

# Aide-Memoire Claude Code

Votre reference essentielle pour maitriser Claude Code

</> CLAUDECODE-LAB

## 1. Commandes de Base

<code>claude</code>	Demarrer le mode interactif
<code>claude -p "..."</code>	Execution unique (non-interactif)
<code>/help</code>	Afficher l'aide
<code>/clear</code>	Effacer le contexte
<code>/compact</code>	Compresser le contexte et continuer
<code>/cost</code>	Verifier le cout de la session
<code>/exit</code>	Quitter Claude Code

## 2. Raccourcis Clavier

<code>Ctrl + C</code>	Annuler l'operation en cours
<code>Esc</code>	Annuler la saisie
<code>Esc</code> x2	Revenir au message precedent
<code>Shift + Tab</code>	Suggestion suivante/changer de mode
<code>↑ / ↓</code>	Parcourir l'historique des saisies

## 3. Rediger CLAUDE.md

- ▶ **Aperçu du projet** — Ce que vous construisez en 1-2 lignes
- ▶ **Stack technique** — Langage, framework, BD, librairies clés
- ▶ **Standards de code** — Nommage, formateur, config lint
- ▶ **Commandes courantes** — build / test / dev / deploy
- ▶ **Zones interdites** — Fichiers/dossiers a ne pas modifier

## 4. Prompts Efficaces

- ▶ **Specifiez les noms de fichiers** — "Corrige @src/api/user.ts"
- ▶ **Preciser le format de sortie** — JSON / Markdown / diff etc.
- ▶ **Montrer des exemples Avant/Après** — Les echantillons E/S ameliorent la precision
- ▶ **Decomposer en etapes** — Concevoir → Implementer → Tester → Refactoriser
- ▶ **Decrire la verification** — "npm test doit passer"

## 5. Integrations Pipe Utiles

```
git diff | claude -p "Revise ce diff"
Revue de code instantanee
```

```
cat error.log | claude -p "Quelle est la cause ?"
Analyse des causes depuis les logs
```

```
cat package.json | claude -p "Nettoie les deps"
Audit des dependances package.json
```

```
git log --oneline -20 | claude -p "Ecris les notes de version"
Generer le CHANGELOG depuis les commits
```

## Astuces Pro

- ▶ **@nom\_fichier** — Ajouter des fichiers directement au contexte
- ▶ **Executer /compact regulierement** — Maintenir la precision dans les longues sessions
- ▶ **Separer planification et execution** — D'abord planifier, puis agir