

Core commands in Claude Code

These are the essential slash commands you will use in every Claude Code session.

These are the essential slash commands you will use in every Claude Code session.

The commands you will use daily

Claude Code has many commands. Most sessions use a few familiar commands. Let's explore those commands first.

File context commands

/add

Add the file to Claude's context. This is probably the command you'll use most often.

Basic usage instructions:

```
/add src/auth.js
```

Add more files:

```
/add src/auth.js src/user.js src/session.js
```

Add more glob patterns:

```
/add src/components/**/*.tsx # T?t c?  
các file TSX trong components /add tests/*.test.js # T?t c? các file ki?  
m th? /add **/*.md # T?t c? các file markdown
```

When to use:

1. Before asking questions about specific code.
2. Before requesting changes to a file
3. When Claude needed to understand how the files worked together.

Helpful tip : Add the file BEFORE asking the question. Claude will give a better answer once he's seen the actual code.

/ls

List the files in context.

```
/ls
```

This displays what Claude currently knows. Use this command to verify that your /add commands worked or to see what has accumulated in the context.

Conversation management

/clear

Completely delete your conversation history.

```
/clear
```

When to use:

1. Start a completely new task.
2. The context becomes muddled with irrelevant history.
3. Claude was confused or gave inconsistent feedback.
4. You want to start over without restarting Claude Code.

Warning : This action is irreversible. Everything from the current session will be deleted.

/compact

Summarize the context to reduce the file size while retaining key information.

```
/compact
```

When to use:

1. The context is getting bigger, but you want to keep working.
2. Claude refers to the possibility of reaching the contextual limit.
3. Responses are slowing down.

The difference compared to /clear: Compact retains the essence of what you discussed. Clear deletes everything.

/undo

Restore Claude's final actions.

```
/undo
```

When to use:

1. Claude has made changes that you didn't want.
2. You want to try a different approach.
3. Something is wrong.

Faster than undoing changes manually.

Help and explore

/help

Display available commands.

```
/help
```

The commands change with updates. /help always displays what is currently available.

/status

Display the current session status.

```
/status
```

Displays contextual usage levels, active files, and other session information.

Actual samples

Start a new mission.

```
/clear /add src/feature-im-working-on/* > Hãy thêm cache vào l?p tìm n?p d?  
li?u
```

Delete all related files or specific tasks.

Refresh the context mid-way.

```
/compact /add src/new-file-i-need.js > Ti?p t?c v?i vi?c tri?  
n khai cache. Bây gi? hãy tích h?p v?i file m?i này.
```

Compact helps preserve history. Add new context. Continue from where you left off.

Explore an unfamiliar codebase.

```
/add README.md package.json > D? án này làm gì và công ngh? s? d?  
ng là gì? /add src/index.js src/app.js > Hãy cho tôi bi?t các ?i?m truy c?  
p chính
```

Start with an overview, then narrow it down.

Think before you commit.

```
/add $(git diff --name-only) > Xem l?i nh?ng thay ??i này. Có l?i ho?c v?n  
?? gì không?
```

Only add the changed files. Consider focusing on them.

Common mistakes

Adding too much:

```
/add src/**/*
```

Don't include your entire source code. The context is limited. Be selective.

Not enough added:

```
> T?i sao hàm này không ho?t ??ng?
```

Claude can't see the function. Please add the file first.

Forget what's in context:

```
# Sau nhi?u l?nh /add. > T?i sao b?n l?i tham chi?u ??n file c? ?ó?
```

Use `/ls` to see what's actually in the context. Use `/clear` if it's too cluttered.

Use `/clear` when `/compact` is functional:

```
/clear # B?y gi? t?i ph?i th?m l?i m?i th? và gi?i th?ch l?i nhi?m v?
```

If you want to preserve the session context but reduce its size, use `/compact` instead of `/clear`.

Quick reference to commands

Command	Its function	When should it be used?
<code>/add</code>	Add files to the context.	Before performing any specific tasks related to files
<code>/ls</code>	List the context files.	To check what Claude saw.
<code>/clear</code>	Delete the conversation	A new beginning with a new mission.
<code>/compact</code>	Context summary	Reduce in size, retain the essence.
<code>/undo</code>	Undo the last action	Claude made an unwanted change.
<code>/help</code>	Display commands	Discover or verify commands
<code>/status</code>	Display session status	Check the usage in context.

Exercise

Open Claude Code in a project and practice:

1. Add some source files using the `/add` command.
2. Use the `/ls` command to verify if they are in context.
3. Ask questions about the code.
4. Use the `/compact` command to see how it's summarized.

5. Use the `/clear` command and notice the difference.

Familiarize yourself with these commands before proceeding. They are fundamental to everything else.

Key points to remember

1. `/add` is your most important command - provide Claude with the appropriate files.
2. `/clear` to start over, `/compact` to reduce the size while retaining context.
3. Check what's in the context using the `/ls` command when confused.
4. `/undo` is faster than manually undoing.
5. Less is more: Being selective about context is better than dumping everything in.

1. Question 1:

What should you do when Claude's feedback starts to become confusing or inconsistent?

1. A. Restart your computer.
2. B. Remove or simplify the context to reduce clutter and confusion.
3. C. Switch to a different AI
4. D. Add more files to provide more information.

EXPLAIN:

A cluttered context often leads to confusing responses. Clearing or simplifying the context will provide Claude with a cleaner platform to work on.

2. Question 2:

When should you use `/add` with a glob template like `'src/**/*.*ts'`?

1. A. Never - always add each file one by one.
2. B. When you want Claude to understand an entire module or feature area.
3. C. Only if you have fewer than 10 files
4. D. For configuration files only

EXPLAIN:

The glob template allows you to add multiple related files at once, which is useful when Claude needs to understand how the files work together.

3. Question 3:

What is the difference between `/clear` and `/compact`?

1. A. They are the same command.
2. B. `/clear` completely deletes the history, `/compact` summarizes it to save space.

3. C. /clear is faster, /compact is more thorough.
4. D. /compact clears the history, /clear summarizes it.

EXPLAIN:

/clear completely deletes the conversation history. /compact summarizes the context to retain important information while minimizing token usage.

Submit your work

Training results

You have completed **0** questions.

-- / --

[Review the lesson](#)

You finished reading the article "**Core commands in Claude Code**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.