Eigenflow

GUIDE

Une scène d'explainer animé, écrite en code par Claude

Le prompt exact, le passage en MP4 sans logiciel de montage, et ce que Claude a vraiment livré en 13 minutes.

PAR

Abyane Nabeebaccus

INSTAGRAM

@abyane_nb

ÉDITION

novembre 2026

Tu as commenté SCÈNE, voici le prompt. C'est le message exact qu'on a envoyé à Claude Code le 13 novembre 2026 à 20 h 46. Copie-le tel quel.

À COLLER

```
Refais en code une scène d'explainer animé dans le style de Kurzgesagt, sur le système immunitaire. Tu inventes tes propres personnages, tu ne copies pas les leurs.

La scène, 20 secondes :
1. Une écharde rouillée perce la peau. Des bactéries entrent par la plaie et commencent à se multiplier.
2. Les macrophages arrivent, gros et calmes, et avalent les bactéries une par une.
3. Les neutrophiles accourent en nombre et se jettent sur les dernières.
4. La plaie se referme, un macrophage repu sourit.

Format 1920 × 1080, dessin vectoriel plat, couleurs franches, personnages ronds avec de grands yeux expressifs, quelques textes courts en français à l'écran. Un seul fichier index.html, sans image ni bibliothèque externe : tout est dessiné en code (SVG ou canvas). L'image affichée dépend seulement d'une fonction window.seek(t), t en secondes de 0 à 20, pour qu'on puisse l'exporter image par image en MP4.
```

Un seul message, 149 mots. 13 min 24 s plus tard, Claude avait écrit un fichier de 785 lignes qui dessine une scène de 20 secondes, image par image. Ce guide explique comment adapter ce prompt à ton sujet, où le lancer, comment passer le fichier en MP4 sans logiciel de montage, et ce que Claude a vraiment livré, défauts compris.

Pages officielles et prix relus le 13 novembre 2026. On n'a aucun lien avec Kurzgesagt ni avec Anthropic.

À savoir avant de lire

- Le résultat est une scène de 20 secondes, sans son. Leur vidéo sur le système immunitaire dure 7 min 49, avec voix off, musique et recherche. Notre scène en fait un peu plus de 4 %.
- Claude a écrit la scène en 13 min 24 s, dont 10 min 22 s de réflexion avant sa première action. Le passage en MP4 a pris 90 s de plus.
- Claude Code est inclus dès Claude Pro, à 20 $ par mois hors taxes (environ 18 € HT), et Opus aussi. Notre test a tourné sur la formule Ultra, sans clé API : non testé sur Pro.
- Le fichier tient tout seul : 0 image, 0 bibliothèque, 0 lien vers internet. Il s'ouvre dans un navigateur, sans rien installer.

- Pour le MP4, deux voies : un enregistrement d'écran (intégré à macOS et à Windows 11), ou notre script de 47 lignes, qui sort une vidéo 1920 × 1080 image par image.
- Claude n'a jamais vu sa scène. Il l'a écrite sans l'ouvrir, et il le dit.
- Kurzgesagt est une marque déposée. Le style est libre, leurs personnages et leur musique ne le sont pas (partie 6).

---

1. Le prompt, et comment l'adapter

Ce qu'il y a dedans

Quatre blocs, dans cet ordre :

1. Le quoi : une scène d'explainer animé, un style, un sujet, et une limite (« Tu inventes tes propres personnages »).

2. La scène minutée : une durée totale et quatre temps numérotés, une action par temps. On n'a pas donné l'horaire de chaque temps : Claude l'a fixé seul (partie 4).

3. Le format : la taille de l'image, le style décrit avec des mots (dessin plat, couleurs franches, personnages ronds, grands yeux) et la langue des textes.

4. La contrainte technique : un seul fichier `index.html`, sans image ni bibliothèque, piloté par une fonction `window.seek(t)`.

Le dernier bloc sert à l'export. Pour chaque instant t, la page dessine l'image exacte de cet instant. Un script n'a plus qu'à la photographier 30 fois par seconde (partie 3). Sans `window.seek(t)`, il ne te reste que l'enregistrement d'écran.

Pour un autre sujet (pas testé tel quel)

Remplace ce qui est entre crochets, garde le reste.

À COLLER

```
Fais en code une scène d'explainer animé sur [ton sujet]. Tu inventes tes
propres personnages.

La scène, [20] secondes :
1. [Le déclencheur : ce qui arrive, et où.]
2. [La réponse : qui arrive, ce qu'il fait.]
3. [La montée : plus de monde, plus vite.]
4. [La fin : ce qui est réglé, et une expression sur un visage.]

Format 1920 × 1080, dessin vectoriel plat, couleurs franches, personnages ronds
avec de grands yeux expressifs, quelques textes courts en français à l'écran.
Un seul fichier index.html, sans image ni bibliothèque externe : tout est
dessiné en code (SVG ou canvas). L'image affichée dépend seulement d'une
fonction window.seek(t), t en secondes de 0 à [20], pour qu'on puisse l'exporter
image par image en MP4. À l'ouverture, la scène se joue en boucle.
```

Deux différences avec notre prompt :

- notre prompt nommait un studio en plus de décrire le style. Ce modèle décrit le style sans nom. C'est plus prudent si tu publies (partie 6), mais on ne sait pas si le résultat est aussi proche.
- la dernière phrase n'était pas dans notre prompt. Claude a fait la lecture en boucle de lui-même. Demande-la : elle sert pour l'enregistrement d'écran.

Un exemple de sujet : « le trajet d'un colis entre l'entrepôt et ta porte », avec le colis qui part, le livreur qui le prend, la tournée qui s'accélère, le colis posé et un client content. Garde une action par temps, comme dans notre test.

En vertical, pour un reel (pas testé)

Remplace la ligne du format par celle-ci :

> **À COLLER**
>
> ```
> Format vertical 1080 × 1920, pour un reel. Garde les personnages et les textes
> au centre de l'image : en bas et sur le bord droit, Instagram affiche la légende
> et les boutons par-dessus.
> ```

La scène de notre test ne passe pas en vertical toute seule : elle garde son format 16:9 dans la fenêtre, donc une fenêtre verticale donnerait deux bandes noires. Il faut redemander la scène en vertical, puis donner la taille au script de rendu (partie 3).

---

2. Où le lancer

Claude Code, ce qu'on a utilisé

Claude Code est inclus dans toutes les formules payantes de Claude, Pro compris, d'après la page des tarifs. Opus, le modèle de notre test, est disponible dès Pro.

Pro coûte 20 $ par mois hors taxes, ou 17 $ par mois à l'année, sur la page des tarifs relue le 13/11/2026. Au cours BCE du jour, ça fait environ 18 € HT. La même page, affichée en euros le même jour, disait 18 € HT par mois (15 € à l'année). Le montant TTC s'affiche au paiement. Ultra, la formule de notre test, coûte 200 $ par mois hors taxes (180 € sur la page en euros).

Le plus simple :

1. Mets Claude Code à jour avec `claude update` : Opus 5.5 demande la version 2.1.280 ou plus.

2. Crée un dossier vide, ouvre un terminal dedans et tape `claude`.

3. Tape `/model opus` pour choisir Opus.

4. Colle le prompt. Claude te demandera sans doute la permission d'écrire `index.html` : accepte.

Sans terminal, l'appli Claude Desktop fait aussi tourner Claude Code, d'après sa documentation. On ne l'a pas fait pour ce test.

Notre commande exacte

Pour refaire le test dans les mêmes conditions (sans tes réglages ni tes skills, avec quatre outils de fichiers seulement), enregistre le prompt dans un fichier `prompt.txt` et lance, dans un dossier vide :

À COLLER

```
claude -p --safe-mode --restricted \
  --model claude-opus-5-5 \
  --tools "Read,Write,Edit,Glob" \
  --permission-mode acceptEdits < prompt.txt
```

Les barres obliques inversées en fin de ligne servent à couper la commande pour la lire : c'est la syntaxe du terminal sur Mac et Linux. Sous Windows, mets tout sur une ligne.

`--safe-mode` coupe tes réglages, tes skills, tes plugins et tes serveurs MCP. `--restricted` retire les outils qui lancent du code et garde Claude dans le dossier. Claude ne pouvait donc que lire, lister et écrire des fichiers. C'est pour ça qu'il n'a pas pu ouvrir sa page pour la regarder.

La limite d'usage

Sur Pro, il y a une limite par session de 5 heures et une limite par semaine, communes au chat et à Claude Code. Anthropic ne donne pas de nombre de jetons : la page des tarifs dit seulement que la vitesse à laquelle tu atteins la limite dépend du modèle et de la longueur du travail.

Notre scène a produit 83 409 jetons en sortie, dont 60 051 de réflexion, sur Opus. On ne sait pas quelle part d'une session Pro ça représente : notre test a tourné sur Ultra, qui donne 10 fois l'usage de Pro chaque semaine, sans limite de 5 heures pour les comptes vérifiés. Ton compteur est dans Paramètres > Utilisation, et `/status` dans Claude Code.

Le chat de Claude, en alternative (pas testé)

Dans le chat (claude.ai ou l'appli de bureau), Claude peut faire une page web d'un seul fichier sous forme d'artéfact, qui s'ouvre à côté de la conversation. D'après l'aide de Claude, les artéfacts existent dès la formule gratuite, à condition d'activer la création de fichiers dans les paramètres, rubrique Capacités (Capabilities en anglais). Mais Opus demande Pro.

Ce qu'on ne sait pas : si la scène tourne aussi bien dans l'artéfact, et comment en récupérer le fichier HTML. L'aide ne décrit l'export que pour les artéfacts faits depuis un modèle. Le modèle Motion, lui, exporte en MP4, mais il est en bêta et réservé aux formules Team et Enterprise. Si tu passes par le chat, l'enregistrement d'écran de la partie 3 reste possible.

3. Passer le fichier en MP4, sans logiciel de montage

La voie simple : enregistrer l'écran

1. Ouvre `index.html` dans Chrome, par exemple en glissant le fichier dans une fenêtre. Notre scène démarre seule et tourne en boucle.

2. Agrandis la fenêtre. La scène s'adapte à sa taille et garde son format 16:9.

3. Enregistre au moins 20 secondes, puis recoupe le début et la fin pour garder un tour complet.

Sur Mac, d'après Apple : Maj + Commande + 5, puis « Enregistrer la partie sélectionnée ». Cadre la scène et clique sur Enregistrer. Pour arrêter : Commande + Contrôle + Échap. Le fichier arrive sur le bureau, en .mov. Clique sur la vignette qui apparaît dans le coin de l'écran pour le recouper.

Sur Windows 11, d'après Microsoft : touche Windows + Maj + R ouvre l'Outil Capture d'écran en mode vidéo. Sélectionne la zone, puis Démarrer et Arrêter. La vidéo sort en MP4.

Le .mov du Mac passe tel quel sur Instagram (MOV ou MP4 d'après la documentation de son API) et sur YouTube. Pour un vrai MP4, une ligne de ffmpeg suffit (installation juste en dessous) :

```
À COLLER
ffmpeg -i "Enregistrement.mov" -c:v libx264 -pix_fmt yuv420p -an scene.mp4
```

Cette voie a deux limites. L'enregistrement suit l'écran en temps réel : si l'ordinateur rame, la vidéo saccade. Et la taille de la vidéo dépend de ta fenêtre. La voie exacte n'a aucun des deux problèmes.

La voie exacte : notre script

Il ouvre la page dans Chrome sans fenêtre, appelle `window.seek(t)` pour chaque image (30 par seconde), prend une capture à chaque fois, puis ffmpeg assemble les 600 captures en MP4. La vitesse de ta machine ne change rien au résultat : chaque image est calculée à part.

Il te faut :

- Google Chrome, installé à l'endroit habituel ;
- Node.js 22.12 ou plus, la version que demande puppeteer-core 25 ;
- ffmpeg. Sur Mac avec Homebrew : `brew install ffmpeg`. Sur Windows : `winget install --id Gyan.FFmpeg --exact`. Sur Ubuntu : `sudo apt install ffmpeg`.

Ensuite, mets `index.html` dans un dossier, enregistre à côté le script ci-dessous sous le nom `rendu-mp4.mjs`, ouvre un terminal dans ce dossier et tape ces deux lignes, l'une après l'autre. La première installe puppeteer-core, la bibliothèque qui pilote Chrome : elle ne télécharge pas de navigateur, elle se sert du tien. La seconde lance le rendu.

> npm install puppeteer-core

> node rendu-mp4.mjs index.html

La vidéo arrive dans `rendu/scene.mp4`, et les 600 images dans `rendu/images`, que tu peux effacer ensuite. Pour une autre durée ou un autre format, ajoute dans l'ordre le dossier, la durée en secondes, les images par seconde, la largeur et la hauteur. En vertical : `node rendu-mp4.mjs index.html rendu 20 30 1080 1920`. Si Chrome n'est pas à l'endroit habituel, mets son chemin dans la variable `CHROME_PATH` avant de lancer.

À COLLER

```
// Page HTML pilotée par window.seek(t) -> vidéo MP4, image par image.
// node rendu-mp4.mjs index.html [dossier] [durée] [images/s] [L] [H]
import puppeteer from "puppeteer-core";
import { mkdirSync } from "node:fs";
import { execFileSync } from "node:child_process";
import { join, resolve } from "node:path";
import { pathToFileURL } from "node:url";

const args = process.argv.slice(2);
const [src, outDir = "rendu"] = args;
const DUR = Number(args[2] || 20), FPS = Number(args[3] || 30);
const W = Number(args[4] || 1920), H = Number(args[5] || 1080);
const N = Math.round(DUR * FPS);
const frames = resolve(outDir, "images");
mkdirSync(frames, { recursive: true });

// Chrome à l'endroit habituel, sinon son chemin dans CHROME_PATH
const chrome = process.env.CHROME_PATH;
const browser = await puppeteer.launch(chrome
  ? { executablePath: chrome, headless: true }
  : { channel: "chrome", headless: true });
const page = await browser.newPage();
await page.setViewport({ width: W, height: H, deviceScaleFactor: 1 });
page.on("pageerror", (e) => console.error("Erreur :", e.message));
await page.goto(pathToFileURL(resolve(src)).href, { waitUntil: "load" });
await page.evaluate(() => document.fonts && document.fonts.ready);
const ok = await page.evaluate(() => typeof window.seek === "function");
if (!ok) {
  console.error("Pas de fonction window.seek(t) dans la page.");
  await browser.close();
  process.exit(1);
}
const t0 = Date.now();
for (let i = 0; i < N; i++) {
  await page.evaluate((t) => window.seek(t), i / FPS);
  const name = "f" + String(i).padStart(4, "0") + ".jpg";
  await page.screenshot({
    path: join(frames, name), type: "jpeg", quality: 92 });
}
await browser.close();
const out = resolve(outDir, "scene.mp4");
execFileSync("ffmpeg", ["-v", "error", "-y",
  "-framerate", String(FPS), "-i", join(frames, "f%04d.jpg"),
  "-c:v", "libx264", "-crf", "16", "-pix_fmt", "yuv420p",
  "-movflags", "+faststart", out], { stdio: "inherit" });
const s = Math.round((Date.now() - t0) / 1000);
console.log(N + " images en " + s + " s. Vidéo : " + out);
```

Sur notre Mac (Chrome 157, Node.js 22.22, puppeteer-core 25.11, ffmpeg 8.0.1), le script a sorti les 600 images et la vidéo en 90 s, puis en 101 s au second essai : 20 s en 1920 × 1080, 30 images par seconde, sans son. La vidéo est identique, image pour image, à celle de notre premier rendu. Pas testé sur Windows ni sur Linux. `npm install` prendra la dernière version de puppeteer-core (25.17.0 le 13/11/2026, même exigence de Node.js), qu'on n'a pas essayée.

Le fichier écrit par Claude et ce script sont dans scene-explainer-claude-code.zip. Décompresse-le, ouvre un terminal dans le dossier, installe puppeteer-core et lance la commande ci-dessus.

Ajouter un son

La scène sort muette. Pour poser une voix ou une musique dessus, toujours sans logiciel de montage :

```
À COLLER
ffmpeg -i rendu/scene.mp4 -i voix.mp3 -c:v copy -c:a aac -shortest scene-son.mp4
```

L'image n'est pas réencodée, et la vidéo s'arrête avec le plus court des deux fichiers. On a vérifié la commande avec un son de test, pas avec une vraie voix.

---

4. Ce que Claude a vraiment livré

Les chiffres

- 13 min 24 s entre l'envoi du message et sa réponse (de 20 h 46 à 20 h 59, heure de Paris), dont 10 min 22 s de réflexion avant sa première action.
- trois tours : il regarde le dossier (vide), écrit `index.html` d'un seul coup, puis répond. Aucune relecture, aucune correction.
- 83 409 jetons en sortie, dont 60 051 de réflexion.
- au prix public de l'API, 2,34 $ d'après Claude Code. Sur l'abonnement d'Abyane, rien de facturé en plus.
- le fichier : 785 lignes, 37 Ko, un seul canvas, 194 appels de dessin, 0 image, 0 lien vers internet, 0 bibliothèque.
- le rendu : 600 images, aucune erreur dans la page, un MP4 de 20 s en 1920 × 1080, sans son.

Le déroulé, tel que Claude le décrit

Claude a découpé nos quatre temps en cinq, qui se chevauchent :

1. de 0 à 2,3 s : une écharde tachée de rouille, trois bactéries accrochées dessus, plonge dans la peau. L'image tremble. Les couches de peau sont nommées à gauche.

2. de 2,3 à 6,8 s : les bactéries tombent dans le derme et se divisent jusqu'à 16. Des ondes jaunes « signal d'alerte » partent vers le vaisseau sanguin.

3. de 5,6 à 12,2 s : trois macrophages sortent du vaisseau, gros, bleus, paupières mi-closes. Chacun avale trois bactéries et grossit à chaque repas.

4. de 11 à 14,2 s : quatorze neutrophiles orange, sourcils froncés, foncent en zigzag et attaquent les sept dernières bactéries à deux contre une.

5. de 14,6 à 20 s : la peau se referme (une croûte, puis une petite cicatrice rose), les neutrophiles pâlissent et disparaissent, et la caméra zoome sur un macrophage repu, yeux fermés, joues roses, petits cœurs.

Il a ajouté de lui-même une légende par étape avec le mot-clé en couleur, des étiquettes qui pointent les cellules, des bactéries qui passent de narquoises à paniquées, et un tirage « au hasard » fixé d'avance : la même valeur de t donne toujours la même image. Sans ça, l'export image par image tremblerait.

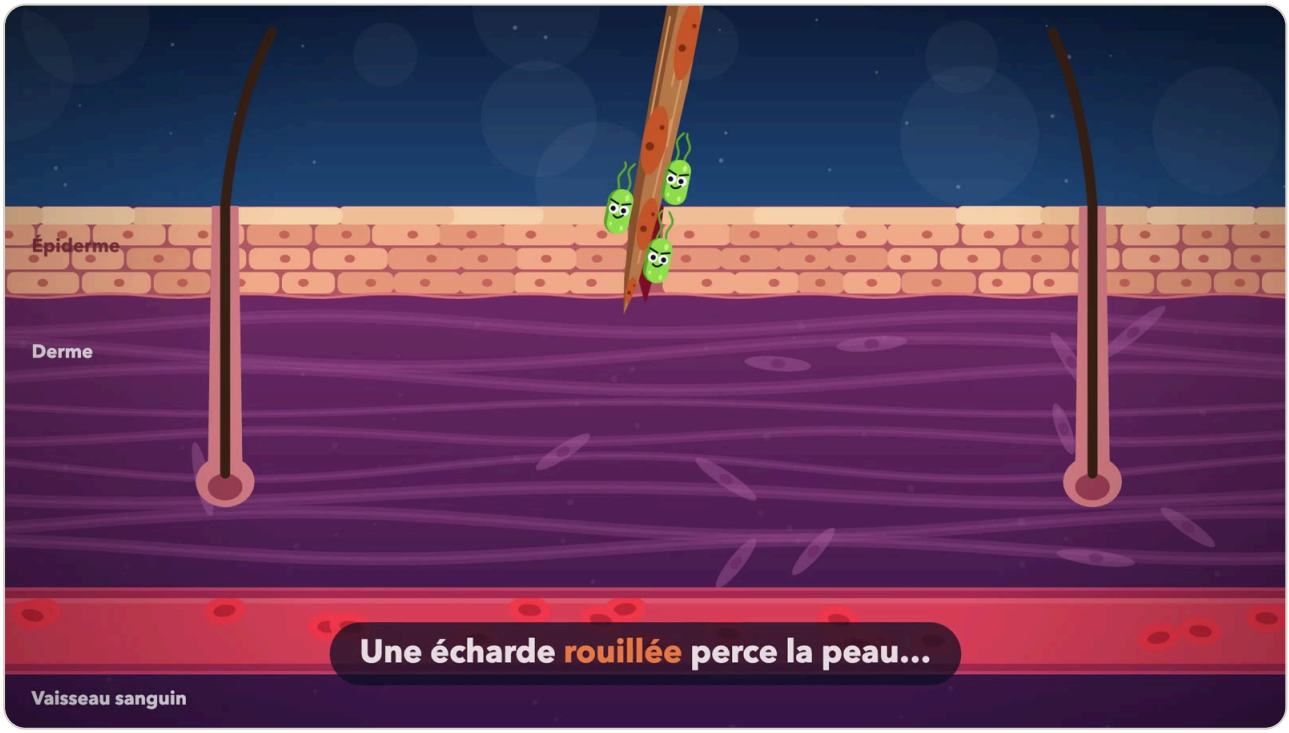
Sa mise en garde

À la fin de sa réponse, Claude a écrit : « Je n'ai pas pu l'ouvrir dans un navigateur : rien n'a encore été vu à l'écran. J'ai seulement relu le code à la main. Ouvre-le une première fois pour vérifier le rendu et la lisibilité des textes. »

C'était exact : avec nos options, il n'avait aucun moyen de lancer un navigateur. Il a aussi prévenu que la police dépend de la machine (Avenir Next sur Mac, sinon Segoe UI, Helvetica ou Arial).

En l'ouvrant, on a vu que tout s'affiche du premier coup et que les textes se lisent. Trois défauts : les noms des couches de peau sont petits, vers 13 s les neutrophiles s'empilent sur les macrophages, et le zoom final coupe les deux macrophages des bords. On n'a rien retouché : les images ci-dessous sortent du MP4 tel quel.

Quatre images de la scène



À 2,2 s : l'écharde rouillée entre dans la peau avec ses bactéries.



À 6,5 s : les bactéries se sont multipliées, les trois macrophages arrivent du vaisseau sanguin.



À 12,6 s : les neutrophiles foncent sur les dernières bactéries.



À 19,5 s : la peau est refermée, zoom sur un macrophage repu.

---

5. Ce que ça ne remplace pas

- Le son. La scène est muette : ni voix, ni musique, ni bruitages. Leur vidéo a une voix off professionnelle et une musique faite pour la chaîne.
- La recherche et la vérification. Leur studio compte 8 personnes à l'écriture et à la vérification des faits, d'après sa page équipe. Claude a dessiné ce qu'on lui a décrit, et on n'a pas fait vérifier la biologie de la scène.
- La durée. Leur vidéo de 2014 fait 7 min 49 et avait 54,7 millions de vues le 13/11/2026. Notre scène fait 20 s. D'après leur making-of de février 2020, une de leurs vidéos demande environ 1 200 heures de travail, de la recherche à la musique.
- Le regard. Claude n'a jamais vu sa scène. Les défauts de la partie 4, c'est nous qui les avons vus.

En 2014, le studio était bien plus petit : sa frise date ses premiers salariés à plein temps de 2015. Sa page équipe compte aujourd'hui 67 personnes, boutique, jeu vidéo et administration compris.

Ce que ce prompt fait bien : une scène courte, dans un style plat, pour illustrer une idée, sans logiciel d'animation, sans générateur d'images et sans crédits vidéo.

---

6. Le cadre

On n'est pas avocats. Ce qui suit se lit au conditionnel.

- Le style est libre. « Le droit d'auteur ne protège pas les idées ou les concepts, mais leur expression concrète », rappelle l'INPI. Leurs personnages, leurs dessins, leur musique et

leurs textes sont des expressions : n'y touche pas. Notre prompt le demandait à Claude (« Tu inventes tes propres personnages, tu ne copies pas les leurs »).

- Kurzgesagt est une marque. « kurzgesagt in a nutshell » est enregistrée à l'OMPI (n° 1464635, au nom de In a nutshell - kurzgesagt GmbH, jusqu'au 13/12/2028) pour l'Union européenne et les États-Unis, entre autres pour les dessins animés et la production de films d'animation. N'utilise ni leur nom ni leur logo pour présenter ou vendre ta vidéo.
- Ne présente pas ta scène comme la leur, ni comme une vidéo faite avec eux.
- Instagram demande l'étiquette « Contenu IA » pour une vidéo photoréaliste ou un son réaliste, créés ou modifiés numériquement, avec l'IA par exemple. Une animation en dessin plat n'en fait pas partie : l'aide d'Instagram cite, parmi les contenus qu'on n'est pas obligé d'étiqueter, « La vidéo d'un paysage extérieur, réalisée dans un style proche de la bande dessinée ». Si tu poses dessus une voix off IA réaliste, l'étiquette devient obligatoire (exemple de la même page : « Un reel raconté avec une voix off réaliste générée par l'IA »). Elle s'active au moment de publier, avec « Ajouter une mention IA ». Meta peut aussi la poser seul s'il détecte des signaux d'IA dans le fichier.
- YouTube demande de déclarer, dans « Utilisation de l'IA », le contenu IA qui semble réaliste. Il cite parmi les cas à ne pas déclarer une animation générée par IA « dans une vidéo entièrement animée ». Une musique générée par IA, elle, est à déclarer. YouTube peut aussi ajouter seul une mention dans la description pour les contenus IA animés.
- On n'a aucun lien avec Kurzgesagt ni avec Anthropic.

---

7. La méthode

Guide écrit le 13 novembre 2026. Test lancé le même jour à 20 h 46 (heure de Paris) avec Claude Code 2.1.280 en ligne de commande (claude -p), modèle Opus 5.5, sur l'abonnement d'Abyane (formule Ultra), sans clé API, dans un dossier vide hors de nos projets, options --safe-mode et --restricted, outils Read, Write, Edit et Glob seulement. Un seul essai, un seul message, sortie gardée telle quelle. Durées, tours, jetons et coût lus dans le journal de session ; coût au prix public de l'API, calculé par Claude Code.

Rendu en MP4 avec notre script (puppeteer-core 25.11.0, Chrome 157 sans profil, ffmpeg 8.0.1), refait avec la version de ce guide : même vidéo, image pour image. Aucun outil d'animation tiers lancé. Chiffres YouTube relevés avec yt-dlp 2026.10.26 (métadonnées seulement, aucune vidéo téléchargée).

Page équipe de kurzgesagt.org, tarifs et aide de Claude, documentation de Claude Code et de Puppeteer, aides d'Apple, de Microsoft, d'Instagram et de YouTube, registre de l'OMPI et page de l'INPI relus le 13 novembre 2026. Euros au cours de référence BCE du même jour (1 € = 1,1052 $), hors taxes. Pas testé : Claude Pro, le chat de Claude, la variante verticale, l'enregistrement d'écran, Windows et Linux.