

Claude Code चीट शीट

Claude Code में महारत हासिल करने के लिए आवश्यक संदर्भ

</> CLAUDECODE-LAB

1. बुनियादी कमांड

<code>claude</code>	इंटरैक्टिव मोड शुरू करें
<code>claude -p "..."</code>	एक बार में निष्पादन (गैर-इंटरैक्टिव)
<code>/help</code>	सहायता दिखाएं
<code>/clear</code>	संदर्भ साफ़ करें
<code>/compact</code>	संदर्भ संपीड़ित करें और जारी रखें
<code>/cost</code>	सत्र लागत जांचें
<code>/exit</code>	Claude Code बंद करें

2. कीबोर्ड शॉर्टकट

<code>Ctrl + C</code>	वर्तमान ऑपरेशन रद्द करें
<code>Esc</code>	इनपुट रद्द करें
<code>Esc</code> x2	पिछले संदेश पर वापस जाएं
<code>Shift + Tab</code>	अगला सुझाव/मोड स्विच
<code>↑</code> / <code>↓</code>	इनपुट इतिहास नेविगेट करें

3. CLAUDE.md लिखने के टिप्स

- ▶ **प्रोजेक्ट अवलोकन** — आप क्या बना रहे हैं 1-2 पंक्तियों में
- ▶ **टेक स्टैक** — भाषा, फ्रेमवर्क, DB, प्रमुख लाइब्रेरी
- ▶ **कोडिंग मानक** — नामकरण, फॉर्मैटर, lint कॉन्फिग
- ▶ **सामान्य कमांड** — build / test / dev / deploy
- ▶ **प्रतिबंधित क्षेत्र** — फ़ाइलें/डायरेक्टरी जो संशोधित नहीं होनी चाहिए

4. प्रभावी प्रॉम्प्ट लिखना

- ▶ **सटीक फ़ाइल नाम दें** — "@src/api/user.ts ठीक करें"
- ▶ **अपेक्षित आउटपुट फॉर्मेट बताएं** — JSON / Markdown / diff आदि
- ▶ **पहले/बाद के उदाहरण दिखाएं** — I/O सैंपल सटीकता बढ़ाते हैं
- ▶ **चरणों में विभाजित करें** — डिज़ाइन → लागू करें → टेस्ट → रीफैक्टर
- ▶ **सत्यापन बताएं** — "npm test पास होना चाहिए"

5. उपयोगी पाइप इंटीग्रेशन

```
git diff | claude -p "इस diff की समीक्षा करें"
तुरंत कोड समीक्षा
```

```
cat error.log | claude -p "कारण क्या है?"
लॉग से मूल कारण विश्लेषण
```

```
cat package.json | claude -p "deps साफ़ करें"
package.json निर्भरता ऑडिट
```

```
git log --oneline -20 | claude -p "रिलीज़ नोट्स लिखें"
कमिट्स से CHANGELOG बनाएं
```

प्रो टिप्स

- ▶ **@फ़ाइलनाम** — फ़ाइलें सीधे संदर्भ में जोड़ें
- ▶ **/compact नियमित चलाएं** — लंबे सत्रों में सटीकता बनाए रखें
- ▶ **योजना और निष्पादन अलग करें** — पहले Plan, फिर Act