Eigenflow

GUIDE · SKILL CLAUDE

La skill Apple Design

Installée et testée sur nos pages : la commande, les prompts, les 17 principes, nos chiffres avant/après

PAR

Abyane Nabeebaccus

INSTAGRAM

@abyane_nb

ÉDITION

novembre 2026

Tu as commenté DESIGN. Voici le guide de la skill du reel : ce qu'elle fait, comment l'installer, les prompts exacts qu'on a utilisés, ses 17 principes avec ce qu'il faut vérifier chez toi, puis nos deux tests, chiffres compris.

On l'a testée deux fois. Le 1er novembre 2026, sur un guide de notre propre site. Le 2 novembre 2026, sur deux pages générées par Claude, en relecture puis en construction. Les chiffres du dépôt GitHub ont été relevés le 2 novembre 2026 à 7 h 55, heure de Paris.

Chaque chiffre vient de nos fichiers de test ou d'une source datée, listée à la fin. Ce qu'on n'a pas pu vérifier est écrit.

À savoir avant de lire

- La skill est gratuite, sous licence MIT. C'est un seul fichier texte, SKILL.md, de 290 lignes, sans aucun script.

- Le dépôt emilkowalski/skills affiche **52 812 étoiles** et 3 075 forks le 2 novembre 2026 à 7 h 55 (API GitHub).

- Elle est forte sur les détails : espacements, tailles de texte, arrondis, quantité de couleur, réaction des boutons, accessibilité du mouvement. Le style général de ta page, lui, reste en place.

- Une bonne partie du fichier parle de gestes (glisser, lancer, tirer un tiroir). Sur une page de texte, 7 sections sur 17 n'avaient rien à relire chez nous.

1. Ce que fait la skill, en clair

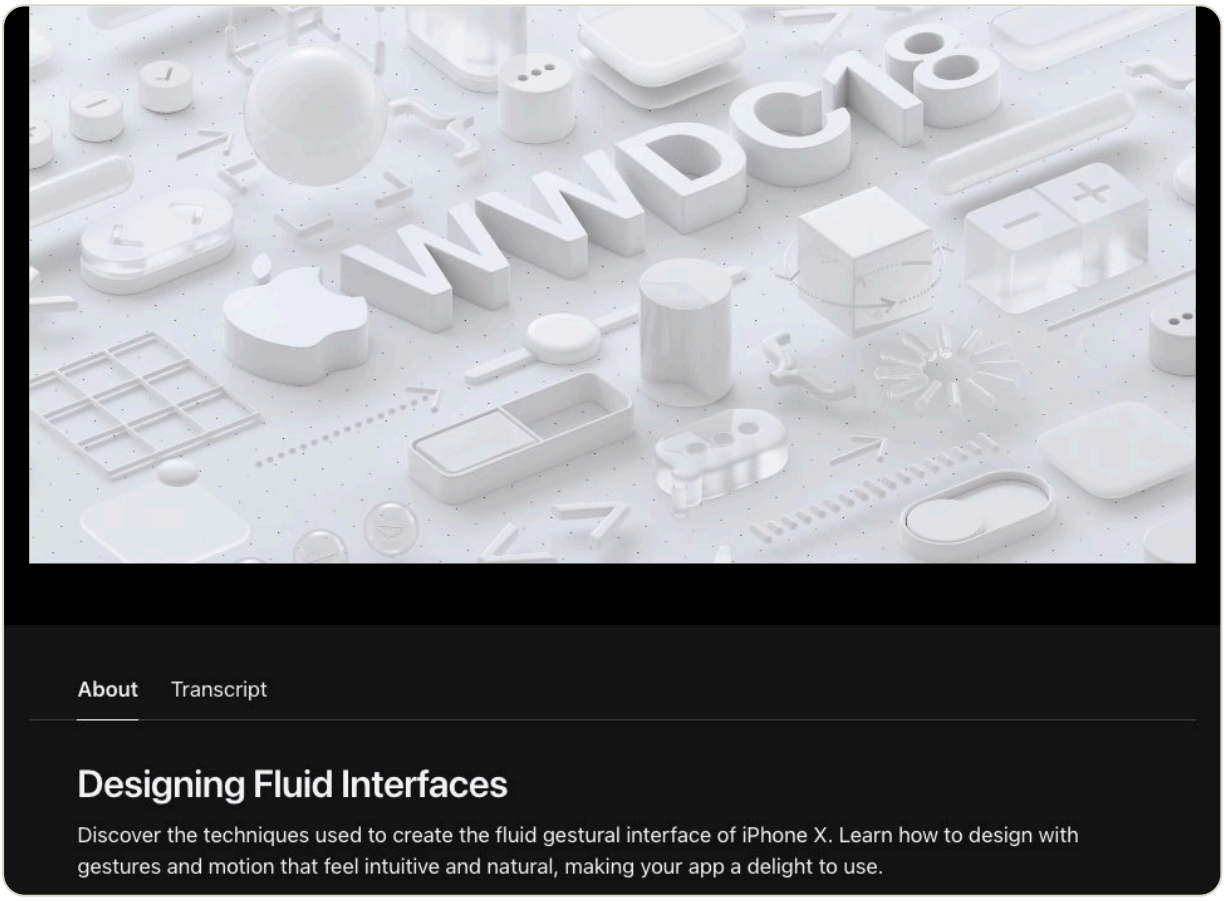
Une skill, c'est quoi

Un dossier de consignes que ton agent de code lit avant de travailler. Pour Claude Code, il se range dans `.claude/skills/` (un seul projet) ou dans `~/.claude/skills/` (tous tes projets). Celle-ci s'appelle `apple-design` et tient en un fichier de 23 036 octets.

D'où elle vient

Emil Kowalski l'a écrite à partir des conférences design de la WWDC d'Apple, surtout « Designing Fluid Interfaces » (WWDC 2018, session 803), puis a transposé chaque idée pour le web : CSS, événements du pointeur, `requestAnimationFrame`, bibliothèques de ressorts comme Motion. Le fichier cite aussi « The Details of UI Typography » (WWDC 2020), « Principles of Great Design » (WWDC 2026) et « Designing Audio-Haptic Experiences ».

Dans son README, Emil dit avoir travaillé chez Vercel et Linear. C'est lui qui a écrit Sonner, une bibliothèque de notifications pour React. La skill est entrée dans son dépôt le 9 juillet 2026.



La page Apple de la session 803 de la WWDC 2018, « Designing Fluid Interfaces », la source principale de la skill. Capture du 2 novembre 2026 à 0 h 40.

Ce que ça change, pour quelqu'un qui n'est pas designer

Une fois la skill chargée, Claude soigne ce que tu sens sur un site sans savoir le nommer :

- un bouton qui réagit à l'instant où ton doigt se pose ;

- un texte qui grossit quand tu agrandis la police de ton téléphone ;

- des espacements et des arrondis qui suivent une règle ;

- une couleur d'accent réservée à ce qui se clique ;

- des surfaces qui ont un peu de relief ;

- des animations qu'on peut interrompre, et qui se calment quand tu as demandé moins de mouvement dans les réglages.

L'idée qui relie tout le fichier : une interface paraît vivante quand le mouvement part de la position affichée à l'écran, garde la vitesse du doigt, prolonge son élan et peut être rattrapé à tout moment. Emil s'appuie pour ça sur des ressorts (springs), qu'on peut interrompre par nature.

2. L'installer, pas à pas

Ce qu'il te faut

- Claude Code, ou un autre agent de code. L'installeur gère Claude Code, Codex, Cursor, OpenCode et 75 autres agents, d'après son README.

- Node.js 22.20 ou plus, exigé par l'installeur. Pour vérifier ta version :

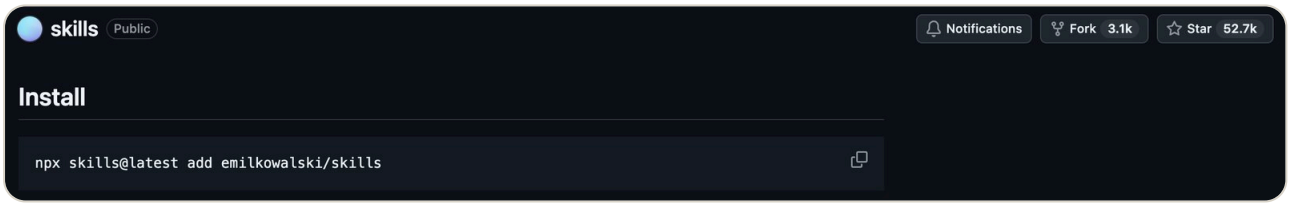
> node --version

La commande du README

Dans le dossier de ton projet :

> npx skills@latest add emilkowalski/skills

C'est la seule méthode écrite dans le README d'Emil. Le programme (le CLI `skills` de Vercel Labs, version 1.7.4 le 2 novembre) télécharge le dépôt, y trouve 16 skills et te demande lesquelles installer, et pour quel agent. Par défaut, il installe dans le projet courant. Ajoute `-g` pour que la skill serve dans tous tes projets.



L'en-tête du dépôt et la section Install du README, avec la commande. Capture du 2 novembre 2026 à 0 h 49 (52,7 k étoiles à cette heure-là).

N'installer que apple-design

> npx skills@latest add emilkowalski/skills --skill apple-design -a claude-code -g

`--skill` choisit la skill, `-a claude-code` choisit l'agent, `-g` l'installe pour tous tes projets. Ces options viennent de la documentation du CLI. Cette ligne précise, on ne l'a pas lancée : notre copie vient d'une installation à la main, le 10 août 2026.

Regarder la liste sans rien installer

> npx skills@latest add emilkowalski/skills --list

Lancée chez nous le 2 novembre à 7 h 56 : le CLI a cloné le dépôt, affiché « Found 16 skills » avec la description de chacune, et n'a rien installé dans notre dossier.

Les autres façons d'installer

Le README n'en donne pas d'autre. Il en existe pourtant deux.

À la main, avec Git. C'est ce qu'on a fait le 10 août 2026 :

> git clone --depth 1 https://github.com/emilkowalski/skills.git emil-skills

> mkdir -p ~/.claude/skills

> cp -R emil-skills/skills/apple-design ~/.claude/skills/

Ouvre ensuite une nouvelle session de Claude Code.

Sans rien installer, le CLI peut préparer un prompt qui contient la skill et lancer l'agent avec (commande `skills use`, documentation du CLI). Non testée par nous :

> npx skills@latest use emilkowalski/skills@apple-design --agent claude-code

Avant d'installer une skill, lis-la : elle tourne avec les droits de ton agent. Celle-ci ne contient que du texte et des exemples de code, aucun script. Le fichier est ici : [SKILL.md sur GitHub](https://github.com/emilkowalski/skills/blob/main/skills/apple-design/SKILL.md).

Bon à savoir sur la version

Depuis le 15 septembre 2026 (commit 0b85d4b), la version GitHub commence par un bloc « Initial Response » : appelée sans question, la skill répond par une phrase d'accueil et attend. Donne-lui donc toujours une tâche précise dans le même message. Notre copie, installée le 10 août, n'a pas ce bloc. C'est la seule différence avec la version GitHub du 2 novembre (8 lignes).

Les 15 autres skills du dépôt

La commande du README te les proposera aussi. On ne les a pas testées. Résumé des descriptions du README :

- **emil-design-eng** : la skill principale d'Emil, surtout de l'animation, avec des conseils de design.

- **animate** : construit une animation de zéro en choisissant courbe, durée et propriétés.

- **animate-expo** : la même chose pour React Native et Expo.

- **review-animations** : relit tes animations selon ses règles, de façon stricte.

- **improve-animations** : audite toutes les animations d'un projet et écrit des plans de correction.

- **find-animation-opportunities** : cherche les endroits qui gagneraient à bouger, et ceux qu'il faut laisser fixes.

- **remove-animations** : retire les animations inutiles (ajoutée le 28 octobre 2026, dernier commit du dépôt).

- **animation-performance** : répare les animations qui sautent des images.

- **animation-vocabulary** : donne le mot exact d'un effet pour mieux le demander à une IA.

- **mobile-native** : les petites corrections qui font qu'un site web se comporte comme une appli sur téléphone.

- **break-ui** : remplit ton interface avec les pires données (noms très longs, listes vides) pour voir ce qui casse.

- **pick-ui-library** : choisit une bibliothèque d'interface parmi celles qu'Emil utilise.

- **prototype** : construit plusieurs versions d'un élément et te laisse passer de l'une à l'autre.

- **ask-sonner** : guide de Sonner, sa bibliothèque de notifications.

- **write-swift** : écrire du Swift moderne.

3. Deux façons de s'en servir

Les deux essais du 2 novembre ont tourné dans Claude Code avec Claude Opus 5.5, sur l'abonnement Claude, sans clé d'API. Chaque essai tournait dans un dossier à part, qui ne contenait que la skill, plus la page à relire pour les revues.

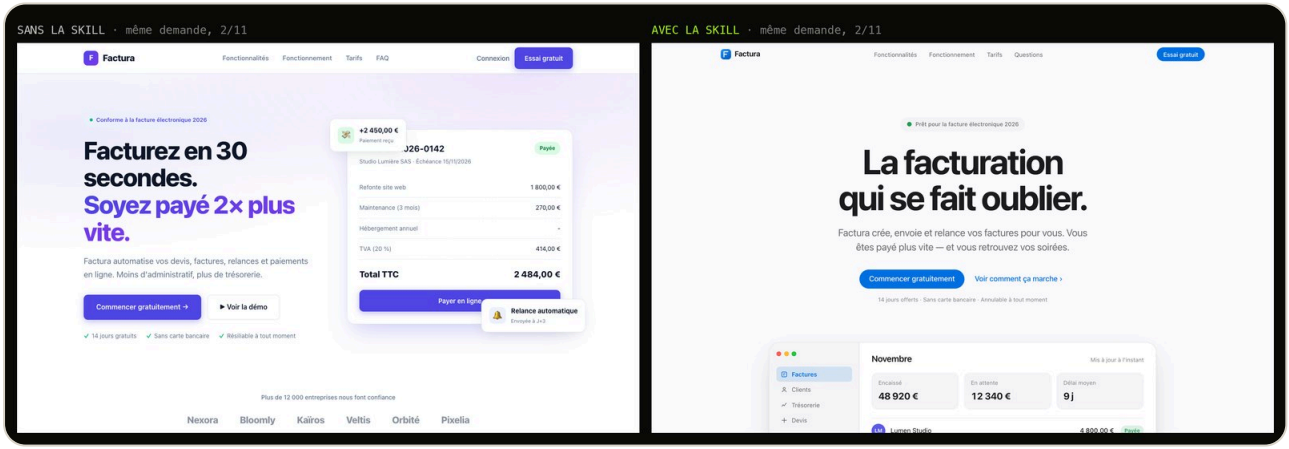
Façon 1 · Construire avec la skill

Dans la nuit du 2 novembre (entre 0 h 23 et 0 h 27), on a demandé trois landing pages à Claude, sans skill. Deux nous intéressent. Pour une startup de facturation, il a sorti « Factura », avec un titre en dégradé violet. Pour un studio de yoga, « Prana », avec un titre en serif italique sur fond crème : le look Claude dont parle le reel.

Le matin, on a relancé la même demande avec une phrase en plus, « Utilise la skill apple-design. ». Le prompt exact, à copier :

> Crée une landing page complète en un seul fichier HTML (CSS inclus) pour une startup SaaS de facturation. Utilise la skill apple-design. Réponds uniquement avec le code HTML.

Résultat en 2 min 12 : un titre centré, « La facturation qui se fait oublier. », en police système, avec un bouton bleu façon Apple. Le dégradé violet et le halo ont disparu. La page prévoit aussi un mode sombre.

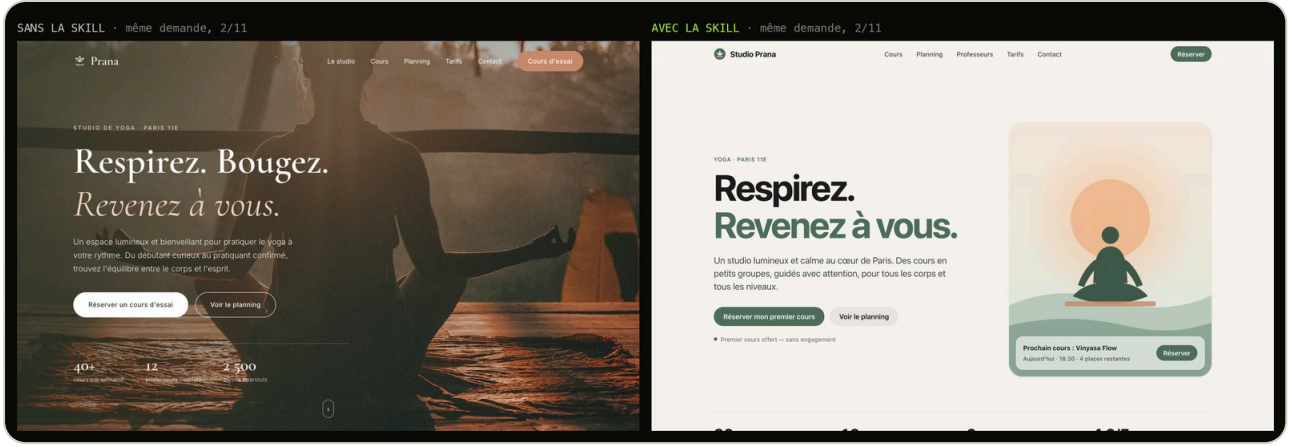


Même demande, sans la skill (à gauche) et avec la skill (à droite). Pages générées par Claude Opus 5.5 le 2 novembre 2026, captures à 1440 × 900.

Pour le yoga :

> Crée une landing page complète en un seul fichier HTML (CSS inclus) pour un studio de yoga. Utilise la skill apple-design. Réponds uniquement avec le code HTML.

Résultat en 3 min 15 : « Respirez. Revenez à vous. » en sans-serif gras, avec un vert sauge. Le serif italique a disparu, la photo sombre a laissé place à une illustration. Le fond reste crème clair (#f6f3ee dans le code), proche du crème de la version sans skill (#fbf8f3).



Même demande pour le studio de yoga, sans la skill (à gauche) et avec (à droite). 2 novembre 2026.

Pour ta propre page, change le métier et garde la phrase « Utilise la skill apple-design. ». On l'a toujours écrite. On n'a pas testé si Claude charge la skill tout seul quand on ne la nomme pas.

Façon 2 · Faire relire une page que tu as déjà

On a posé la page Factura, la violette, dans un dossier avec la skill, sous le nom `index.html`, puis collé ce prompt :

> Passe index.html en revue avec la skill apple-design, comme le ferait un designer d'Apple. Puis corrige tout ce que tu as relevé : écris la page corrigée dans apres.html (même contenu, même texte, un seul fichier HTML) et la liste des problèmes dans revue.md, chacun relié à sa section de la skill.

Remplace `index.html` par le fichier de ta page. Le prompt fait écrire la version corrigée dans un fichier à part : ta page d'origine reste intacte, et tu compares les deux.

Durée : 4 min 45 (de 6 h 55 à 7 h 00). Le fichier `revue.md` liste **30 problèmes corrigés**, chacun relié à sa section. Quelques-uns :

- aucun retour à l'appui sur les boutons (§1). Claude ajoute un léger rétrécissement à 97 % en 80 ms ;

- des cartes qui flottent en boucle toutes les 5 secondes, l'oscillation lente que la section 14 déconseille. Animation supprimée ;

- des compteurs animés qui affichent « 0 » tant que le script n'a pas tourné, donc « 0 entreprises » sans JavaScript (§1, §14). Les vrais chiffres sont écrits en dur ;

- `transition: all` sur les boutons et les cartes (§11) ;

- un trait fixe sous l'en-tête, même tout en haut de la page (§12) ;

- la carte « Le plus populaire » grossie à 103 %, ce qui rend son texte flou (§16). Remplacée par un double contour et une ombre plus profonde ;

- des tailles en px partout, qui ignorent la taille de texte choisie par l'utilisateur (§15). Tout passe en rem ;

- des emojis utilisés comme icônes (§16, Craft). Remplacés par des icônes dessinées en SVG ;

- des avatars blancs sur vert et sur orange, à un contraste d'environ 2,5 pour 1 (§16).

Il a aussi relevé 5 problèmes de contenu, qu'il a laissés tels quels puisque le prompt imposait le même texte :

- une carte annonce « +2 450,00 € » reçus alors que la facture totalise 2 484,00 € ;

- la facture est marquée « Payée » et propose « Payer en ligne » ;

- « Voir la démo » mène à la section Fonctionnement ;

- la ligne « Hébergement annuel » affiche un tiret à la place d'un montant ;

- trois boutons principaux se disputent le haut de la page.



Ce qui change en haut de la page Factura après la revue : le titre en dégradé passe en couleur pleine, les emojis des cartes deviennent des icônes dessinées. Captures à 1440 × 900, agrandies.

Le reste du haut de page bouge à peine, mise en page et couleur d'accent comprises.

Même prompt sur la page Prana : 8 min 17 et 26 problèmes, dont trois vrais bugs. Le menu mobile restait coincé dans l'en-tête dès qu'on avait fait défiler la page (un `backdrop-filter` sur l'en-tête en était la cause). La carte « Le plus populaire » n'était jamais mise en avant, son agrandissement étant écrasé par l'animation d'apparition. Et sans JavaScript, presque toute la page restait invisible.

Ce que ça coûte : au prix de l'API, le CLI de Claude Code estimait la revue de Factura à 2,47 dollars et celle de Prana à 3,26 dollars. Sur l'abonnement, ça s'est déduit de notre quota, sans facture en plus.

Si tu veux seulement la liste, sans corrections, une variante (non testée telle quelle) :

> Passe index.html en revue avec la skill apple-design, comme le ferait un designer d'Apple. Écris la liste des problèmes dans revue.md, chacun relié à sa section de la skill, sans modifier la page.

4. Les 17 principes, et quoi vérifier sur ton site

Le fichier compte bien 17 sections numérotées, comme le dit le reel. Pour chacune : ce qu'elle demande, résumé par nous, puis un test à faire toi-même. Les citations sont en anglais, telles qu'écrites dans la skill.

1. Réagir sans attendre

L'interface réagit à l'instant où le doigt appuie et suit le geste en continu, et chaque délai sur le chemin (anti-rebond, minuterie, environ 300 ms au tap) compte comme une régression. « Response is the foundation everything else is built on. »

À vérifier : sur ton téléphone, appuie sur un bouton sans relâcher. S'il ne bouge pas, il lui manque un état `:active`. La skill propose `scale(0.97)` en 100 ms. Sur notre guide, aucun élément ne réagissait à l'appui avant la revue.

2. Suivre le doigt un pour un

Ce que tu fais glisser reste collé au doigt, à l'endroit précis où tu l'as attrapé, même quand le pointeur sort de l'élément. « Touch and content should move together. »

À vérifier : si ton site a un carrousel, un curseur ou un tiroir, attrape-le par un bord. Il ne doit pas sauter pour se recentrer sous ton doigt. Rien à faire si ton site n'a rien à faire glisser.

3. Pouvoir interrompre toute animation

C'est le principe que la skill place au-dessus des autres : une animation doit pouvoir être rattrapée et renversée en vol, en repartant de la position affichée. Pour tout ce qui suit un geste, elle écarte donc les transitions CSS et les `@keyframes`. « The thought and the gesture happen in parallel. »

À vérifier : ouvre ton menu mobile et retape aussitôt pour le refermer. Il doit repartir d'où il est, sans finir d'abord son ouverture.

4. Animer avec des ressorts

Un ressort se règle avec l'amortissement (le rebond) et la réponse (la vitesse) : 1,0 par défaut, donc sans rebond, et 0,8 seulement après un geste qui avait de l'élan. Valeurs d'Apple citées : 1,0 et 0,4 pour déplacer, 0,8 et 0,4 pour tourner, 0,8 et 0,3 pour un tiroir. « Think of animation as a conversation between you and the object »

À vérifier : un menu qui apparaît ne doit pas rebondir. Le rebond se mérite par un geste.

5. Transmettre la vitesse du geste

Au lâcher, l'animation repart exactement à la vitesse du doigt, pour qu'aucune couture ne se voie entre le geste et l'animation. La skill donne le calcul (vitesse du geste divisée par la distance restante) pour les bibliothèques qui attendent une vitesse relative.

À vérifier : lance un tiroir ou une carte d'un coup sec. S'il freine net au lâcher puis repart, la vitesse se perd.

6. Viser là où le geste allait

La position finale se choisit d'après l'endroit où l'élan aurait mené l'élément, avec la fonction tirée de l'exemple de code d'Apple et un taux de décélération de 0,998. « Take a small input and make a big output. »

À vérifier : sur une feuille qui monte du bas de l'écran, un petit coup rapide vers le haut doit suffire à l'ouvrir en entier.

7. Repartir par où on est venu

Un panneau entré par la droite repart par la droite, et un menu naît du bouton qui l'a ouvert, avec des courbes symétriques à l'aller et au retour. « If something disappears one way, we expect it to emerge from where it came. »

À vérifier : ouvre puis ferme chaque menu, panneau et fenêtre de ton site. Même chemin à l'aller et au retour ?

8. Montrer la direction pendant le mouvement

Les étapes intermédiaires d'une animation annoncent déjà où elle va, comme les modules du Centre de contrôle de l'iPhone qui grandissent vers le doigt.

À vérifier : quand le contenu change d'onglet, il arrive du côté de l'onglet choisi. La revue de Prana l'a corrigé : le planning montait à la verticale sous des onglets alignés à l'horizontale.

9. Résister au bord

Au bout d'un carrousel, l'élément résiste de plus en plus au lieu de s'arrêter d'un coup : on comprend qu'il n'y a plus rien derrière, sans croire la page figée. La skill fournit la formule, avec une constante de 0,55.

À vérifier : tire ton carrousel au-delà de sa première image.

10. Les détails des gestes

Surligner à l'appui, valider au relâcher, prévoir environ 10 px de marge invisible autour des cibles et attendre environ 10 px de déplacement avant de choisir la direction d'un glissement. Le délai du double tap ne se paie que là où le double tap existe.

À vérifier : mesure tes plus petits boutons. Sur notre guide, « Copier » faisait 31 px de haut et la bascule « Lecture / Pour ton IA » 28 px.

11. Soigner chaque image de l'animation

N'animer que `transform` et `opacity`, que le navigateur anime sans recalculer la page, au rythme de `requestAnimationFrame`, et ajouter un léger flou sur les déplacements très rapides.

À vérifier : cherche `transition: all` dans ton CSS. Notre guide en avait sur 4 éléments en mobile, Factura sur ses boutons et ses cartes.

12. Matières et profondeur

Les barres et les feuilles flottantes sont translucides, le contenu défile dessous, les grandes surfaces ont plus de flou et d'ombre que les petites, et deux surfaces claires translucides ne s'empilent jamais. Sous l'en-tête, un léger fondu remplace le trait, seulement là où le contenu passe dessous. « Bigger surfaces should read as thicker »

À vérifier : remonte tout en haut de ta page. Un trait sous la barre du haut alors que rien ne passe dessous, c'est le cas qu'on a trouvé sur notre guide et sur Factura.

13. Le son et la vibration

D'après la conférence Designing Audio-Haptic Experiences : le retour part de l'événement qui le cause, le visuel, le son et la vibration tombent sur la même image, et on le réserve aux moments qui comptent (réussite, erreur, validation, élément qui se cale).

À vérifier : rien à faire si ton site n'émet ni son ni vibration. Sinon, coupe tout retour qui n'accompagne pas une vraie action.

14. Moins de mouvement, sans perdre le retour

Respecter `prefers-reduced-motion`, `prefers-reduced-transparency` et `prefers-contrast` en remplaçant glissements et rebonds par des fondus courts, tout en gardant les changements de couleur qui aident à comprendre. Éviter aussi les fonds qui bougent sur tout l'écran et les boucles lentes autour de 0,2 Hz, soit un cycle toutes les 5 secondes.

À vérifier : cherche ces trois mots dans ton CSS. Notre guide n'en contenait aucun, sur 165 Ko de CSS. Sur iPhone, active Réglages, Accessibilité, Mouvement, Réduire les animations, puis recharge ta page.

15. Typographie

D'après la conférence The Details of UI Typography (WWDC 2020) : l'approche (l'espace entre les lettres) devient négative sur les grands titres, -0,02 em dans l'exemple, et reste proche de zéro sur le texte, avec un interligne serré en haut et plus large dans le corps. Les tailles en `rem` suivent le réglage de l'utilisateur, et la police du système passe en premier choix. « Respect the user's text-size setting »

À vérifier : agrandis la taille de police dans les réglages de ton navigateur et recharge la page. Si ton texte ne grossit pas, tes tailles sont fixées en px. C'était le cas de notre guide.

16. Les fondations : les huit principes d'Apple

D'après la conférence Principles of Great Design (WWDC 2026) : but, contrôle laissé à l'utilisateur, responsabilité, familiarité, flexibilité, simplicité, soin du détail (Craft) et plaisir, ce dernier venant des sept autres. La section ajoute des règles concrètes : quatre sortes de retour (état, fin, alerte, erreur), des formulaires qui signalent l'erreur pendant la saisie, des écrans qui disent où tu es et comment en sortir, des libellés précis. « Nothing is random. »

À vérifier : la couleur de tes liens sert-elle aussi à des éléments qu'on ne peut pas cliquer ? Sur notre guide, oui : 31 éléments verts en mobile, dont l'avatar et l'encadré « L'essentiel ». Et sur téléphone, un long article a-t-il un sommaire ? Le nôtre n'en avait pas.

17. La méthode

Prototyper en version interactive, qu'on peut toucher, concevoir le mouvement et le visuel en même temps, tester avec de vraies personnes, et revoir chaque animation au ralenti. « You shouldn't be able to tell where one ends and the other begins. »

À vérifier : filme ton écran pendant que tu utilises ton site, puis repasse la vidéo image par image.

5. Notre test sur notre propre guide (1er novembre 2026)

Ce qu'on a fait

La page testée : [L'art du Claude maxxing](https://www.eigenflow.fr/fr/guides/claude-maxxing), un guide de notre site. On en a fait une copie statique : elle ne diffère de la page en ligne que de 0,00 % des pixels en mobile et de 0,31 % en desktop. Claude Code a lu la skill en entier, a relu la copie, puis a écrit ses corrections dans une seule feuille de style ajoutée (157 lignes, chaque bloc renvoie à son problème), plus une modification du HTML pour le sommaire mobile.

Le site en ligne n'a pas été touché pendant le test. Les mesures viennent du navigateur (styles calculés sur 22 blocs) et des captures, avant et après, aux mêmes réglages : mobile 360 px de large en densité 3, desktop 1440 × 900.

Pour ce premier test, on n'a pas archivé de prompt exact. Pour le refaire chez toi, prends celui de la façon 2.

Les 13 problèmes trouvés

1. Le texte ignorait le réglage de taille du navigateur : tout était fixé en px (§15).

2. L'approche était la même à toutes les tailles de titre (§15).

3. 7 espacements tombaient hors d'une grille de 4 px, dont 22 px en haut et 24 px sur les côtés du même encadré (§16, Craft).

4. 7 arrondis différents en mobile, 9 en desktop, sans échelle (§16, Craft).

5. Le vert servait aux liens et à des éléments qui ne se cliquent pas, comme « Testé par nous » et l'encadré « L'essentiel » (§16, Familiarité).

6. Deux verts dans le code (#9eff00 écrit 25 fois, à côté de l'accent #abff15), et un bouton d'en-tête bleu ardoise hors charte (§16).

7. Aucune profondeur : même filet de 1 px partout, zéro ombre, et un bloc de code plus noir que la page, qui se lisait comme un trou (§12).

8. Un filet de 1 px sous la barre du haut, visible même quand rien ne passait dessous (§12).

9. Aucune règle `prefers-reduced-motion`, `prefers-reduced-transparency` ni `prefers-contrast` dans 165 Ko de CSS, et une bordure de bouton qui tournait sans fin, un tour toutes les 2 secondes (§14).

10. Aucun élément ne réagissait à l'appui (§1).

11. Des transitions à durée fixe, dont des `transition: all` (§4, §11).

12. Des cibles trop petites : « Copier » à 31 px de haut, « Lecture / Pour ton IA » à 28 px (§10).

13. Pas de sommaire sur téléphone pour un guide de 13 minutes en 8 sections (§16, orientation).

Les chiffres, avant et après

- Taille du texte en mobile, police du navigateur réglée à 20 px : paragraphe 16 → 20 px, chapô 19 → 23,75 px, titre 40 → 50 px.

- Approche du grand titre en desktop (64 px) : -0,01 em → -0,02 em. Titres de section : 0 → -0,01 em.

- Espacements hors de la grille de 4 px : 7 → 0. Valeurs d'espacement différentes : 21 → 15 en mobile, 23 → 16 en desktop.

- Arrondis différents : 7 → 5 en mobile, 9 → 6 en desktop. Carte de fin : 40 → 28 px. Tuiles : 24 → 20 px.

- Éléments verts sur la page : 31 → 19 en mobile, 34 → 21 en desktop. Dans les trois premiers écrans mobiles : 8 → 4.

- **Pixels teintés de vert dans les trois premiers écrans mobiles : 30,29 % → 0,38 %.** En haut de page desktop : 8,82 % → 1,21 %.

- Verts différents dans le code : 2 → 1. Bouton d'en-tête : #020617 → #080807, le noir du site.

- Éléments qui réagissent à l'appui : 0 → 14 en mobile, 0 → 20 en desktop (`scale(0.97)` en 100 ms, la valeur de la skill).

- Éléments en `transition: all` en mobile : 4 → 0.

- Hauteur de « Copier » : 31 → 45 px. Bascule « Lecture » : 28 → 44 px.

- Règles pour les trois réglages d'accessibilité : 0 → 1 chacune.

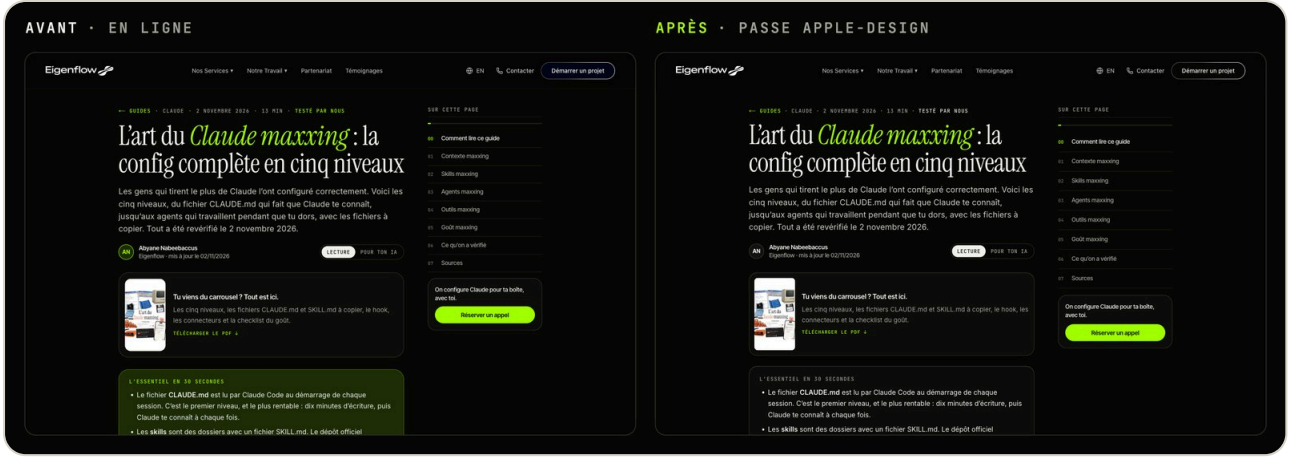
- Sommaire sur téléphone : absent → 8 entrées, repliable, 44 px de haut fermé.

Un pixel « teinté de vert », c'est un pixel dont le vert dépasse le rouge et le bleu de plus de 8 points sur 255, compté sur les captures. Les trois premiers écrans mobiles, ce sont les 5 760 premiers pixels de la capture de la page entière.

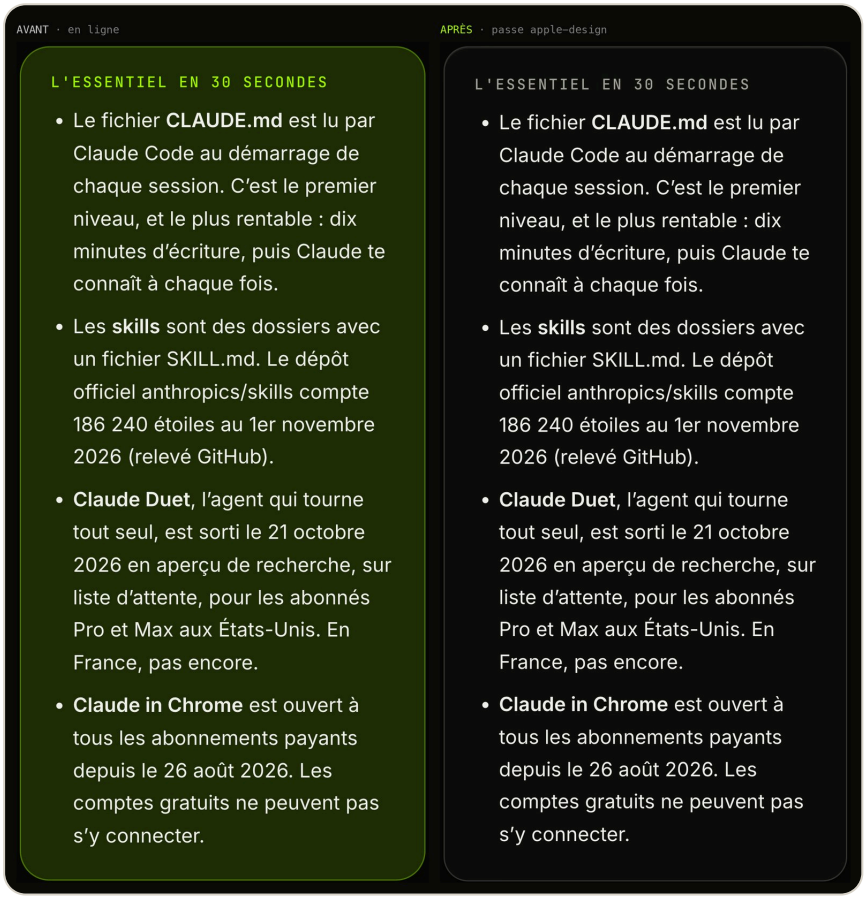
Ce que le sommaire coûte : l'encadré « L'essentiel » descend de 62 px.

Ce qui était déjà en règle : l'interligne qui baisse quand la taille monte (1,02 sur le titre, 1,65 sur le corps), les intertitres collés à leur paragraphe, les flèches du pied de page qui partent dans leur direction, la barre du haut déjà translucide.

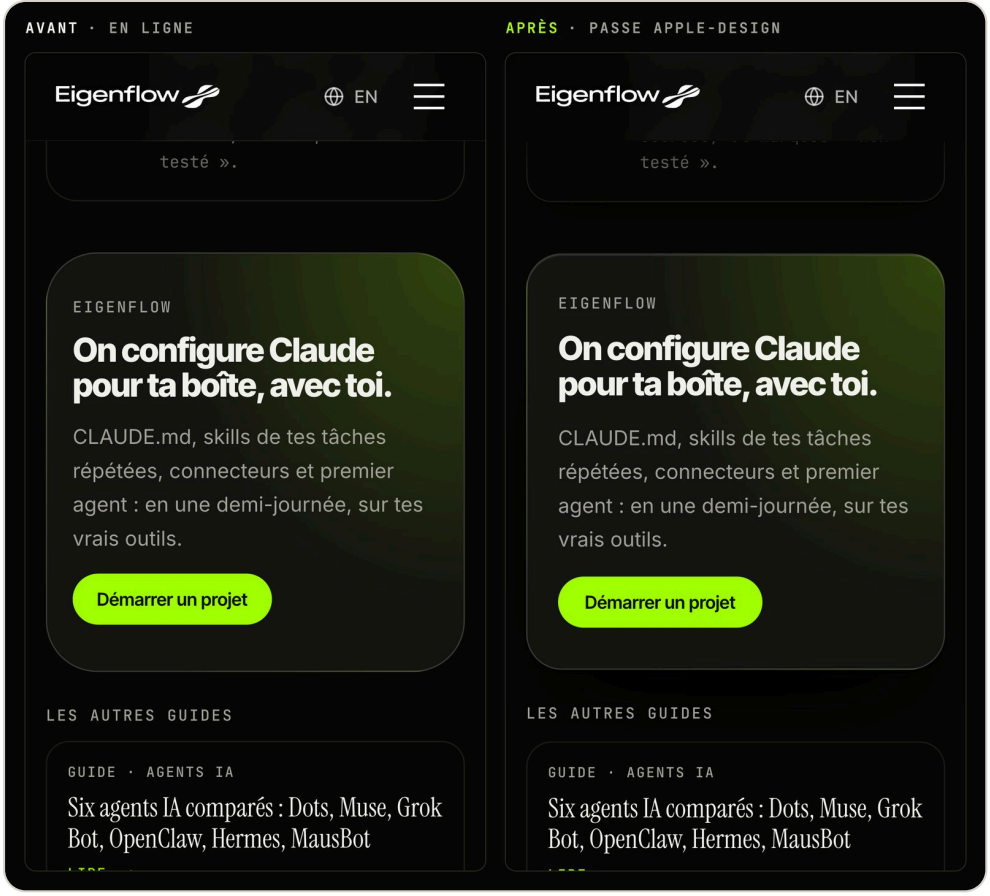
Ce qui ne s'appliquait pas : les sections 2, 3, 5, 6 et 9 (rien à faire glisser sur la page), la 13 (ni son ni vibration) et la 17 (une méthode de travail).



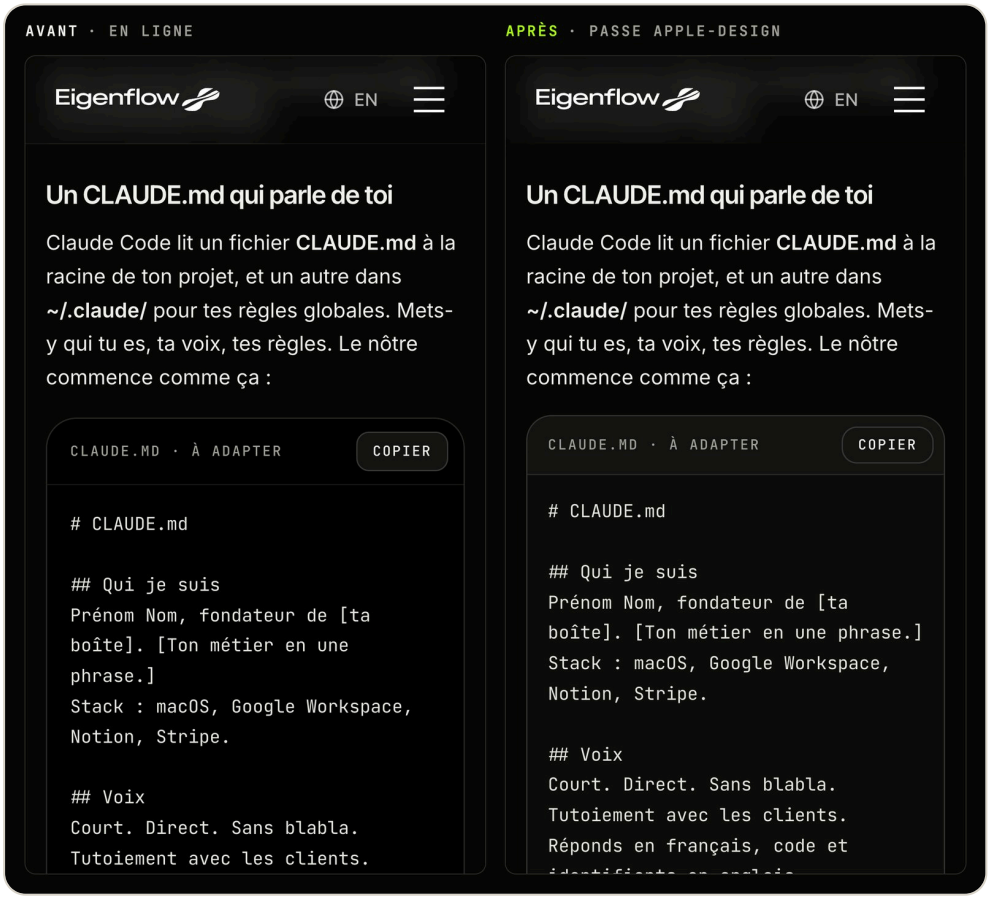
Le haut de notre guide en desktop, avant et après : l'avatar et « Testé par nous » sortent du vert, l'encadré « L'essentiel » passe en gris, la mise en page reste la même. Planche du 1er novembre 2026.



L'encadré « L'essentiel » en mobile, avant et après. C'est la plus grande surface verte du haut de page.



La carte de fin en mobile : arrondi de 40 à 28 px, une arête claire et une ombre pour lui donner du relief.



Le bloc à copier en mobile : son fond passe de #000, plus noir que la page, à #0e0e0c, et le bouton « Copier » grandit.

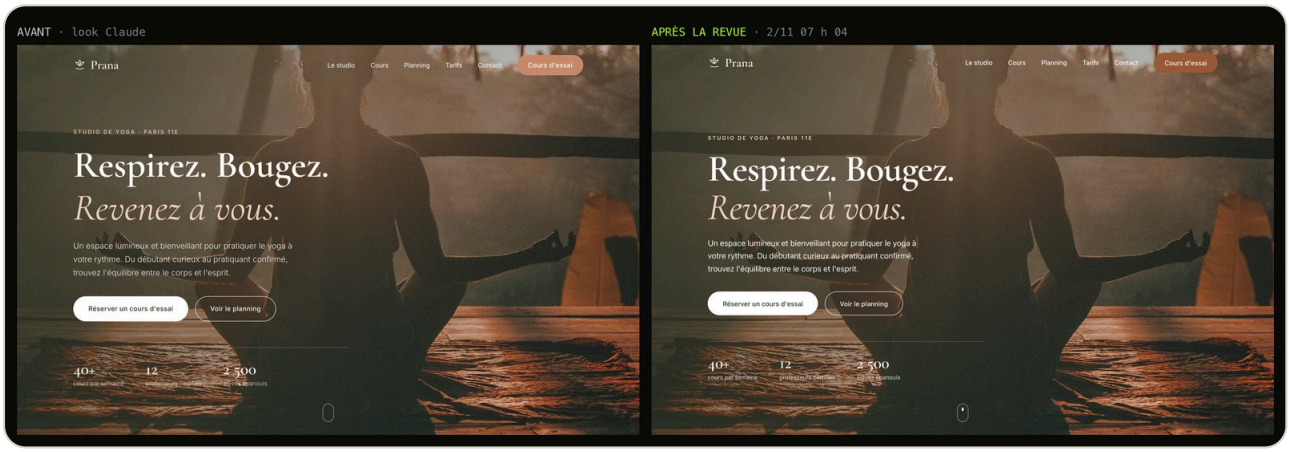
6. Les limites qu'on a constatées

La revue corrige les détails et garde le style

Sur Prana, la page au look Claude, le haut de page a presque l'air identique après la revue. Le serif italique et la photo sont toujours là, les couleurs aussi. Ce qu'on voit changer : le paragraphe passe d'une graisse fine à normale, et le bouton terracotta fonce pour gagner en contraste. Sur Factura, le dégradé et les emojis partent, la mise en page et l'indigo restent.

Notre propre guide garde aussi son titre en serif italique sur fond noir, ce que le reel appelle le look Claude. La skill ne l'a pas relevé.

Si tu veux changer de style, il faut le demander en plus. Même en construction, la page de yoga faite avec la skill garde un fond crème.



La page Prana avant et après la revue du 2 novembre : le style reste le même.

Un seul essai par page

On a fait un seul essai par demande. Sans la skill, le look Claude est sorti une fois sur trois pages (seul le yoga l'avait). Avec la skill, deux pages, deux essais. Ça montre ce qui peut arriver, sans valeur de mesure : chez toi, le résultat peut varier d'un essai à l'autre.

Claude n'a pas ouvert la page corrigée

Dans les deux revues, Claude l'écrit lui-même à la fin : il n'a pas ouvert `apres.html` dans un navigateur. Le rendu, le mode sombre et les animations n'ont été vérifiés qu'à la relecture du code. Nos captures « après » sont nos propres rendus dans Chrome. Ouvre toujours la page corrigée toi-même avant de la publier.

Ce que nos captures ne montrent pas

Le mouvement (l'appui, le ressort, la version en mouvement réduit) ne se voit pas sur une image fixe. Notre copie du guide n'avait plus de JavaScript : les boutons « Copier », la bascule et le menu n'y marchaient pas. La courbe du ressort de retour a été calculée par nous à partir des valeurs par défaut de la skill (amortissement 1,0, réponse 0,35 s). Et la règle « arrondi intérieur égal à l'arrondi extérieur moins la marge » vient de nous : la skill ne l'écrit pas, on l'a rattachée au principe Craft.

La version testée

Nos deux tests ont utilisé la copie installée le 10 août 2026 (282 lignes). La version GitHub du 2 novembre (290 lignes) n'en diffère que par le bloc « Initial Response ».

7. La checklist à copier

- Appuie sur chaque bouton au doigt : il réagit dès l'appui.

- Agrandis la police du navigateur : ton texte grossit avec (tailles en rem).

- Grands titres en approche négative (vers -0,02 em), corps de texte proche de 0.

- Interligne serré sur les titres, plus large sur le texte.

- Tous les espacements tombent sur une même grille (nous, 4 px).

- Les arrondis suivent une échelle courte. Dans une carte, l'arrondi intérieur vaut l'arrondi extérieur moins la marge.

- La couleur d'accent est réservée à ce qui se clique.

- Une seule valeur par couleur dans tout le code.

- Les grandes surfaces ont plus de relief que les petites (ombre, arête claire).

- Aucun trait sous la barre du haut quand rien ne passe dessous.

- Ton CSS contient `prefers-reduced-motion`, `prefers-reduced-transparency` et `prefers-contrast`.

- Aucun `transition: all`. On anime `transform` et `opacity`.

- Aucune animation lente en boucle infinie.

- Cibles tactiles d'au moins 44 px de haut (la valeur prise dans nos deux revues).

- Les menus sortent de leur bouton, repartent par le même chemin et se laissent interrompre.

- Un long article a un sommaire sur téléphone.

- Les formulaires signalent l'erreur dès qu'on quitte le champ.

- Le focus clavier se voit.

- Tu as ouvert la page corrigée dans un navigateur, en mobile et en desktop, avant de la publier.

8. Sources

Relevés du 2 novembre 2026, heure de Paris, sauf mention contraire.

La skill et son dépôt

- [Dépôt emilkowalski/skills](https://github.com/emilkowalski/skills), API GitHub interrogée le 2/11/2026 à 7 h 55 : 52 812 étoiles, 3 075 forks, licence MIT, 58 commits, 16 skills, dernier commit a81dbca du 28/10/2026 (« add remove-animations skill »), dépôt créé le 16/03/2026.

- [README du dépôt](https://github.com/emilkowalski/skills/blob/main/README.md), relu le 2/11/2026 à 7 h 55, identique à notre copie du 2/11/2026 vers 0 h 35. Une seule méthode d'installation y figure.

- [SKILL.md de apple-design](https://github.com/emilkowalski/skills/blob/main/skills/apple-design/SKILL.md), 290 lignes, 23 036 octets, relu le 2/11/2026 à 7 h 55.

- [Historique de apple-design](https://github.com/emilkowalski/skills/commits/main/skills/apple-design) : premier commit df49b5d du 09/07/2026, bloc « Initial Response » ajouté par 0b85d4b le 15/09/2026, dernière modification 85e8e23 le 15/09/2026.

- [Licence MIT](https://github.com/emilkowalski/skills/blob/main/LICENSE), copyright 2026 Emil Kowalski.

- [Sonner](https://sonner.emilkowal.ski), lien donné par le README.

L'installeur

- [CLI skills de Vercel Labs](https://github.com/vercel-labs/skills), README relu le 2/11/2026 : Node.js 22.20 ou plus, agents pris en charge.

- [Référence des commandes](https://github.com/vercel-labs/skills/blob/main/docs/commands.md) (options `--skill`, `-a`, `-g`, `--list`, commandes `use` et `update`) et [liste des agents](https://github.com/vercel-labs/skills/blob/main/docs/agents.md) (dossiers `.claude/skills/` et `~/.claude/skills/`), relues le 2/11/2026.

- Paquet npm `skills`, version 1.7.4, publiée le 19/10/2026 (registre npm, relevé le 2/11/2026).

- Notre essai de `npx skills@latest add emilkowalski/skills --list`, le 2/11/2026 à 7 h 56 : « Found 16 skills ».

Apple

- [Designing Fluid Interfaces, WWDC 2018, session 803](https://developer.apple.com/videos/play/wwdc2018/803/), page capturée le 2/11/2026 à 0 h 40.

- « The Details of UI Typography » (WWDC 2020), « Principles of Great Design » (WWDC 2026) et « Designing Audio-Haptic Experiences » : cités par la skill, pages non ouvertes par nous.

Nos tests

- Test du 1/11/2026 sur [notre guide Claude maxxing](https://www.eigenflow.fr/fr/guides/claude-maxxing) : copie statique, revue en 13 problèmes, mesures avant et après dans le navigateur.

- Test du 2/11/2026 : pages Factura et Prana générées sans skill par Claude Opus 5.5 entre 0 h 23 et 0 h 27, revues de 6 h 55 à 7 h 04, pages construites avec la skill de 7 h 05 à 7 h 09. Claude Code sur l'abonnement, aucune clé d'API.

Non vérifié

- L'installation par `--skill apple-design -a claude-code -g` et la commande `skills use` : tirées de la documentation, pas lancées par nous.

- Le chargement automatique de la skill quand on ne la nomme pas dans la demande.

- Les 15 autres skills du dépôt.

- Le rendu des animations corrigées sur un vrai téléphone.

Abyane · @abyane_nb · eigenflow.fr

Eigenflow