AI TOOLS WIKI · MÉMO

Impeccable *mémo*

Les commandes à connaître pour **Impeccable**, la skill de design de Paul Bakaus pour Claude Code, Codex, Cursor… Elle donne à l'agent un vocabulaire de design commun (`polish`, `audit`, `bolder`…) et un détecteur automatique de 61 défauts typiques des interfaces générées par IA.

Famille Design en code · **Éditeur** pbakaus · **Licence** Apache-2.0
Note du wiki 11/15 · **Commandes vérifiées le** 25 septembre 2026
Fiche, notes et avis aitoolswiki.viellard.fr/outils/design-en-code/impeccable

Installation

À la racine du projet (détecte Claude Code, Codex, Cursor… et installe le hook de design)	
`npx impeccable install`	
`npx impeccable update`	mettre à jour
Ou via la marketplace Claude Code	
`/plugin marketplace add pbakaus/impeccable`	puis installer depuis /plugin

Puis, **dans l'agent, une fois par projet** : `/impeccable init`. Il pose quelques questions et écrit `PRODUCT.md` (public, objectif, contraintes, ton), que toutes les autres commandes utilisent.
Sur Codex : ouvrir `/hooks` après l'installation pour approuver le hook, et appeler la skill avec `$impeccable`.

Les 24 commandes

Toutes passent par `/impeccable <commande> [cible]`, ex. `/impeccable audit le header`.

Démarrer et structurer

`init`	Recueille le contexte produit et écrit `PRODUCT.md`. **À lancer en premier.**
`shape`	Planifie l'UX/UI avant d'écrire du code.
`craft`	Enchaînement complet : cadrage puis construction avec itérations visuelles.
`document`	Génère un `DESIGN.md` à partir du code existant.
`extract`	Extrait composants et tokens réutilisables vers le design system.

Les 24 commandes (suite)

Diagnostiquer

`critique`	Revue UX : hiérarchie, clarté, impact émotionnel.
`audit`	Contrôles techniques : accessibilité, performance, responsive.

Corriger et ajuster

`polish`	Dernière passe avant mise en ligne, alignement sur le design system.
`bolder` / `quieter`	Rend un design plus affirmé / plus sobre.
`distill`	Retire le superflu.
`layout`	Corrige mise en page, espacements, rythme.
`typeset`	Corrige polices, hiérarchie et tailles de texte.
`colorize`	Introduit la couleur de façon stratégique.
`animate`	Ajoute des animations utiles.
`clarify`	Améliore les textes d'interface peu clairs.
`harden`	Gestion d'erreurs, i18n, débordements de texte, cas limites.
`onboard`	Premier lancement, états vides, parcours d'activation.
`adapt`	Adapte à d'autres appareils.
`optimize`	Améliore les performances.
`delight` / `overdrive`	Ajoute des touches de plaisir / des effets techniques spectaculaires.

Itérer dans le navigateur

`live`	Mode variantes : itérer sur des éléments directement dans le navigateur.
`generate`	Génère des variantes d'un élément nommé.

Raccourci : `/impeccable pin audit` crée une commande `/audit` autonome.

Détecteur en ligne de commande (sans IA ni clé API)

`npx impeccable detect src/`	Analyse un dossier (ou un fichier HTML).
`npx impeccable detect https://example.com`	Analyse une page en ligne (Chrome requis).
`npx impeccable detect --json .`	Sortie JSON pour la CI (code 2 = défauts trouvés).
`npx impeccable ignores add-value overused-font Inter --reason "Police de marque"`	Autorise une exception justifiée.

Workflow conseillé

1. `/impeccable init` (une seule fois)
2. `/impeccable shape` pour une nouvelle page, ou `/impeccable audit` + `/impeccable critique` sur l'existant
3. Commandes ciblées selon les problèmes : `layout`, `typeset`, `bolder`, `harden`…
4. `/impeccable polish` avant de livrer

Conseils d'utilisation

- **Toujours cibler** : `/impeccable polish le formulaire de paiement` donne de meilleurs résultats qu'un `polish` sur toute l'application.
- **Diagnostiquer avant de corriger** : `audit` et `critique` d'abord, les corrections ensuite.
- **Les défauts traqués** : police Inter partout, dégradés violet-bleu, cartes imbriquées, texte gris sur fond coloré, noir pur, animations « rebond ».
- **Ajouter la détection à la CI** avec `detect --json`, pour attraper ces défauts à chaque PR.
- **Ne pas committer les fichiers temporaires** de `.impeccable/` (captures, sessions) : ajouter le bloc `.gitignore` fourni dans le README. Garder en revanche `config.json`, `design.json` et les rapports de critique.
- **Complémentaire de GStack** : gstack utilise le détecteur d'Impeccable dans ses revues de design s'il est installé.

Sources (documentation officielle)

- README Impeccable
- Documentation impeccable.style
- Impeccable Skill – Skillselion