): Resolve connection timeout issue
docs(readme): Update installation instructions
```

### Commit Types

| Type | Purpose |
|---|---|
| **feat** | Adds a new feature |
| **fix** | Fixes a bug |
| **refactor** | Restructures code without fixing bugs or adding features |
| **chore** | Miscellaneous changes (dependencies, .gitignore, etc.) |
| **perf** | Performance improvements |
| **ci** | Continuous integration changes |
| **ops** | Operational components (infrastructure, deployment, backup) |
| **build** | Build system changes (build tools, CI pipeline, dependencies) |
| **docs** | Documentation updates (README, etc.) |

| Type | Purpose |
| --- | --- |
| **style** | Code formatting (whitespace, semicolons, etc.) |
| **revert** | Reverts a previous commit |
| **test** | Adds or corrects tests |

---

## The Golden Rules

1. **Limit header to 50 characters**
2. **Capitalize the header**
3. **Do not end header with a period**
4. **Separate header from body with a blank line**
    - Use multiple `-m` flags: `git commit -m "header" -m "body"`
    - For multiline body (no blank lines): Press Enter inside quotes:

```bash
git commit -m "header" -m "Line one.
Line two.
Line three."
```

5. **Wrap body at 72 characters**
6. **Use body to explain what and why** (not how)
7. **Use imperative mood** (command-like)
    - ✅ "Add unit tests for user authentication"
    - ❌ "Added unit tests" or "Adding unit tests"

---

## Examples

### Good ✅

```
feat: Add user authentication module

Implement JWT-based authentication with refresh tokens.
This addresses security concerns raised in issue #123.
```

**Bad** ❌

```
Tweaked a few things
```

```
fixed stuff
```

```
update
```

---

## Quick Tips

- Think: "This commit will **[your message here]**"
- Ask: "Will someone understand this in 6 months?"
- Be specific but concise
- Focus on the "why" in the body, not the "how"

---

## View Your Commit History

```bash
git log
```

Review your past commits to see if they make sense!

---

## Terminal Tip: Multiline Input

When typing a multiline body, press Enter inside the quotes. The terminal continues on the next line:

```bash
$ git commit -m "header" -m "Line one.
> Line two.
> Line three."
```

The `>` prompt means you're still inside the quoted string. Close with `"` and press Enter to execute.