Eigenflow

GUIDE · ENNUYEUX

Un mot, une vidéo : le setup Claude Code

Le fichier de consigne et le script de rendu à copier, la commande et les permissions, nos trois tests chiffrés et ce qu'il faut relire avant de publier.

PAR

Abyane Nabeebaccus

INSTAGRAM

@abyane_nb

ÉDITION

novembre 2026

Tu as commenté ENNUYEUX, voici le setup. Il tient en deux fichiers : SETUP.md, une consigne de 3 312 caractères, et render.mjs, un script de rendu de 30 lignes. Tu donnes un mot à Claude Code. Il cherche sur le web ce que les gens ignorent du sujet, choisit une question, vérifie ses faits, écrit le script, fait la voix, cale chaque sous-titre sur son mot, dessine les scènes en 3D dans une page web et sort une vidéo verticale d'une trentaine de secondes. Les deux fichiers sont à la fin du guide, prêts à copier.

On l'a testé le 21 novembre 2026, avec Claude Code 2.1.327 et Opus 5.5, sur trois mots : « Ascenseurs. », « Agrafeuses. » et « Peinture beige. ». Trois vidéos finies, sans qu'on touche à rien, en 33 à 48 minutes chacune. Ensuite, on a revérifié leurs faits un par un. La partie 4 dit ce qui tient, et ce qui ne tient pas.

À savoir avant de lire

- Un seul mot, oui, mais pas tout seul : le setup est chargé avant le mot, et c'est lui qui impose l'ordre du travail. L'angle, puis les faits, le script, la voix, les scènes et le rendu.
- Compte 30 à 50 minutes par vidéo, ordinateur allumé. Au prix de l'API, chacun de nos tests aurait coûté entre 5,36 $ et 7,00 $. Avec un abonnement Claude, c'est pris sur ton quota, rien de plus n'est facturé.
- Tel quel, le setup tourne sur un Mac à puce Apple. Sur Windows ou Linux, deux étapes changent, la voix et le temps des mots (partie 2).
- La voix du Mac (commande say, voix Thomas) sonne robotique, et la licence de macOS n'autorise pas sa publication. Elle sert de maquette. Pour publier, il faut une autre voix (partie 5).
- Relis les faits avant de publier. Dans nos tests, Claude a cité des pages qu'il n'avait pas pu ouvrir, et deux de ses phrases allaient plus loin que ses sources.

1. Ce que fait le setup, et pourquoi dans cet ordre

Le setup découpe le travail en six étapes, et Claude ne peut pas en sauter une :

1. L'angle. Il cherche sur le web ce que les gens ignorent du sujet et écrit cinq angles possibles, chacun sous forme de question. Il garde celle dont on veut le plus connaître la réponse, à condition qu'elle soit vérifiable et surprenante. Chaque fait doit tenir sur deux sources sérieuses. Une anecdote célèbre qui ne tient pas sort, avec sa raison, dans un fichier angle.md.

2. Le script. Sept à dix répliques courtes. La première pose la question d'un trait, chacune fait avancer la réponse (mais, donc), la dernière répond.

3. La voix. Une prise par réplique, assemblées en un seul fichier voix.wav. Puis le temps de chaque mot, recalé sur le texte exact du script, dans words.json.

4. Les scènes. Une seule page web de 1080 × 1920 qui dessine l'image exacte de n'importe quelle seconde. Une idée visuelle par réplique, en 3D, et chaque scène naît d'un objet de la précédente.

5. Le rendu. Chrome, sans fenêtre, prend 30 images par seconde ; ffmpeg les assemble avec la voix dans final.mp4.

6. Le contrôle. Une planche de vignettes, qu'il regarde lui-même, puis deux tours de corrections au plus.

Pourquoi l'angle d'abord : un mot seul ne dit pas quoi raconter. Le setup force Claude à choisir une question avant d'écrire une ligne, et à jeter ce qu'il ne peut pas prouver. Sur nos trois tests, il a sorti trois questions précises et écarté à chaque fois au moins quatre idées, dont une anecdote célèbre qui ne tenait pas : Otis et son « All safe, gentlemen », Louis XV et sa première agrafeuse, Steve Jobs et ses deux mille beiges. L'idée de départ est simple : une question dont on veut la réponse se regarde jusqu'à la réponse. On ne l'a pas mesuré, ces trois vidéos n'ont pas été publiées.

Pourquoi les scènes en dernier : la page dessine ce que dit la voix au mot près, grâce au fichier des temps. Sans voix finie, pas de temps, donc pas de scènes.

---

2. Ce qu'il te faut avant

Claude Code et un abonnement

Claude Code demande un compte Pro, Max, Ultra, Team, Enterprise ou Console. Le forfait gratuit ne l'inclut pas (documentation de Claude Code, lue le 21 novembre).

Les prix en France le 21 novembre, sur la page des tarifs de Claude, hors taxes (la page précise que les taxes ne sont pas comprises) :

- Pro : 18 € par mois au mois, ou 15 € par mois à l'année (180 € payés d'avance). Soit 21,60 € TTC au mois, avec la TVA française à 20 % (notre calcul).
- Max : 90 € par mois (108 € TTC), cinq fois l'usage de Pro.
- Ultra : 180 € par mois (216 € TTC), pas de limite de 5 heures pour les comptes vérifiés, dix fois l'usage de Pro par semaine.

Opus est inclus dès Pro, et c'est le modèle par défaut de Claude Code sur Pro, Max et Ultra (sur Pro, depuis la version 2.1.280). Nos tests ont tourné sur un abonnement, pas sur une clé d'API. Une vidéo a consommé l'équivalent de 5 à 7 $ d'API : sur Pro, on n'a pas mesuré si elle tient dans la limite des 5 heures. À tester.

Pour installer Claude Code sur Mac ou Linux :

À COLLER

```
curl -fsSL https://claude.ai/install.sh | bash
```

Sur Windows, dans PowerShell (Claude Code tourne sous Windows directement, ou dans WSL) :

À COLLER

```
irm https://claude.ai/install.ps1 | iex
```

Les outils de la vidéo

- Node.js 22.12 ou plus récent, pour render.mjs (puppeteer-core le demande). La version LTS du jour est la 26.13.1. Nous avions la 22.22.2.
- ffmpeg. Sur Mac : `brew install ffmpeg`. Sur Linux : `sudo apt install ffmpeg`. Sur Windows, gyan.dev donne `winget install ffmpeg`. Nous avions la 8.0.1.
- Google Chrome. render.mjs pilote le Chrome déjà installé : puppeteer-core ne télécharge aucun navigateur, il lui faut le chemin du tien.
- three (la 3D) et puppeteer-core, installés dans le dossier de travail avant de lancer Claude (partie 3).
- Python 3, pour les temps des mots.

La voix

- Sur Mac, la commande say est déjà là. La voix Thomas est en français de France. Pour voir les voix installées :

À COLLER

```
say -v '?' | grep fr_FR
```

- Sur Windows et Linux (et sur Mac si tu veux mieux), Piper est gratuit et tourne hors ligne. Le projet a déménagé en octobre 2025 (OHF-Voice/piper1-gpl, licence GPL-3.0), le paquet s'appelle piper-tts (version 1.8.0 du 4 septembre 2026). Voix françaises : siwis, upmc et tom, en qualité medium. Chaque voix a sa licence : siwis est en CC BY 4.0 (crédit obligatoire), upmc en CC BY-SA 4.0, tom en AGPLv3. On ne l'a pas testé.

À COLLER

```
pip install piper-tts
python3 -m piper.download_voices fr_FR-siwis-medium
python3 -m piper -m fr_FR-siwis-medium -f voix/01.wav -- "Si le câble de votre
ascenseur casse, vous tombez de combien d'étages ?"
```

- Windows a aussi des voix françaises intégrées (Hortense, Julie) qu'on peut enregistrer en WAV par PowerShell (classe SpeechSynthesizer, méthode SetOutputToWaveFile). À tester : on n'a pas vérifié quelles voix elle voit sur une vraie machine.

Le temps de chaque mot

- Sur Mac à puce Apple : mlx-whisper, qui fait tourner Whisper sur la puce. Le setup l'attend dans un environnement Python à cet endroit, ~/.venvs/reels. Pour le créer :

```
À COLLER

python3 -m venv ~/.venvs/reels
~/.venvs/reels/bin/pip install mlx-whisper
```

- Ailleurs : faster-whisper (pip install faster-whisper), qui marche sur un processeur ordinaire (device="cpu", compute_type="int8") et n'a pas besoin de ffmpeg. L'option word_timestamps=True donne le début et la fin de chaque mot. On ne l'a pas testé. La partie 6 donne la ligne à changer dans le setup.

---

3. La marche, pas à pas

La version la plus simple, avec un CLAUDE.md

Claude Code lit le fichier CLAUDE.md du dossier où tu le lances, au début de chaque session (documentation « memory » : « Claude reads them at the start of every session »). Mets-y le setup, et il le suivra sans que tu le colles.

1. Crée un dossier vide et entre dedans.

2. Installe la 3D (three) et le pilote de Chrome (puppeteer-core) dans ce dossier.

3. Crée un fichier CLAUDE.md et colle-y le texte de SETUP.md (partie 6). Crée render.mjs à côté, avec son texte (partie 6 aussi).

4. Lance Claude Code avec les mêmes permissions que nos tests.

5. Tape ton mot, avec un point, et laisse-le travailler.

Les commandes, dans l'ordre :

```
À COLLER

mkdir reel-un-mot && cd reel-un-mot
```

```
À COLLER

npm install three puppeteer-core
```

À COLLER

```
claude --permission-mode dontAsk --allowedTools Bash Read Write Edit Glob Grep WebSearch WebFetch --disallowedTools "Bash(sudo:*)" "Bash(npm:*)" "Bash(npx:*)" "Bash(pip:*)" "Bash(pip3:*)" "Bash(brew:*)" "Bash(curl:*)" "Bash(wget:*)" "Bash(pkill:*)" "Bash(killall:*)" "Bash(kill:*)" "Bash(git:*)"
```

Puis, dans Claude Code :

À COLLER

```
Ascenseurs.
```

À la fin, le dossier contient final.mp4, angle.md (ses questions, ses sources, ce qu'il a écarté), script.md, voix.wav, words.json, la page film/index.html et sa planche de contrôle.

Sans les options de permission, Claude Code te demande l'autorisation à chaque commande, et rien n'avance pendant que tu es parti. Pendant une session, Maj+Tab fait défiler les modes de permission.

Notre commande exacte

Voici ce qu'on a lancé pour le test « Ascenseurs. », mot pour mot (les chemins sont ceux de notre Mac). Le dossier /tmp/r154-mot ne contenait que render.mjs et les modules three et puppeteer-core.

À COLLER

```
cd /tmp/r154-mot; mkdir -p voix
MAX_THINKING_TOKENS=24000 claude -p --model claude-opus-5-5 --safe-mode --permission-mode dontAsk \
  --allowedTools Bash Read Write Edit Glob Grep WebSearch WebFetch \
  --disallowedTools "Bash(sudo:*)" "Bash(npm:*)" "Bash(npx:*)" "Bash(pip:*)" "Bash(pip3:*)" "Bash(brew:*)" "Bash(curl:*)" "Bash(wget:*)" "Bash(pkill:*)" "Bash(killall:*)" "Bash(kill:*)" "Bash(git:*)" \
  --append-system-prompt-file /tmp/r154-setup/SETUP.md --output-format stream-json --verbose \
  "Ascenseurs." < /dev/null > /tmp/r154-setup/stream.jsonl 2> /tmp/r154-setup/err.txt
```

Ce que fait chaque option :

- `-p` lance Claude Code sans interface : il travaille, puis rend la main.
- `--safe-mode` coupe toutes nos personnalisations (CLAUDE.md, skills, plugins, hooks, serveurs MCP). Notre test n'a donc profité d'aucun de nos réglages. Avec la version CLAUDE.md, ne la mets pas : elle empêcherait la lecture du fichier.
- `--append-system-prompt-file SETUP.md` ajoute le setup à la consigne système de Claude. C'est l'équivalent du CLAUDE.md quand on veut un test propre.
- `--permission-mode dontAsk` refuse d'office tout ce qui demanderait une autorisation, sauf ce qui est déjà permis.

- `--allowedTools` liste ce qui est permis. `--disallowedTools` retire des commandes précises.
- `--output-format stream-json --verbose` écrit le journal de la session (chaque recherche, chaque commande, le coût). C'est de là que viennent les chiffres de la partie 4.
- `MAX_THINKING_TOKENS=24000` fixe un plafond à sa réflexion. Facultatif, on n'a pas comparé sans.

Les permissions

Pourquoi un dossier vide : Claude écrit des fichiers et lance des commandes là où il travaille. Dans un dossier vide, la seule chose qu'il peut abîmer, c'est son propre travail. N'y mets ni mot de passe, ni fichier .env, ni document personnel.

Ce qu'on a permis : lire, écrire et modifier des fichiers, lancer des commandes, chercher sur le web et lire des pages.

Ce qu'on a interdit, et pourquoi :

- sudo : aucune commande d'administrateur.
- npm, npx, pip, pip3, brew : aucune installation. Tout ce qu'il lui faut est installé avant, rien ne s'ajoute à ta machine sans toi.
- curl, wget : aucun téléchargement en dehors de ses outils web.
- kill, pkill, killall : il ne peut arrêter aucun programme. D'autres tâches tournaient sur notre Mac.
- git : il ne touche à aucun dépôt.

Une limite à connaître : d'après la documentation de Claude Code, une règle d'interdiction ne reconnaît pas le même programme appelé par son chemin complet ou à l'intérieur de sh -c. Ces interdictions sont des garde-fous, pas une cage. Si tu veux une vraie séparation, fais tourner le test dans un compte utilisateur à part ou une machine virtuelle.

---

4. Nos trois tests, en vrai

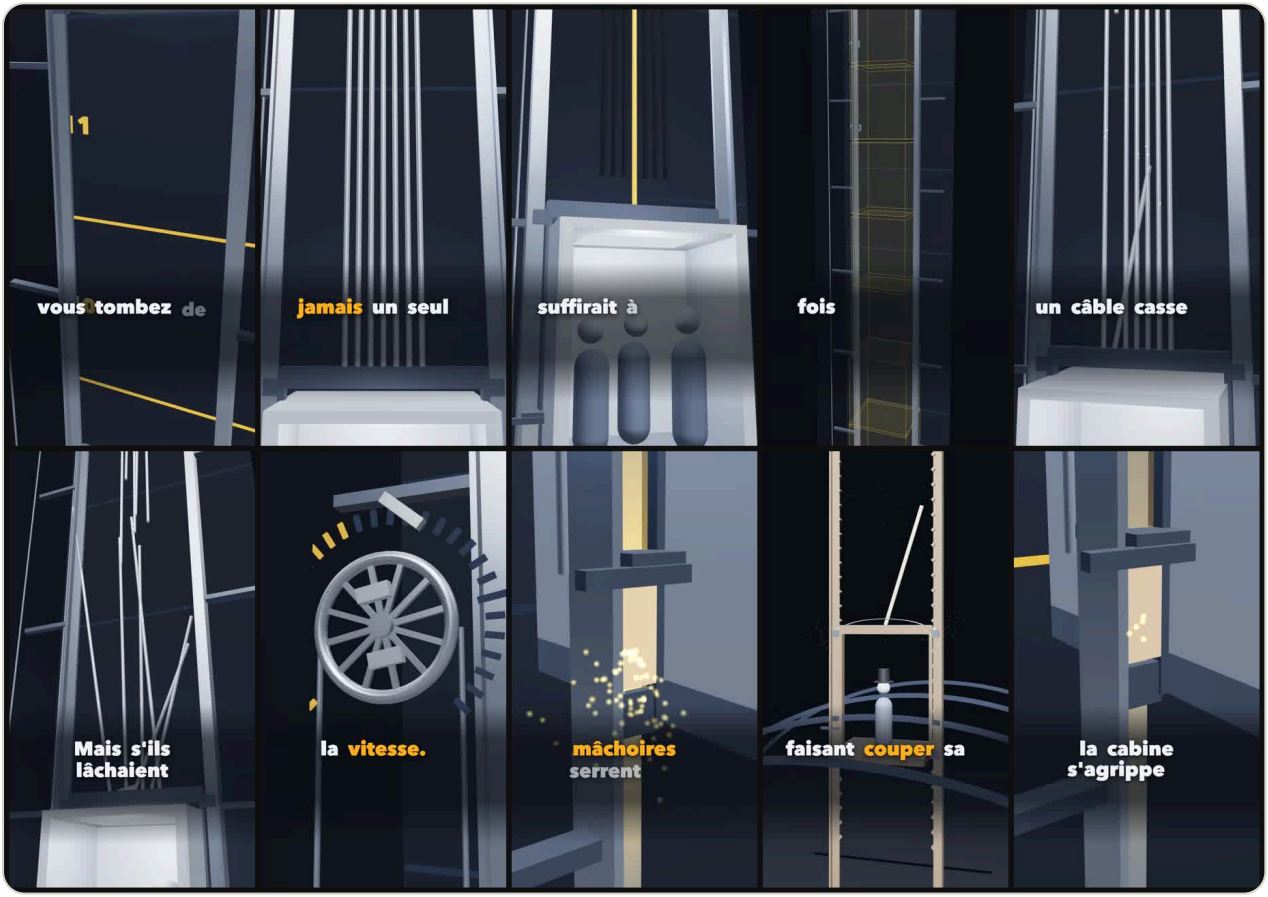
Le 21 novembre 2026, sur un Mac à puce Apple, avec la commande de la partie 3.
« Ascenseurs. » est parti à 3 h 36, les deux autres à 3 h 39, tous les trois en même temps. On n'est intervenu sur aucun. Les coûts sont ceux que Claude Code écrit dans son journal, au prix de l'API : nous ne les avons pas payés, ils sont pris sur l'abonnement.

Ascenseurs

- Durée : 47 min 50 s (de 3 h 36 à 4 h 24). Vidéo de 33,8 secondes. 10 recherches web, 9 pages demandées, dont 3 refusées (erreur 403). Coût au prix de l'API : 6,87 $.
- Ses cinq questions : le câble qui casse, le bouton « fermer la porte », la limite des câbles dans les très hautes tours, les miroirs, le saut au dernier moment. L'angle choisi, à 3 h 41, 4 min 44 s après le départ : « Si le câble de votre ascenseur casse, vous tombez de combien d'étages ? » Sa réponse : aucun. Il y a plusieurs câbles, un limiteur de vitesse et un parachute, des mâchoires qui serrent les rails. Preuve à l'appui : Chicago en 2018, et Otis en 1854.

- Ce qu'il a écarté : « l'ascenseur de Chicago a chuté de quatre-vingt-quatre étages » (les premiers articles le disaient, le rapport final le dément) ; la phrase d'Otis « All safe, gentlemen, all safe » (aucune source d'époque trouvée) ; « Otis a inventé l'ascenseur » (faux, il a inventé le frein) ; la hache ou le couteau (les sources se contredisent, son script dit « faire couper ») ; l'explication du « coussin d'air » de la chute de 1945 dans l'Empire State Building.

Ce qu'on a revérifié derrière lui. Chicago tient : le 16 novembre 2018, un des sept câbles de l'ascenseur n° 2 du 875 North Michigan Avenue casse vers le 20e étage, la cabine est abaissée sur neuf étages, six passagers, aucun blessé, et le frein de secours ne s'est jamais déclenché. Mais le rapport qu'il présente comme une inspection est celui d'Otis, l'entreprise qui entretient l'ascenseur, repris par la Ville de Chicago. Deux phrases de son film vont plus loin que les sources. « Ensemble, ils tiennent douze fois son poids » : douze est le minimum européen avec trois câbles ou plus, et aux États-Unis le minimum va de 7,6 à 11,9 selon la vitesse. « Réponse, aucun » : vrai pour un ascenseur aux normes et entretenu, et Otis lui-même pose cette condition.



Le film « Ascenseurs » de Claude, une image par réplique : la cabine qui descend, les sept câbles, un seul qui porte la cabine, les douze cabines du coefficient, le câble qui casse à Chicago, les câbles qui lâchent, la poulie du limiteur, les mâchoires sur le rail, Otis en 1854, la cabine agrippée (test du 21 novembre 2026)

Agrafeuses

- Durée : 32 min 42 s. Vidéo de 32,3 secondes. 9 recherches web, 7 pages demandées, dont 2 refusées (erreur 403). Coût au prix de l'API : 5,36 $.
- L'angle choisi : « Pourquoi la petite plaque de métal sous votre agrafeuse peut-elle tourner ? » C'est l'enclume. Un demi-tour inverse le pliage des pattes : vers l'intérieur, l'agrafe tient pour de bon ; vers l'extérieur, elle est provisoire et s'enlève facilement.
- Ce qu'il a écarté : la « première agrafeuse » faite pour Louis XV (aucun document d'époque, les articles se recopient), l'agrafeuse rouge du film « Office Space » (vraie, mais les sources

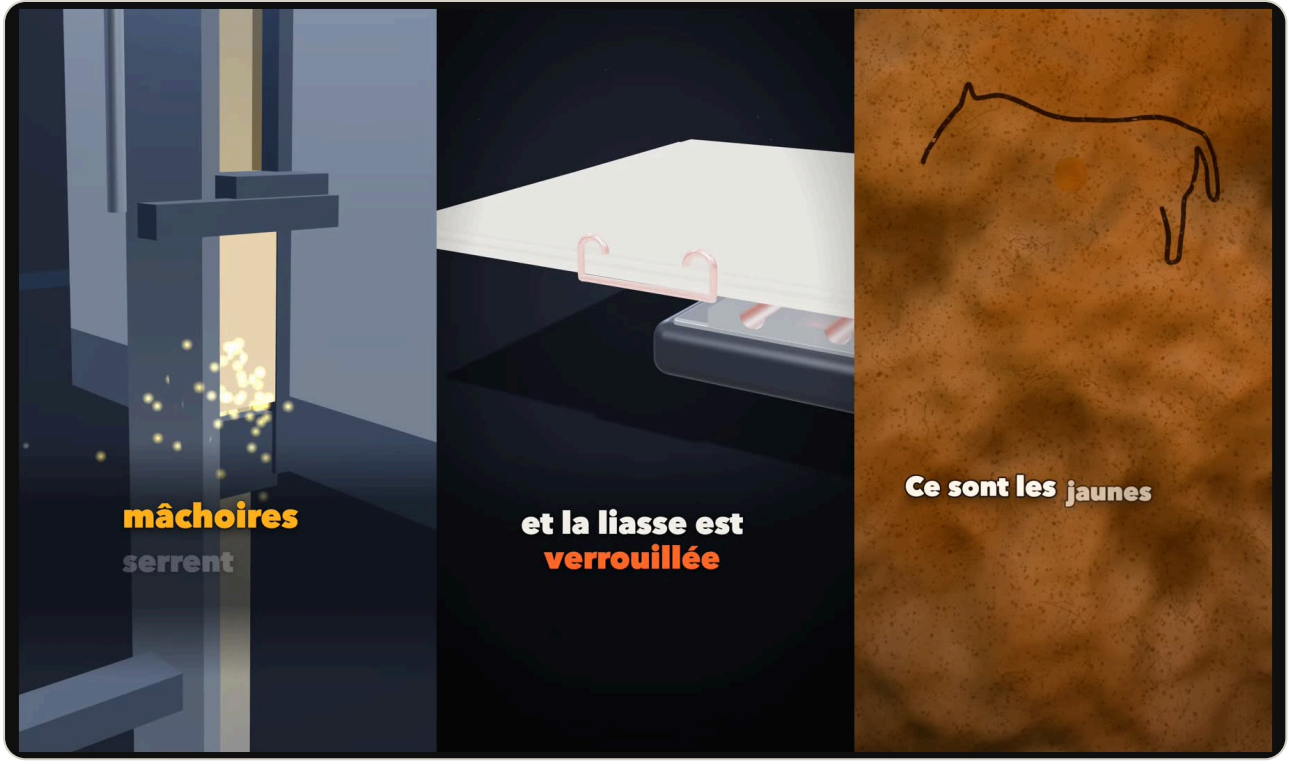
se contredisent sur la date et le modèle), « Swingline invente le chargement par le dessus en 1937 » (une seule source sérieuse), et « toutes les enclumes tournent » (faux).

Ce qu'on a revérifié : l'enclume qui tourne pour des agrafes provisoires est bien sur la fiche Swingline du modèle 747 (« Anvil rotates for pinning ability to create temporary staples »). Une phrase est à nuancer : « ce mode provisoire figure déjà dans un brevet de 1929 ». Le brevet de 1929 a bien une enclume qui tourne pour une agrafe plus facile à retirer, mais elle plie une patte vers l'intérieur et l'autre vers l'extérieur. Les deux pattes vers l'extérieur n'y sont qu'une variante possible : le réglage complet est décrit dans un brevet déposé en 1938, avec une enclume qui coulisse.

Peinture beige

- Durée : 41 min 13 s. Vidéo de 31,3 secondes. 17 recherches web, 7 pages demandées. Coût au prix de l'API : 7,00 $.
- L'angle choisi : « Avec quoi colore-t-on une peinture beige ? » Sa réponse : un pot de blanc et à peine une cuillère de colorant pour près de quatre litres. Ce colorant est surtout de l'oxyde de fer, goethite pour le jaune et hématite pour le rouge, les deux minéraux des jaunes et des rouges de Lascaux.
- Ce qu'il a écarté : les « deux mille beiges refusés par Steve Jobs » (le nuancier Pantone comptait alors moins de mille couleurs en tout), l'étude Zillow sur le salon beige qui ferait vendre plus cher (Zillow la donne lui-même avec réserve), l'origine du « beige des propriétaires » (aucune source sérieuse), et « Lascaux a dix-sept mille ans » (révisé à environ vingt et un mille ans en 2019).

Ce qu'on a revérifié : Lascaux tient (ministère de la Culture : « the red are hematites and the yellows are goethites » ; occupation entre 21 500 et 21 000 ans avant le présent). La cuillère tient pour la formule qu'il cite, une fiche de teinte d'une peinture d'étanchéité pour maçonnerie (27/48 d'once par gallon, environ 16,6 mL pour 3,8 L). Il l'a signalé lui-même : une seule formule, et pas celle d'une peinture murale. « Votre mur beige est peint avec la palette de Lascaux » est une image : mêmes minéraux, fabriqués en usine.



Les trois films, tels que Claude les a rendus : « Ascenseurs. », « Agrafeuses. » et « Peinture beige. » (tests du 21 novembre 2026)

Ce qui ne va pas encore

- La voix. say et la voix Thomas sonnent robotiques, et la licence de macOS en interdit la publication (partie 5). Sur la peinture beige, Whisper a entendu « gautite » pour « goethite » : la prononciation est à vérifier à l'oreille, et Claude l'a dit lui-même.
- Le temps. Entre 33 et 48 minutes pour une trentaine de secondes. Le rendu seul a pris 4 minutes pour 940 images (avec trois tests en même temps sur la machine), parce que Chrome fait une capture par image, et Claude le refait après chaque correction.
- Les sources. Sur « Ascenseurs », il a cité deux articles d'Elevator World qu'il n'avait pas pu ouvrir (erreur 403), et un lien qui donne un coefficient de 8 là où il écrit 12 (le 12 est sur la page voisine du même code). Ses faits tiennent quand même, mais « deux sources par fait » reste une consigne, pas une garantie.
- Les phrases trop fortes. « Douze fois son poids », « Réponse, aucun », « dès 1929 » : chaque fois, un détail que la source ne dit pas tout à fait.
- Les images. Tout est dessiné en code : les personnes sont des capsules, les objets des formes simples. Ça se tient, mais ça reste schématique.
- Il ne s'arrête pas net. Sur « Ascenseurs », il a lancé ses rendus en tâche de fond, et la session a rendu la main trois fois avant d'avoir vraiment fini. Au dernier rendu, Chrome a planté entre l'image 750 et l'image 900 : il a gardé pour la suite les images du rendu précédent, en expliquant pourquoi elles étaient identiques. Il l'a écrit dans son compte rendu. Vérifie quand même à l'œil.

---

5. Les limites et les précautions

Relire les faits avant de publier

- Ouvre chaque lien d'angle.md et vérifie que la page dit bien ce qu'il cite, au chiffre près.
- Demande-lui quelles pages il a vraiment lues en entier, et lesquelles il n'a vues qu'en extrait de recherche ou n'a pas pu ouvrir.
- Repère les mots absolus (jamais, aucun, toujours, tous) et cherche la source qui les porte. C'est là que nos trois films ont forcé.
- Un chiffre sans source et sans date sort du script, ou devient « à vérifier ».

Les droits

- La voix du Mac : la licence de macOS Tahoe 26 (section 2.F) permet les voix système pour des créations « personal, non-commercial » et exclut leur publication « in a profit, non-profit, public sharing or commercial context ». La voix Thomas sert de maquette, pas de voix publiée. Pour publier : ta propre voix, une voix Piper dont la licence le permet (avec le crédit demandé), ou un service de synthèse qui accorde un usage commercial.
- Les images : le setup dessine tout en code dans la page, il n'y a rien à créditer. S'il ajoute une photo, un logo ou une capture, demande-lui d'où elle vient et sous quelle licence.
- Les polices : celles de Google Fonts sont sous licence OFL. Leur usage commercial est permis, et la licence ne s'applique pas aux documents créés avec elles : rien à mentionner sous ta vidéo.
- La musique : le setup n'en met pas. Si tu en ajoutes dans Instagram, la Sound Collection de Meta (plus de 14 000 titres) est libre de droits et utilisable à des fins commerciales. La bibliothèque musicale sous licence est prévue pour un usage personnel non commercial, et certains comptes professionnels n'y ont pas accès.

La mention « contenu généré par IA »

- Instagram demande d'étiqueter, avant de publier, une vidéo photoréaliste ou un son réaliste créés ou modifiés par IA, et cite l'exemple d'un reel raconté par une voix off d'IA réaliste. Une vidéo « dans un style qui ressemble à un dessin animé » n'a pas à l'être. Sans étiquette quand elle est due, Instagram prévoit des sanctions, sans dire lesquelles. Avec une 3D schématique et la voix du Mac, l'étiquette n'est pas clairement exigée. Avec une voix de synthèse réaliste, elle l'est. Dans le doute, active-la : c'est un interrupteur à cocher au moment de publier (« Add AI label » dans l'aide d'Instagram en anglais).
- En Europe, l'article 50 du règlement sur l'IA s'applique depuis le 2 août 2026. Pour un usage professionnel, il impose de signaler les hypertrucages (un contenu qui ressemble à des personnes, des lieux ou des objets réels et pourrait passer pour authentique), ainsi que les textes générés publiés pour informer le public sur des questions d'intérêt public, sauf s'ils ont été relus par une personne qui en assume la responsabilité éditoriale. Des capsules en 3D ne risquent pas de passer pour authentiques. Pour le texte, l'exception joue si quelqu'un l'a relu et l'assume : une raison de plus de suivre la liste plus haut. Ce n'est pas un avis juridique.

Sur ta machine

- Lance-le toujours dans un dossier neuf. Regarde ce qu'il contient avant de le copier ailleurs.
- Ne lui donne aucune clé ni aucun mot de passe : il n'en a pas besoin.
- S'il te faut un outil en plus, installe-le toi-même, avant ou après, jamais en lui rendant npm ou pip.

6. Le texte du setup

Copie chaque fichier tel quel. SETUP.md va dans CLAUDE.md pour la version simple, ou reste SETUP.md avec notre commande.

SETUP.md

À COLLER

```
# Setup : un mot, un reel

Tu reçois un seul mot : c'est le sujet. Tu fabriques seul, sans poser de
question ni attendre de validation, un reel vertical en français (1080 × 1920,
25 à 35 secondes) qui donne envie de regarder jusqu'au bout. Tu travailles dans
le dossier courant.

## 1. L'angle d'abord
- Cherche sur le web (WebSearch, WebFetch) ce que les gens ignorent sur ce
sujet. Écris cinq angles possibles, chacun sous forme de question.
- Garde la question dont on veut le plus connaître la réponse, et dont la
réponse est vérifiable et surprenante.
- Vérifie chaque fait sur deux sources sérieuses (fabricant, norme, organisme
public, presse établie). Une anecdote célèbre qui ne tient pas se retire, et tu
écris pourquoi.
- `angle.md` : les cinq questions, celle retenue et pourquoi, les faits avec
leurs sources, ce que tu as écarté et pourquoi.

## 2. Le script
- `script.md` : sept à dix répliques courtes. La première pose la question, d'un
trait. Chaque réplique fait avancer la réponse (mais, donc), jamais une liste de
faits. La dernière répond et laisse une image en tête.
- Nombres écrits en lettres, pas de deux-points, pas de sigle épelé.

## 3. La voix
- Une prise par réplique : `say -v Thomas -r 185 -o voix/NN.aiff "texte"`, puis
un seul `voix.wav` (ffmpeg, 48 kHz, 0,25 s entre deux répliques).
- Temps de chaque mot : `~/.venvs/reels/bin/python` avec `mlx_whisper`
(`path_or_hf_repo="mlx-community/whisper-large-v3-turbo"`,
`word_timestamps=True`, `language="fr"`), recalé sur le texte exact du script →
`words.json` (mot, début, fin, réplique).

## 4. Les scènes
- Une seule page `film/index.html`, 1080 × 1920, qui expose `window.seek(t)` :
elle dessine l'état exact de la seconde t (aucune horloge interne, aucun hasard
sans graine). three.js est là : `<script type="importmap">{"imports":
{"three":"/node_modules/three/build/three.module.js","three/addons/":"/node_modu
les/three/examples/jsm/"}}</script>`. La 3D en CSS marche aussi.
- Une idée visuelle par réplique, en 3D, qui montre ce que dit la voix. Chaque
scène naît d'un objet de la précédente (l'objet se transforme, la caméra passe à
travers) : ni coupe sèche, ni fondu.
- Le mouvement : jamais linéaire (départ et arrivée amortis, léger dépassement),
les éléments entrent l'un après l'autre, la caméra ne s'arrête jamais tout à
fait, quelque chose change toutes les demi-secondes.
- Les sous-titres : chaque mot apparaît au moment où il est dit (`words.json`),
gros et lisible, le mot clé de chaque réplique en couleur. Rien d'important dans
les 250 px du bas ni dans les 130 px de droite.
- Une palette resserrée, une seule couleur d'accent, une police lisible.

## 5. Le rendu
- `node render.mjs film/index.html <durée> frames 30` (Chrome sans écran, une
image tous les trentièmes de seconde), puis
  `ffmpeg -framerate 30 -i frames/f%05d.jpg -i voix.wav -c:v libx264 -pix_fmt
yuv420p -crf 18 -c:a aac -b:a 192k -shortest final.mp4`.

## 6. Le contrôle
- Planche : `ffmpeg -i final.mp4 -vf fps=2,scale=270:-1,tile=8x8 planche.jpg`,
regarde-la (outil Read) : image vide, texte illisible ou coupé, scène qui ne
```

```
montre pas ce que dit la voix, moment où rien ne bouge. Corrige et refais le
rendu, deux tours au plus.
- Termine par un compte rendu de cinq lignes : l'angle, les faits écartés, la
durée, ce que tu as corrigé.
```

render.mjs

À COLLER

```
// Rend une page film/index.html image par image : la page expose window.seek(t)
(t en secondes).
// usage : node render.mjs <page.html> <durée en s> <dossier d'images> [fps=30]
import puppeteer from "puppeteer-core";
import http from "node:http";
import { readFile, mkdir } from "node:fs/promises";
import { resolve, extname, join, relative, dirname } from "node:path";
const [page_, dur, outDir, fpsArg] = process.argv.slice(2);
const fps = Number(fpsArg || 30), root = process.cwd();
const types = { ".html": "text/html", ".js": "text/javascript", ".mjs":
"text/javascript", ".css": "text/css", ".json": "application/json", ".png":
"image/png", ".jpg": "image/jpeg", ".svg": "image/svg+xml", ".woff2":
"font/woff2", ".ttf": "font/ttf", ".glb": "model/gltf-binary", ".wav":
"audio/wav", ".mp3": "audio/mpeg" };
const srv = http.createServer(async (req, res) => {
  try { const p = join(root, decodeURIComponent(req.url.split("?")[0])); const
buf = await readFile(p); res.writeHead(200, { "Content-Type": types[extname(p)]
|| "application/octet-stream" }); res.end(buf); }
  catch { res.writeHead(404); res.end(); }
}).listen(0);
const port = srv.address().port;
await mkdir(outDir, { recursive: true });
const browser = await puppeteer.launch({ executablePath: "/Applications/Google
Chrome.app/Contents/MacOS/Google Chrome", headless: true, args: ["--ignore-gpu-
blocklist", "--enable-webgl", "--use-angle=metal", "--hide-scrollbars", "--
force-color-profile=srgb"] });
const page = await browser.newPage();
await page.setViewport({ width: 1080, height: 1920, deviceScaleFactor: 1 });
page.on("pageerror", e => console.error("ERREUR PAGE :", e.message));
page.on("console", m => { if (m.type() === "error") console.error("console :",
m.text()); });
await page.goto(`http://127.0.0.1:${port}/${relative(root, resolve(page_))}`, {
waitUntil: "networkidle0" });
await page.waitForFunction("typeof window.seek === 'function'", { timeout: 20000
});
await page.evaluate("window.ready ? window.ready : true");
const n = Math.round(Number(dur) * fps);
for (let i = 0; i < n; i++) {
  await page.evaluate(t => window.seek(t), i / fps);
  await page.screenshot({ path: join(outDir, `f${String(i).padStart(5,
"0")}.jpg`), type: "jpeg", quality: 92 });
  if (i % 150 === 0) console.log(`${i}/${n}`);
}
await browser.close(); srv.close();
console.log(`fini : ${n} images dans ${outDir}`);
```

Si tu n'es pas sur Mac

Rien de ce qui suit n'a été testé par nous. Dans SETUP.md, remplace les deux lignes de la section « 3. La voix » par :

À COLLER

```
- Une prise par réplique : `python3 -m piper -m fr_FR-siwis-medium -f
voix/NN.wav -- "texte"`, puis un seul `voix.wav` (ffmpeg, 48 kHz, 0,25 s entre
deux répliques).
- Temps de chaque mot : `python3` avec `faster_whisper` (`WhisperModel("small",
device="cpu", compute_type="int8")`, `word_timestamps=True`, `language="fr"`),
recalé sur le texte exact du script → `words.json` (mot, début, fin, réplique).
```

Dans render.mjs, change le chemin de Chrome (executablePath) pour celui de ta machine, en général C:\Program Files\Google\Chrome\Application\chrome.exe sur Windows ou /usr/bin/google-chrome sur Linux, et retire l'option --use-angle=metal, propre au Mac.

Sources

- Claude et Claude Code : les tarifs de Claude en France, l'installation et les comptes acceptés, le choix du modèle, CLAUDE.md, les options de la ligne de commande, les modes de permission, les règles de permission.
- Outils : Node.js, ffmpeg, puppeteer-core, three.js, Piper, les voix de Piper, mlx-whisper, faster-whisper, SetOutputToWaveFile, Microsoft.
- Droits et étiquettes : la licence de macOS Tahoe 26, l'étiquette IA sur Instagram, la musique sur Instagram, la FAQ de Google Fonts, l'article 50 du règlement sur l'IA.
- Ascenseurs : Otis, la sécurité des ascenseurs de grande hauteur, ABC7 Chicago, 19 novembre 2018, ABC7 Chicago, 23 avril 2019, le rapport d'Otis du 31 janvier 2019, le règlement de Californie sur les câbles, ASME sur Elisha Otis, American Heritage, 1978, D. A. Cooper, A History of the Lift Safety Gear.
- Agrafeuses : Swingline 747, le brevet US 1 898 678, le brevet US 2 181 904.
- Peinture beige : la fiche de teinte UGL DRYLOK, les pigments de Lascaux, la datation de Lascaux.