

Guide Complet

Comment créer un script qui fonctionne avec ton skill Claude Code

*+ Template SKILL.md prêt à l'emploi
+ 3 exemples concrets complets*

@jasmina.alan.duality

Avant de commencer

Ce guide t'accompagne pas à pas pour créer un fichier SKILL.md fonctionnel dans Claude Code. À la fin, tu auras un skill opérationnel que Claude charge automatiquement dès qu'il détecte la bonne situation.



Ce que tu vas apprendre

Structurer un SKILL.md · Écrire des instructions précises · Définir des triggers · Ajouter des exemples · Connecter le skill à ton projet · Tester et itérer

Les Skills Claude Code — C'est quoi ?

Un skill est un fichier Markdown que Claude charge dynamiquement pour devenir expert dans une tâche précise. Plutôt que de réexpliquer ton contexte à chaque conversation, le skill s'en charge automatiquement.

Avantages concrets :

- Zéro répétition : Claude connaît déjà tes règles
- Résultats cohérents d'une session à l'autre
- Facile à partager avec une équipe
- Open source : tu peux t'inspirer des skills existants



Où vivent les skills ?

Dans ton projet : `.claude/skills/SKILL.md` — Claude les détecte automatiquement quand tu travailles dans ce dossier.

Le Guide Pas-à-Pas

1 Créer la structure de dossiers

Dans ton projet, crée les dossiers suivants :

```
mon-projet/
├── .claude/
│   ├── CLAUDE.md           ← point d'entrée principal
│   └── skills/
│       └── SKILL.md        ← ton skill
└── src/
```

Commande rapide : `mkdir -p .claude/skills && touch .claude/skills/SKILL.md .claude/CLAUDE.md`

2 Écrire l'en-tête YAML du skill

Tout SKILL.md commence par un bloc front matter YAML. Il indique à Claude le nom du skill, sa description et sa licence. Sans ce bloc, Claude ne reconnaît pas le fichier comme un skill valide.

```
---
name: mon-skill-excel
description: >
  Crée, lit et nettoie des fichiers Excel (.xlsx).
  Utilise openpyxl. Sauvegarde dans /outputs/.
license: MIT
---
```

✓ Règle d'or

La description doit être précise et concise. C'est elle que Claude lit en premier pour décider si le skill est pertinent.

3 Écrire les instructions principales

Après l'en-tête, écris en Markdown les instructions que Claude devra suivre. Pense à cela comme un brief ultra-précis adressé à un collaborateur expert.

```
## Comportement général

Quand tu travailles sur des fichiers Excel :
- Utilise TOUJOURS openpyxl (jamais xlrd)
- Ajoute un en-tête en ligne 1 avec fond #D4EAC8
- Formate les colonnes automatiquement (auto_fit)
- Sauvegarde le résultat dans /mnt/user-data/outputs/
- Valide le fichier après création

## Règles de nommage
- Fichiers : snake_case (ex: rapport_ventes_2024.xlsx)
- Feuilles : Titre En Majuscule (ex: Données Clients)
```

4 Définir les triggers

Les triggers indiquent à Claude dans quelles situations activer ce skill. Il les détecte automatiquement dans tes messages. Plus c'est précis, mieux c'est.

```
## Quand utiliser ce skill

Charge et applique ce skill quand l'utilisateur mentionne :
- Des mots-clés : Excel, XLSX, XLS, tableur, spreadsheet
- Des actions : créer un tableau, nettoyer des données, lire un fichier Excel, exporter en Excel
- Des fichiers uploadés avec extension .xlsx ou .xls

NE PAS utiliser ce skill pour : Google Sheets, CSV simple, PDF, ou tout format non-Excel.
```

5 Ajouter des exemples concrets

Les exemples sont la partie la plus puissante d'un skill. Ils montrent à Claude ton style, tes standards et le niveau de détail attendu. Ajoute au minimum 2-3 exemples.

```
## Exemples

### Exemple 1 - Créer un tableau de suivi
User: Crée un tableau de suivi de prospection
Claude: [génère prospect_tracker.xlsx avec colonnes :
  Nom, Prénom, Email, Entreprise, Statut,
  Date de contact, Notes - en-tête vert, auto-fit]

### Exemple 2 - Nettoyer des données
User: Nettoie ce fichier Excel (supprime les doublons)
Claude: [charge le fichier, détecte les colonnes clés,
  supprime les doublons, sauvegarde cleaned_*.xlsx]
```

6 Connecter le skill dans CLAUDE.md

Dans ton fichier CLAUDE.md à la racine du dossier .claude/, dis à Claude où trouver tes skills et quand les charger. Ce fichier est chargé à chaque session Claude Code.

```
# Mon Projet - CLAUDE.md

## Skills disponibles

Avant toute tâche, lis les skills pertinents dans
.claude/skills/ selon la nature de la demande :

| Fichier | Quand l'utiliser |
|-----|-----|
| skills/SKILL.md | Tâches Excel / XLSX |
| skills/PDF_SKILL.md | Tâches PDF |

## Règles globales du projet
- Langue par défaut : français
```

- Outputs : toujours dans /mnt/user-data/outputs/

7 Tester et affiner ton skill

Lance Claude Code dans ton projet et envoie une demande qui doit déclencher ton skill. Observe attentivement :

- Claude mentionne-t-il avoir lu le SKILL.md ?
- Respecte-t-il toutes tes règles ?
- Le fichier output est-il au bon endroit ?
- Le format correspond-il à tes exemples ?

Itère jusqu'à la perfection

Un bon skill s'affine en 3-5 itérations. À chaque test raté, identifie quelle instruction manquait et ajoute-la. C'est normal.

Les 3 Erreurs à Éviter

Erreur 1 — Instructions trop vagues

✗ **Mauvais** : "Crée des fichiers Excel propres"

✓ **Bon** : "Utilise openpyxl, en-tête ligne 1 fond #D4EAC8, colonnes auto-fit, sauvegarde /outputs/"

Erreur 2 — Aucun trigger défini

Sans section "Quand utiliser ce skill", Claude ne sait pas l'activer automatiquement. Il te demandera des précisions au lieu d'agir directement.

Erreur 3 — Pas d'exemples

Sans exemples, Claude improvise selon son interprétation. Les exemples ancrent ses comportements dans ta réalité métier. Ce sont eux qui font la différence entre un skill correct et un skill excellent.

Template SKILL.md — Prêt à Copier-Coller

Remplace chaque [PLACEHOLDER] par tes propres informations.

```
---
name: [nom-de-ton-skill]
description: >
  [Description courte et précise de ce que fait le skill.
  Mentionner les outils utilisés et le type de fichiers.]
license: MIT
---

## Quand utiliser ce skill

Charge et applique ce skill quand l'utilisateur mentionne :
- [Mot-clé 1], [Mot-clé 2], [Mot-clé 3]
- [Action déclenchante 1]
- [Action déclenchante 2]

NE PAS utiliser pour : [cas exclus]

## Comportement général

Quand tu exécutes une tâche liée à ce skill :
- [Règle 1 - outil à utiliser]
- [Règle 2 - format de sortie]
- [Règle 3 - emplacement de sauvegarde]
- [Règle 4 - validation obligatoire]

## Règles spécifiques

- [Règle métier 1]
- [Règle métier 2]
- [Règle de nommage]

## Exemples

### Exemple 1 - [Cas d'usage principal]
User: [Message type de l'utilisateur]
Claude: [Description du comportement attendu et du résultat]

### Exemple 2 - [Cas d'usage secondaire]
User: [Message type de l'utilisateur]
Claude: [Description du comportement attendu et du résultat]

### Exemple 3 - [Cas limite / edge case]
User: [Message type de l'utilisateur]
Claude: [Comment gérer ce cas particulier]
```

3 Exemples Complets Prêts à l'Emploi

Exemple 1 — Skill DOCX (Word)

Idéal pour les freelances, rédacteurs, consultants qui génèrent des documents Word régulièrement.

```
---
name: docx-generator
description: >
  Crée et édite des fichiers Word (.docx) professionnels.
  Mise en page, titres hiérarchiques, tableaux, styles.
  Utilise la bibliothèque docx-js (npm).
license: MIT
---

## Quand utiliser ce skill

Active ce skill pour : Word, document Word, .docx, rapport,
brief, devis, cahier des charges, contrat, facture Word.

## Comportement général

- Utilise TOUJOURS docx-js (npm install -g docx)
- Police par défaut : Arial 12pt
- Marges : 2,54 cm (1 pouce) sur tous les côtés
- En-tête : nom du document + date automatique
- Pied de page : numérotation des pages
- Sauvegarde dans /mnt/user-data/outputs/

## Hiérarchie des styles

- H1 : 18pt, gras, noir
- H2 : 14pt, gras, vert #2D5A1B
- Corps : 12pt, interligne 1,5
- Note : 10pt, italique, gris

## Exemples

### Exemple 1 - Devis client
User: Crée un devis pour une mission de 3 jours à 600€/jour
Claude: [génère devis_client.docx avec section Prestation,
tableau Quantité/Prix/Total, conditions de paiement,
signature, mise en page professionnelle]

### Exemple 2 - Rapport d'analyse
User: Génère un rapport d'analyse SEO
Claude: [génère rapport_seo.docx avec table des matières,
sections Résumé exécutif / Analyse / Recommandations,
tableaux de données, numérotation automatique]
```

Exemple 2 — Skill Contenu Social

Parfait pour les créateurs de contenu qui produisent régulièrement des posts, newsletters et scripts.

```
---
name: social-content-creator
```

```

description: >
  Crée du contenu pour LinkedIn, Instagram et newsletters.
  Respecte une ligne éditoriale définie. Ton : expert,
  accessible, sans jargon inutile. Langue : français.
license: MIT
---

## Quand utiliser ce skill

Active pour : post LinkedIn, carrousel Instagram, newsletter,
thread, caption, script vidéo, contenu réseaux sociaux.

## Ma ligne éditoriale

- Ton : direct, bienveillant, concret
- Audience : freelances, entrepreneurs, créateurs
- Jamais de : buzzwords vides, fausse urgence, clichés AI
- Toujours : un insight actionnable par contenu

## Formats par plateforme

LinkedIn : 150-300 mots, 1 idée principale, hook fort,
pas de bullet points en masse, 1 question en fin

Instagram carrousel : 8-12 slides, slide 1 = hook,
dernière slide = CTA clair

Newsletter : 400-600 mots, structure Contexte / Insight
/ Action / Ressource

## Exemples

### Exemple 1 – Post LinkedIn
User: Écris un post sur les erreurs des débutants en IA
Claude: [génère un post 220 mots avec hook question,
3 erreurs numérotées, conseil actionnable, CTA]

### Exemple 2 – Carrousel Instagram
User: Carrousel sur les 5 outils IA gratuits
Claude: [génère 10 slides : cover hook + 5 outils +
2 slides tips + 1 bonus + 1 CTA DM]

```

Exemple 3 — Skill Analyse de Données

Pour les analystes, data scientists et ops qui traitent des fichiers CSV/Excel régulièrement.

```

---
name: data-analysis
description: >
  Analyse, nettoie et visualise des données tabulaires.
  CSV, Excel, JSON. Utilise pandas, matplotlib, seaborn.
  Génère des rapports HTML et graphiques PNG.
license: MIT
---

## Quand utiliser ce skill

Active pour : analyser des données, nettoyer un CSV,
doublons, valeurs manquantes, graphique, visualisation,
statistiques, corrélation, rapport data.

```

```
## Stack technique

- pandas pour la manipulation
- matplotlib + seaborn pour les graphiques
- Toujours : pip install --break-system-packages
- Outputs : /mnt/user-data/outputs/analysis/

## Process standard

1. Charger et afficher les 5 premières lignes
2. Résumé statistique (describe())
3. Détecter les valeurs nulles
4. Appliquer le nettoyage demandé
5. Valider le résultat
6. Sauvegarder le fichier propre + rapport

## Exemples

### Exemple 1 – Nettoyage CSV client
User: Nettoie ce fichier clients (doublons sur email)
Claude: [charge le CSV, détecte les doublons sur la
         colonne email, garde la ligne la plus récente,
         génère cleaned_clients.csv + rapport de suppression]

### Exemple 2 – Visualisation ventes
User: Graphique des ventes par mois depuis ce fichier
Claude: [génère ventes_par_mois.png avec seaborn,
         courbe + barres, légende claire, couleurs cohérentes]
```

Checklist Avant de Lancer

Coche chaque point avant de tester ton skill dans Claude Code :

- Le dossier `.claude/skills/` existe dans mon projet

- Mon fichier `SKILL.md` commence par le bloc `--- YAML ---`

- La description est précise et mentionne les outils utilisés

- J'ai une section "Quand utiliser ce skill" avec des triggers

- J'ai au moins 3 règles concrètes dans les instructions

- J'ai au moins 2 exemples User/Claude

- Mon fichier `CLAUDE.md` référence le skill

- J'ai testé avec une vraie demande dans Claude Code

- Claude a bien lu et appliqué le skill lors du test

*Ce guide t'a été envoyé par @jasmina.alan.duality
Tous les skills sont open source et maintenus par Anthropic.
github.com/anthropics/claude-code-skills*