Eigenflow

SKILL CLAUDE CODE · GRATUITE · MIT

La skill « direction »

Diriger une animation en code au lieu de la décrire : la fiche en 7 blocs, la flamme en exemple, le test du 7 novembre 2026

PAR

Abyane Nabeebaccus

INSTAGRAM

@abyane_nb

ÉDITION

novembre 2026

Tu as commenté DIRECTION, voici le guide. Le fichier : direction.zip, 14 Ko. Dedans : `SKILL.md` (la méthode), `references/fiche-direction.md` (le gabarit à remplir et l'exemple rempli), `references/fire.mjs` (le moteur de la flamme, 240 lignes commentées), `examples/flamme.html` (la page prête), la licence MIT. On l'a écrite nous-mêmes le 7 novembre 2026 ; tu peux la modifier et la redistribuer.

Une skill est un dossier de consignes que Claude lit avant de travailler. Celle-ci lui impose de remplir une fiche de direction en sept blocs chiffrés avant d'écrire la première ligne de code, puis de fabriquer l'animation comme une fonction du temps, puis de la relire en planche.

À savoir avant de lire

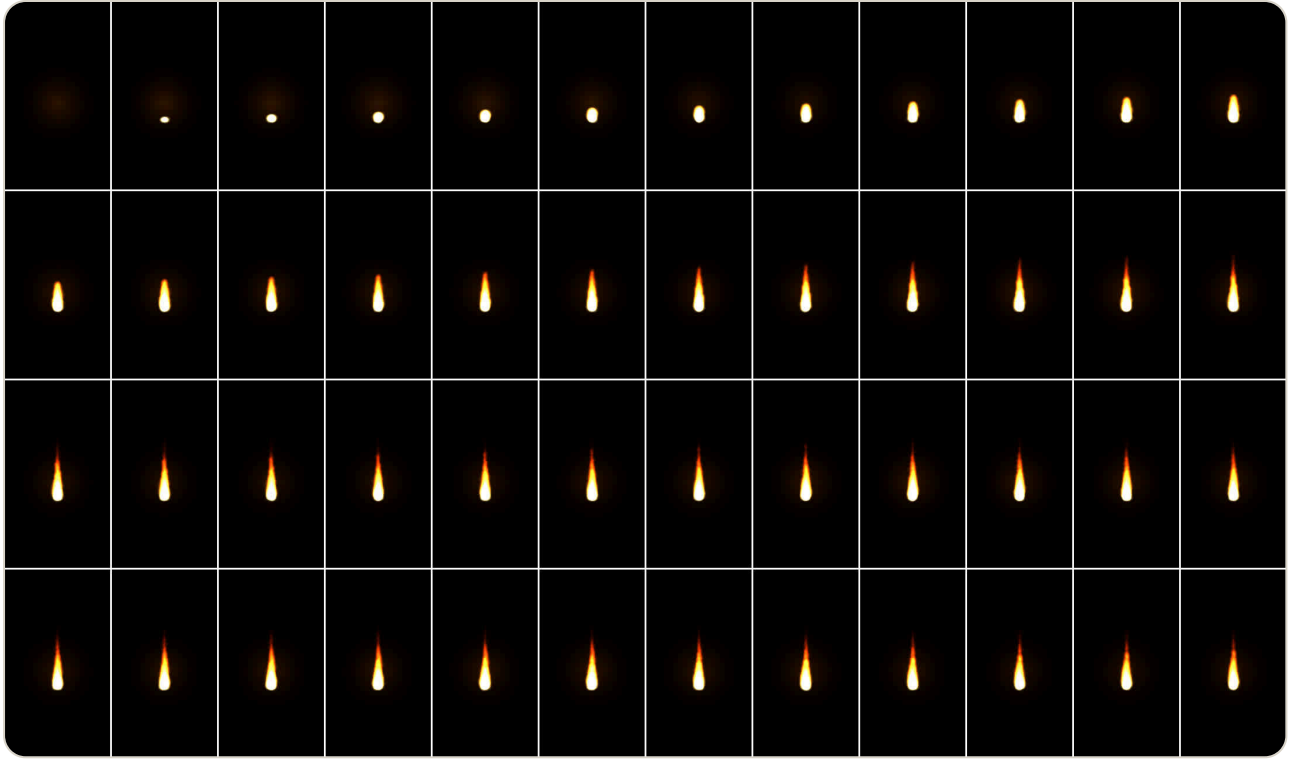
- Même demande, même modèle, session neuve de Claude Code (Opus 5.5, 7 novembre 2026, 23 h). Sans la skill : une page en 60 secondes et 2 tours, une goutte floue faite de 1 100 particules additives par seconde. Avec la skill : 97 secondes et 11 tours, une flamme en aplats avec sa fiche de direction en commentaire, et une palette chaude choisie par Claude, pas copiée de l'exemple.
- Au prix de l'API, le journal donne 0,19 $ sans la skill et 0,68 $ avec. Sur un abonnement Claude Pro ou Max, ça ne se paie pas à l'unité, mais la skill consomme 3,5 fois plus.
- Claude n'ouvre pas de navigateur dans une session en ligne de commande. Il l'a écrit lui-même à la fin du test : « je n'ai pas pu relire la planche ». La relecture en planche, c'est toi qui la fais, et la partie 4 te donne la commande.
- Le moteur d'exemple est en canvas 2D, sans dépendance. Pour du CSS, du SVG, Three.js ou Remotion, la fiche s'applique telle quelle ; le code de la flamme ne se copie pas, il se réécrit.
- Les deux pages du test sont montrées dans la vidéo telles que Claude les a écrites, capturées image par image à temps virtuel. Aucune retouche.

---

1. Décrite ou dirigée : la même flamme deux fois

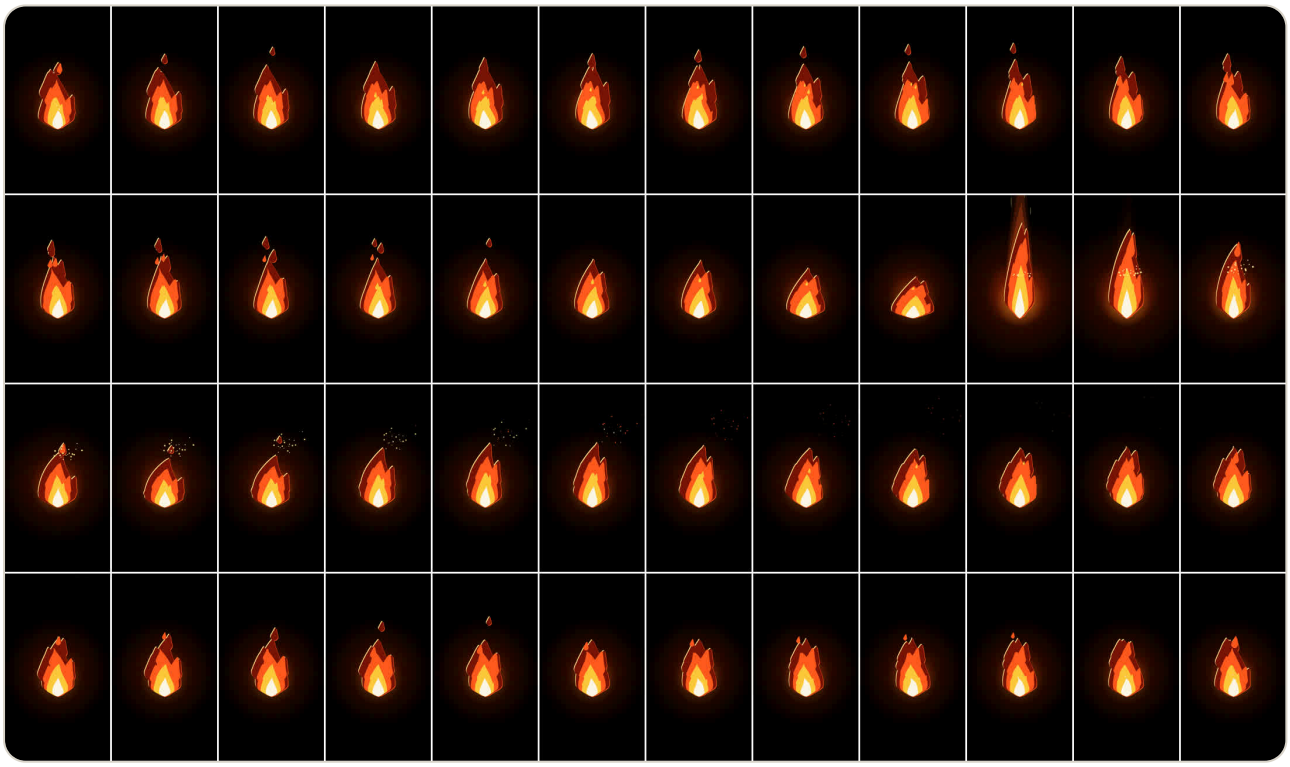
La demande, mot pour mot : « Fais-moi une animation de flamme en HTML canvas : une seule page, sans dépendance, fond noir, la flamme au centre, en boucle. » Pour le second essai, une phrase en plus : « Utilise la skill direction. »

Sans la skill, Claude a choisi des particules : 1 100 par seconde, additionnées à l'affichage, un halo flou. Le résultat ressemble à ce que donnent presque toutes les demandes de ce genre : une tache lumineuse qui monte.



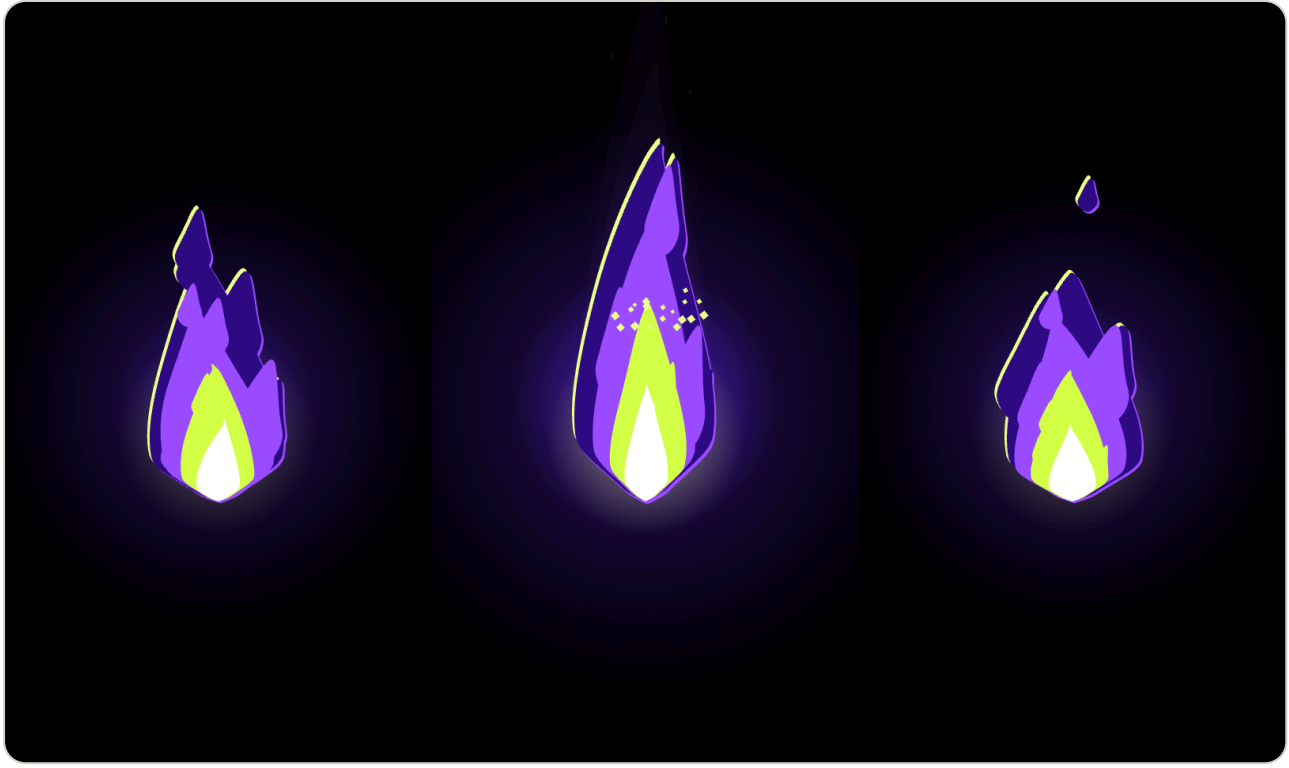
Quarante-huit images d'une boucle de la page écrite sans la skill, à 12 images par seconde : une goutte qui naît, grandit et reste une goutte.

Avec la skill, Claude a d'abord lu la fiche et le moteur, puis écrit une page de 14 Ko qui commence par sa propre fiche de direction, en commentaire : style cel, quatre tons chauds (`#7A1408`, `#FF5A1F`, `#FFC93C`, `#FFF8E7`), un liseré `#FFE08A`, un halo orange, moment clé à 0,42 de la boucle, anticipation sur 3 poses, détente sur 2, ressort amorti, 12 poses par seconde, braises vers le haut à droite, période de 4 secondes.



Quarante-huit images d'une boucle de la page écrite avec la skill : les langues montent le long des bords, la flamme se tasse, jaillit, revient au calme, les braises partent en haut à droite.

Ce n'est pas la palette de l'exemple fourni (violet, lime, blanc). Claude a gardé la méthode et pris ses propres décisions : c'est exactement ce que la skill demande.



Trois poses du moteur d'exemple fourni dans la skill : au repos, au moment du coup avec ses images de traînée, et pendant le retour au calme.

---

2. La fiche de direction, bloc par bloc

Chaque bloc force une décision chiffrée. Les nombres ci-dessous sont ceux de la flamme d'exemple ; ils sont dans `references/fiche-direction.md` avec la raison de chacun.

Le look et la palette

Un style nommé (cel, peint, pixel ou filaire), jamais « réaliste » : en code, le réaliste rate et le dessiné réussit. Quatre tons du plus sombre au plus clair, le cœur en plus clair, et un liseré : un trait clair décalé vers la lumière de 6 px à gauche et 8 px en haut, pas un contour noir. Un seul halo, du ton extérieur, large (0,95 fois la hauteur de l'objet), qui respire avec le moment clé.

Le moment clé, et quand il frappe

Une boucle a un seul temps fort, placé à une fraction précise de la période. Pour la flamme, 0,42, soit 1,68 s sur 4 s. Tout le reste prépare ou digère ce moment. Une boucle sans moment clé est un écran de veille.

Anticipation, détente, retour au calme

Avant le coup, l'objet fait le contraire du coup : la flamme se tasse (hauteur fois 0,70, largeur fois 1,24) pendant 3 poses, soit 0,25 s. Le coup dure 2 poses (hauteur fois 1,72, largeur fois 0,76). Puis un ressort amorti sur 1,15 s : amplitude 0,46, amortissement en e^(-4,2 τ), 2,1 oscillations. Une oscillation fait « rebond de dessin animé », trois font « gelée ».

Images de traînée, animé sur deux

24 images par seconde, une pose dessinée sur deux : 12 poses par seconde, obtenues en quantifiant le temps avant de dessiner. Le lissé à 60 images fait « rendu par ordinateur ». Pendant les 2 poses de la détente, la pose s'étire (fois 1,9, 1,5 et 1,22 pour trois fantômes à 16, 30 et 45 % d'opacité) avec quatre stries de vitesse. Au repos, une respiration de plus ou moins 3,5 %, deux cycles par boucle, pour que rien ne soit jamais immobile.

Où va l'œil, et l'espace vide

Une seule direction de lecture : les 22 braises partent toutes vers le haut à droite (angle -1,05 rad, plus ou moins 0,25). L'objet occupe le tiers central ; le reste du cadre reste vide. Pas de grille décorative, pas de vignettage.

Une boucle parfaite

Période fixe, nombre entier de poses : 4 s, 48 poses. Tout bruit est périodique parce qu'il est échantillonné sur un cercle, (x, cos 2πφ, sin 2πφ) avec φ le temps divisé par la période. Jamais `Math.random()` ni `Date.now()` pendant le rendu : l'image à t = 0 est l'image à t = 4 s, à l'identique.

Ce qu'il faut éviter

Une boule (une tache floue au centre), tout ce qui est boueux (dégradés qui se mélangent, plus de cinq couleurs, additif à outrance), la symétrie, la vitesse constante, et le bruit uniforme : trois calques qui bougent du même bruit ne font qu'un calque.

---

3. Installer la skill

Dézippe `direction.zip`. Tu obtiens un dossier `direction` avec `SKILL.md` à la racine.

Dans Claude Code, pour toutes tes sessions :

> mkdir -p ~/.claude/skills && cp -R direction ~/.claude/skills/direction

Pour un seul projet, le même dossier va dans `.claude/skills/direction` à la racine du projet. C'est ce qu'on a fait pour le test : un dossier vide qui ne contenait que la skill.

Dans l'appli Claude, d'après la doc d'Anthropic : Personnaliser, Skills, bouton +, puis importe le zip. L'exécution de code doit être activée dans les réglages. On n'a pas testé la skill dans l'appli : la relecture en planche demande une capture image par image, que l'appli ne fait pas.

---

4. S'en servir

Décris l'objet en une phrase et nomme la skill :

> Fais-moi une animation de flamme en HTML canvas : une seule page, sans dépendance, fond noir, la flamme au centre, en boucle. Utilise la skill direction.

Claude lit `SKILL.md`, la fiche et le moteur, puis écrit un fichier HTML dont les premières lignes sont sa fiche de direction remplie. Lis cette fiche avant d'ouvrir la page : si un bloc n'a pas de nombre, renvoie-le avec « remplis le bloc 3 avec des poses, pas avec des adjectifs ».

La page expose `window.showAt(t)` et `window.DURATION`. Pour la relire en planche sans rien installer, ouvre-la dans Chrome et, dans la console, dessine les poses d'une boucle à la suite :

> for (let i = 0; i < 48; i++) { window.showAt(i / 12); await new Promise(r => setTimeout(r, 250)); }

Pour une vraie grille d'images, le script `capture.mjs` qu'on a utilisé est décrit dans la partie 6. Ce que tu cherches sur la grille : une pose immobile de plus de 0,5 s, une forme qui redevient une boule, un ton de trop, une image 48 différente de l'image 0.

---

5. Ce que le test a coûté, et ce qu'il n'a pas fait

Les deux sessions ont tourné en ligne de commande, en mode `--print`, sur notre abonnement, sans aucun réglage personnel, avec le modèle Opus 5.5 et la version 2.1.319 de Claude Code. Les outils autorisés : lire, écrire, modifier, chercher, et charger une skill. Pas de terminal, pas de navigateur.

- Sans la skill : 60 s, 2 tours, 6 583 tokens produits, 0,19 $ au prix de l'API.
- Avec la skill : 97 s, 11 tours (la skill, la fiche, le moteur, l'exemple, l'écriture, trois corrections), 0,68 $ au prix de l'API.

Ce que Claude n'a pas fait, et l'a dit : ouvrir la page et regarder la planche. La skill le lui demande, mais sans navigateur il ne peut pas. Si tu lui donnes un terminal et Chrome, il pourra ; sinon c'est ton travail, et il prend deux minutes.

Un essai par condition ne prouve rien de général. On montre deux pages, écrites avec la même demande à dix minutes d'écart, et tu juges.

---

6. Adapter la méthode à autre chose qu'une flamme

Le moteur `fire.mjs` sépare le temps dirigé du dessin. La fonction `beat(t)` ne sait rien du feu : elle renvoie, pour un instant, l'échelle en largeur et en hauteur, la phase (anticipation, détente, retour au calme, repos) et un coefficient de « coup ». Pour de l'eau, une fumée, une étincelle ou un logo, garde `beat()` et le bruit périodique, et remplace seulement le corps et les langues par la forme de ton objet.

Pour capturer une page en images, notre script tient en 40 lignes : il lance Chrome sans écran, remplace `requestAnimationFrame` et `performance.now` par un temps qu'il avance lui-même, puis capture une image par pose. Il est dans le dossier du reel, pas dans la skill, parce qu'il dépend de `puppeteer-core` et de Chrome. Si tu le veux, demande-le en commentaire.

---

7. Les limites, en clair

- La skill force les décisions. Si tu réponds « fais comme tu veux » à chaque bloc, tu retrouves la goutte floue.
- Elle est écrite pour des objets seuls sur un fond : une flamme, une goutte, un logo. Pour une scène à plusieurs objets, remplis une fiche par objet et une fiche pour la caméra.
- L'exemple est en canvas 2D. Claude sait transposer les règles dans d'autres techniques, mais le code de l'exemple ne tourne pas ailleurs.
- Le test du 7 novembre a été fait une fois par condition, sur un seul modèle. Un autre jour, un autre modèle ou un autre prompt donneront autre chose.

Eigenflow, 7 novembre 2026. La skill est sous licence MIT ; tout ce guide vient de notre test et de nos fichiers.
