Jumeau numérique · Sous-section

Utiliser l'IA dans un terminal

Pour qui veut un workflow terminal en plus d'OpenCode Desktop : installer Claude Code ou OpenCode en CLI, configurer un terminal moderne (Ghostty, WezTerm, iTerm2), utiliser Tmux pour le multi-pane, créer un alias `cc`.

Source : https://prisme.one/plateforme/ia-juillet-2026/semaine-1-utiliser-ia-terminal

Objectif

Comprendre quand le terminal est pertinent (dev avancé, workflow flexible) et savoir installer un workflow IA terminal complet.

Output attendu

Terminal moderne installé, Claude Code ou OpenCode en CLI fonctionnel, alias `cc` qui fait cd + lancement IA, optionnellement Tmux pour le multi-pane.

Utiliser l'IA dans un terminal

Setup terminal et sessions IA en parallèle

LOOM

Setup terminal et sessions IA en parallèle

https://www.loom.com/share/9a6d584703f54cb7ac7c71c729d4213a

Démo concrète : comment lancer plusieurs sessions IA en parallèle dans un terminal moderne.

OpenCode Desktop suffit largement pour suivre ce bootcamp. Tu n'as pas besoin du terminal pour construire ton second cerveau, créer tes skills, ou faire tourner tes rituels.

Mais le terminal devient pertinent dans deux cas. Cette leçon est pour toi si tu te reconnais dans l'un des deux.

Pourquoi utiliser l'IA dans un terminal ?

Cas 1
Développement avancé
Tu veux que l'IA t'aide à coder, gérer git, builder un projet, lancer des scripts en parallèle de ton coffre. L'IA dans le terminal te donne un accès système complet : elle peut exécuter des commandes, manipuler git, installer des dépendances.
Cas 2
Workflow plus flexible
Tu veux plusieurs sessions IA en parallèle dans des fenêtres différentes, des alias custom pour démarrer plus vite, des sessions persistantes que tu retrouves même après avoir fermé ton terminal. C'est la puissance d'un workflow terminal mature.

Si tu ne te reconnais dans aucun des deux, reste sur OpenCode Desktop. Tu pourras toujours revenir à cette leçon plus tard.

Les commandes de base du terminal

Le terminal (aussi appelé « ligne de commande » ou « console ») est une interface texte pour communiquer avec ton ordinateur. Au lieu de cliquer, tu tapes.

Tu n'as besoin que de **4 commandes** pour le bootcamp.

Mac / Linux :

```
# Voir où tu es (Print Working Directory)
pwd

# Lister les fichiers du dossier actuel
ls

# Changer de dossier
cd nom-du-dossier

# Revenir au dossier parent
cd ..
```

Windows (PowerShell) :

```
# Voir où tu es
pwd

# Lister les fichiers
dir
# (ls fonctionne aussi sur PowerShell moderne)

# Changer de dossier
cd nom-du-dossier

# Revenir au dossier parent
cd ..
```

Exercice pratique (3 minutes)

1. Ouvre ton terminal
2. Tape `pwd` puis Entrée : tu vois ton dossier actuel
3. Tape `ls` (ou `dir` sur Windows) : tu vois les fichiers
4. Tape `cd Desktop` puis `ls` : tu entres dans le Bureau et tu vois son contenu
5. Tape `cd ..` : tu reviens en arrière

> **Astuce Mac :** Tu peux glisser-déposer un dossier dans le terminal après avoir tapé `cd` (avec un espace) pour naviguer directement dedans.

> **Astuce Windows :** Tu peux faire `Shift + Clic droit` sur un dossier dans l'Explorateur et choisir « Ouvrir PowerShell ici ».

C'est tout. Tu n'auras pas besoin de plus pour ce bootcamp.

Quel terminal utiliser ?

Les terminaux par défaut (Terminal.app sur Mac, PowerShell sur Windows) fonctionnent. Mais un terminal moderne est plus rapide, plus lisible, et surtout te permet de gérer **plusieurs sessions IA en parallèle**, ce qui change vraiment le quotidien.

Notre recommandation principale

Choisis selon ton OS :

Sur macOS
cmux

Un terminal moderne pensé pour piloter **plusieurs sessions IA en parallèle** dans la même fenêtre. Tu peux avoir Claude Code sur ton coffre, OpenCode sur un repo client, et une session libre, et basculer entre elles d'un raccourci. C'est la couche qui rend le terminal vraiment puissant.

cmux.com
Sur Windows
WezTerm

cmux n'est pas dispo sur Windows. **WezTerm** est la meilleure alternative : cross-platform, très configurable, GPU-accéléré, splits natifs pour avoir plusieurs sessions côte à côte. Excellent rendu, communauté active.

wezterm.org

Les alternatives solides

Si tu veux explorer d'autres options :

Ghostty
Gratuit, ultra-rapide, GPU-accéléré, minimaliste. Mac et Linux uniquement.
iTerm2
Le classique sur Mac. Splits, search, profils, hotkeys. Très solide, communauté énorme.
Alacritty
Ultra-léger, GPU-accéléré, minimaliste à l'extrême. Mac, Windows, Linux.
Kitty
Puissant, très configurable. Mac et Linux.

Installer Claude Code ou OpenCode en CLI

Choisis ton interface. Les deux sont compatibles avec ton coffre, tu peux changer d'avis à tout moment sans rien perdre.

Claude Code (recommandé si tu as un abonnement Claude Pro/Max)

Suis le guide d'installation officiel : docs.anthropic.com/en/docs/claude-code.

Une fois installé, tape `claude` dans ton terminal, connecte-toi avec ton compte Anthropic (le navigateur s'ouvre), c'est fait.

OpenCode (multi-modèle)

Suis le guide d'installation officiel : opencode.ai/docs.

Une fois installé, tape `opencode` dans ton terminal, puis `/model` pour configurer ton provider (OpenRouter recommandé, cf. la leçon principale d'installation).

Toujours ouvrir l'IA depuis ton workspace

L'IA n'a accès qu'au dossier dans lequel tu la lances. Si tu lances `claude` ou `opencode` depuis ton Bureau, elle ne voit pas ton coffre.

Le geste à faire à chaque session :

```
cd ~/Documents/WORKSPACE   # adapte au chemin de ton workspace
claude                      # ou opencode
```

Tu te retrouves dans le bon dossier, avec l'IA chargée de tout le contexte (AGENTS.md, skills, etc.).

L'alias `cc` : démarrer en une frappe

Plutôt que de taper `cd` puis `claude` (ou `opencode`) à chaque session, crée un alias `cc` qui fait les deux d'un coup.

Le plus simple : demande à ton IA de le créer pour toi.

Demande à ton IA

Lance ton IA une première fois, puis donne-lui ce prompt :

```
J'aimerais que tu crées un alias "cc" dans mon terminal.

Quand je tape "cc", il devrait :
```

```
1. Aller dans le dossier de mon workspace (adapte le path)
2. Lancer claude (ou opencode, à toi de choisir)

Modifie le bon fichier de config (.zshrc, .bashrc ou équivalent
selon mon shell) pour que l'alias soit permanent.
```

L'IA va détecter ton shell, modifier le bon fichier de configuration, et créer l'alias.

Tester

Ferme et rouvre ton terminal, puis tape :

```
cc
```

Tu devrais te retrouver dans ton workspace avec ton IA lancée. Une seule frappe au lieu de deux commandes.

- [ ] Je sais quand le terminal devient pertinent (dev avancé ou workflow flexible)
- [ ] Je connais les 4 commandes de base (pwd, ls, cd, cd ..)
- [ ] J'ai choisi mon terminal (cmux sur macOS, WezTerm sur Windows, ou une alternative)
- [ ] J'ai installé Claude Code ou OpenCode en CLI
- [ ] Je sais toujours lancer l'IA depuis mon workspace
- [ ] J'ai créé l'alias `cc` qui fait cd + lancement de l'IA

Vidéos

LOOM

Setup terminal et sessions IA en parallèle

https://www.loom.com/share/9a6d584703f54cb7ac7c71c729d4213a
