Damien
@damien.mapt

Clickflow

5 plugins pour que Claude Code code comme un dev senior

Agent Skills, Context7, Graphify, 21st MCP et Strix : ce que fait chacun, comment l'installer, et la phrase à envoyer à Claude pour t'en servir.

Avant de commencer : les cinq sont open source (code gratuit). Il te faut **Claude Code** (application Claude pour ordinateur, onglet Code, ou le terminal). Le plus simple : **colle chaque bloc « à copier-coller » dans Claude Code**, il fait l'installation pour toi. Deux outils peuvent coûter un peu à l'usage (21st MCP pour la génération par IA, Strix qui consomme une clé d'IA) : c'est indiqué plus bas.

1 Agent Skills : la méthode d'un dev senior

25 skills écrits par Addy Osmani (ancien de Google, aujourd'hui chez Anthropic) : idée, spécification, plan, code, tests, revue, mise en ligne. Ils se déclenchent tout seuls selon ce que tu fais. Licence MIT. github.com/addyosmani/agent-skills

À COPIER-COLLER

Installe le plugin Agent Skills : /plugin marketplace add addyosmani/agent-skills puis /plugin install agent-skills@addy-agent-skills

Pour t'en servir : `/spec` pour décrire ce que tu veux, `/plan` pour découper, `/build` pour coder étape par étape, `/test`, `/ship` avant la mise en ligne.

2 Context7 : la documentation à jour

Claude a appris sur des documentations qui datent. Context7 lui donne la version à jour de tes librairies et outils, au moment où il code. Licence MIT. github.com/upstash/context7

À COPIER-COLLER

Installe Context7 pour Claude Code avec la commande officielle : npx ctx7 setup

Pour t'en servir : ajoute **« utilise Context7 »** à ta demande, par exemple « Ajoute la connexion Google à mon appli, utilise Context7 ».

3 Graphify : la carte de ton projet

Il transforme ton projet (code, notes, PDF) en une carte qui relie fichiers et fonctions. Claude la consulte au lieu de tout relire à chaque question. Licence Apache 2.0. github.com/Graphify-Labs/graphify

> À COPIER-COLLER
>
> Installe Graphify : pipx install graphifyy puis graphify install (si pipx n'est pas là, installe-le d'abord). Ensuite lance /graphify . sur mon projet.

Pour t'en servir : pose tes questions sur le projet (« où est gérée la connexion ? »), Claude passe par la carte. Après de gros changements, relance `/graphify .`

4 21st MCP : 10 000 composants d'interface

Une bibliothèque de plus de 10 000 composants React / Tailwind (boutons, formulaires, tableaux de bord…) que Claude ajoute à ton projet. Licence ISC. Clé gratuite sur 21st.dev/mcp ; la génération de nouveaux composants par l'IA consomme des crédits, la recherche dans la bibliothèque non. github.com/21st-dev/magic-mcp

> À COPIER-COLLER
>
> Installe 21st MCP pour Claude Code : npx @21st-dev/cli@latest init --client claude. Je te donne ma clé quand tu me la demandes.

Pour t'en servir : « Cherche dans 21st un tableau de prix à trois colonnes et ajoute-le à ma page d'accueil ».

5 Strix : le hacker qui teste ta propre appli

Des agents IA attaquent ton appli comme un vrai pirate, prouvent chaque faille et proposent la correction. Licence Apache 2.0. github.com/usestrix/strix

> À COPIER-COLLER
>
> Installe Strix (curl -sSL https://strix.ai/install | bash). Vérifie que Docker tourne, aide-moi à régler ma clé d'IA, puis lance strix --target ./ sur mon projet et explique-moi les failles trouvées.

Important : il faut Docker et une clé d'IA (OpenAI, Anthropic, Google…) : chaque analyse consomme un peu de crédits chez ce fournisseur. Et **uniquement sur tes propres applis** : tester celle de quelqu'un d'autre sans son accord est illégal.

Dans quel ordre ?

1. **Agent Skills + Context7** dès le premier jour : la méthode et la bonne documentation.
2. **Graphify** quand ton projet dépasse quelques fichiers.
3. **21st MCP** quand tu travailles l'interface.
4. **Strix** avant de mettre ton appli en ligne, puis après chaque grosse mise à jour.

Tu veux d'autres astuces IA comme celle-ci ? Suis **@damien.mapt** sur Instagram : une nouvelle chaque semaine.