TOGINO / MOTION LP PLAYBOOK

AIとつくる、動くLP。制作から公開・運用まで。

架空の包丁研ぎ直しサービス「TOGINO」を題材に、動くLPを一から作った記録です。キャラクターの6ポーズ、スクロール演出、導入動画のつなぎ方まで、実際に使った仕組みと、別のLPでも使える手順をまとめました。

CHAPTER 01

AIとつくる、動くLP

静止画・キャラクター・動画を、スクロールでつながる体験へ。制作から公開・運用までの再現ガイドです。



TOGINOの導入ムービーの1コマ。砥石のキャラクター「トギまる」が、サビた包丁を磨き上げます。

このLPについて

TOGINO（トギノ）は、このサンプルのために考えた架空の「包丁研ぎ直し・宅配サービス」です。商品名、コピー、キャラクター、画像、動画はすべて今回新しく作ったもので、特定の企業や案件の情報は含みません。そのまま見本として共有できます。

この資料でできるようになること

- 動かす対象ごとに素材を分ける
- 自然言語で動きを具体的に指示する
- 動画と本文をつなぐ
- キャラクターをページ内で移動させる
- 公開後の確認と費用管理を行う

読む順番

まずは仕組みと素材設計（2〜5章）。キャラクターと動画は6〜10章。制作手順・公開・品質確認・費用は11〜16章を参照してください。

対象：AIを使ってLPを制作したい企画担当者・制作者。AIへの指示で進められますが、素材の良否・実機確認・公開内容の判断は人が担当します。

CHAPTER 02

動いているのは、Webの部品

AIが素材とコードの制作を助け、閲覧時はブラウザがアニメーションを実行します。

01　素材を用意する

背景、包丁、キャラクター、動画、文字を別々に準備します。1枚の完成画像をそのまま動かすだけでは、包丁の後ろにいたキャラクターを自然に移動できません。

02　ページを組み立てる

HTMLが文章・ボタン・動画などの構造を持ち、CSSが位置・サイズ・色・重なりを決めます。JavaScriptがスクロール位置や表示状態を読み取り、CSSへ進行度を渡します。

03　ブラウザで再生する

拡大・移動・透明度・ぼかし・画像切り替えを組み合わせます。このLPは通常のHTML／CSS／JavaScriptだけで実装し、React、GSAP、WebGLや3Dの骨格アニメーションは使っていません。外部ライブラリの読み込みもありません。

制作中と公開後の費用は別

公開済みの画像・動画を表示するだけなら、訪問者が見るたびに画像生成AIを呼び出す必要はありません。公開後に発生するのは主にホスティング、通信、ドメイン、保守などです。AI生成の費用は、素材を作成・再生成するときにだけ発生します。

3つの動きを使い分ける

- 時間連動：呼吸のような揺れ、まばたき、動画再生。
- スクロール連動：拡大、背景の速度差、本文への切り替え、研ぎ直し前後の比較。
- イベント連動：画面への登場、ジャンプ、比較スライダーの操作。

CHAPTER 03

使用ツールと、必要なスキル

ツールは制作手段。スキルは、繰り返せる制作・確認手順です。

Claude Code：制作を進める窓口

構成案、自然言語からのコード実装、修正、ローカル表示、ブラウザでの確認を担当しました。依頼者は「どこで・何が・どう動き・どこに止まるか」を伝え、画面を見てフィードバックします。

画像生成：背景・商品・キャラクター素材

OpenAIの画像生成API（モデル gpt-image-2.6-sunburst、品質 high）で、商品の切り抜き、透過キャラクター、追加ポーズ、工程シーン、動画用の静止画を作りました。採用したのは17枚です。2枚目以降は、最初に作った包丁とキャラクターの画像を「参照画像」として渡し、見た目をそろえています。

動画：生成AIではなく、静止画3枚をコードで動かす

導入ムービーは動画生成AIを使っていません。「サビた状態」「磨き上げた状態」「引きの決めカット」の静止画3枚をHTML上で重ね、クロスフェード・ゆっくりした寄り引き・光の筋・きらめきを付け、1コマずつ撮影して動画ファイルにしました。作り方は9章にあります。

実装・検証・公開

HTML／CSS／JavaScript、ブラウザ、Chromeを自動操作して撮影するPlaywright、動画変換のffmpeg、画像圧縮のcwebp、版管理のGitを使いました。映像の仕様確認にはffprobeが役立ちます。

再利用する制作スキルの中身

動くLPの設計、レイヤー素材の制作、画像生成、ブラウザ検証、成果物の品質確認をひとつの手順にします。スキル名だけで再現できるわけではなく、入力素材・手順・合格条件をセットにします。

CHAPTER 04

11シーンを、一本の流れにする

派手な動きの数より、「次に何を見てほしいか」を先に決めます。

TOGINOの構成

導入動画 → ① メイン訴求 → ② 受付案内 → ③ 研ぎ直し比較 → ④ 対象商品 → ⑤ 職人の仕事 → ⑥ 細部の比較 → ⑦ 料金 → ⑧ 最短48時間の案内 → ⑨ 利用の流れ → ⑩ FAQ → ⑪ 登録への誘導。

演出を割り当てる

- 導入：動画を全面表示し、言葉を重ねて世界観へ。
- 商品紹介：三徳包丁・ペティナイフ・キッチンばさみに段差と速度差。
- 研ぎ直し比較：同じ位置のBefore／Afterをマスクで切り替え。
- 職人・細部：写真を大きく見せ、刃先・刃元へズーム。
- 料金・手順：読みやすさを優先し、軽い登場演出。
- 各章の移動：キャラクターが左右へジャンプして案内。

最初の試作は3シーンで十分

「導入」「商品の価値が伝わる比較」「問い合わせ」の3シーンで、色・文字・動きの速度・スマホでの見え方を固めます。その後に同じ設計ルールで残りを展開すると、作り直しを減らせます。

寂しい余白を埋める方法

写真を端まで広げる、背景に薄い工房の質感を入れる、大きな輪郭文字や帯を置く、商品サイズに強弱をつける。すべてを動かさず、主役・補助・静止の3段階で密度を調整します。本文やCTAにキャラクターが重ならない位置を確保します。

CHAPTER 05

商品とキャラクターを、動かせる素材へ

一枚絵ではなく、動かす単位ごとに画像を分けて作ります。



研ぎ直し前の包丁。研ぎ直し後の画像と、位置・大きさ・角度がそろっています。

01　基準になる画像を先に作る

最初に作ったのは2枚だけです。背景透過の「研ぎ上げた三徳包丁」と、キャラクター「トギまる」の正面立ち。この2枚を以降のすべての生成で参照画像として渡し、別の絵でも同じ包丁・同じキャラクターに見えるようにしました。

02　背景・商品・キャラクターを分ける

工房の背景は人物も文字もない状態で用意。商品とキャラクターは背景透過で、全体が欠けずに写るよう余白を確保します。文字・ボタンはすべてHTMLで書き、画像には焼き込みません。

03　Before／Afterは「後」から作る

先に完成状態（After）を作り、それを参照画像にして「位置・角度・大きさ・輪郭はそのまま、状態だけをサビと刃こぼれのある姿に変える」と指示してBeforeを作りました。2枚がぴったり重なるので、境界線を動かすだけで研ぎ直しの比較になります。

04　Webに向けて整理

生成したPNG（制作マスター）はそのまま保存し、配信用はWebPに圧縮しました。マスター合計は約34MB、配信用の画像・動画は合計約11MBです。

検品で見るところ

透明に見える背景が本当に透明か、輪郭に白い縁がないか、手足が崩れていないか、キャラの大きさがポーズ間で変わらないか、意図しない文字やロゴが入っていないかを確認します。今回は梱包シーンの1枚で画像の一部が白く抜けたため、「画面全体を不透明の写真にする」と指示を足して作り直しました。AIで作ったイメージを実際の施工実績として扱わないことも重要です。

CHAPTER 06

6ポーズで「生きている」動きへ

今回の方式は、透過画像を切り替えながら移動する2Dアニメーションです。



3列×2行のスプライトシート。上段は右向き、下段は左向き。「しゃがむ・空中・着地」の3ポーズです。

待機 → 予備動作 → ジャンプ → 着地 → 案内

待機時は微かな揺れやまばたき。移動の直前にしゃがみ、空中ポーズで弧を描いて移動し、着地時に沈み込みます。その後は手を振るポーズへ。向き別の絵を使うことで、単なる左右反転でハチマキの結び目が逆になることを避けます。

今回の実装値

移動は約1.3秒。しゃがみ・空中・着地のタイミングで画像を切り替えます。PCとスマホで移動距離とジャンプの高さを変え、次の目的地は最新の1件だけ保持。高速スクロールで古い移動が延々と続くことを防ぎます。

生成したシートに合わせて調整したこと

生成されたシートは、下段（左向き）の並びが「着地・空中・しゃがむ」と上段の逆になっていました。作り直さず、CSS側で左向きのときだけ列の対応を入れ替えています。また、立ち姿の画像とシー

ト内のキャラの大きさが違ったため、シートの表示倍率を合わせて、切り替えの瞬間に大きさが変わらないようにしました。

6枚は連続動画ではないため、複雑な歩行・表情・口パクには追加素材や別方式が必要です。まずは少ないポーズの位置とタイミングをそろえる方が、自然に見えやすくなります。

CHAPTER 07

スクロールとジャンプの仕組み

ページの進行度を0〜1に変換すると、複数の動きを同じ基準で制御できます。

画面が止まって見える仕組み

長いセクションの中に position: sticky の表示領域を置きます。ページ自体はスクロールしていますが、表示領域が一定位置に留まり、その間に画像・文字・背景の状態が変わります。stickyの親の高さやoverflow指定によっては固定が効かないため確認します。

進行度から見た目を決める：概念コード

```
const clamp = v => Math.max(0, Math.min(1, v));
const progress = clamp(-sectionTop /
  Math.max(1, sectionHeight - stickyHeight));
// 例：進行度に応じて1倍から1.25倍へ拡大
const scale = 1 + progress * 0.25;
```

ジャンプの軌道：概念コード

```
const ease = p * p * (3 - 2 * p);
const x = startX + (endX - startX) * ease;
const y = startY + (endY - startY) * ease
  - Math.sin(Math.PI * p) * jumpHeight;
// pは飛行区間だけを0〜1に正規化した値
```

移動方向で上段／下段の画像を選び、時間帯で列を切り替えます。シートのCSSは background-size: 300% 200%。列は0／50／100%、行は0／100%のbackground-positionで選択できます。

滑らかに動かすための約束

requestAnimationFrameで描画をまとめ、主にtransformとopacityを変更します。画面外はIntersectionObserverで処理を抑え、毎回のスクロールイベントで重い計算を繰り返さないようにします。上記コードは仕組みの抜粋で、単体で完成LPにはなりません。

CHAPTER 08

動画を、冒頭の体験につなぐ

映像の再生と、スクロールによる画面演出を分けて扱います。

実際のつなぎ方

動画をobject-fit: coverで全面表示 → メッセージを浮かび上がらせる → スクロールで動画を最大約1.25倍に拡大 → ぼかしを強める → 次のヒーローへクロスフェード。動画の再生時間をスクロールで巻き戻す方式ではありません。

枠をなくす代わりに、切れる範囲を設計

全面表示では画面比率に合わせて映像の上下または左右が切れます。そのため動画用の静止画は「重要なものを画面の中央〜やや右に置く」と指示して作り、スマホでは表示位置を右寄りにしてキャラクターと刃が見えるようにしました。重要な文字は動画に焼き込まず、HTML側で重ねています。

止まっても本文が読める設計

muted／playsinline、静止画ポスター、再生ボタンを用意。自動再生が拒否されても導線は止めません。画面外や非表示タブでは再生を抑え、「動きを止める」ボタンと、動きを減らすOS設定の両方に対応します。

CHAPTER 09

導入ムービーの制作手順

制作記録：2026年11月20日。動画生成AIは使わず、静止画3枚から作りました。

01　開始画像と終了画像を作る

大きな包丁の上でキャラクターが布で刃を磨いている構図を、サビた状態で1枚生成。次にその画像を参照に、構図・カメラ・キャラクターの位置は変えず、刃の状態だけを磨き上げた姿にした終了画像を作りました。

02　決めカットを作る

研ぎ上げた包丁を砥石に置き、横でキャラクターが親指を立てている引きの1枚を生成。締めの画として使います。

03　HTMLの上で動かす

3枚を重ねた1920×1080のページを作り、「何秒のときに、どの画像を何%見せ、どれだけ拡大するか」を1つの関数にまとめました。0〜5秒でサビが消えて輝きへ変わり、途中で光の筋が横切り、5.6秒から決めカットへ切り替わり、7.3秒からきらめきが灯ります。

04　1コマずつ撮影して動画にする

Chromeを自動操作して時刻を1/30秒ずつ進め、360コマを撮影。ffmpegでH.264の動画にまとめました。同じ手順を実行すれば、いつでも同じ動画を作り直せます。

できあがった動画

PC用：12.0秒、1920×1080、30fps、H.264、約5.5MB。スマホ用：1280×720、約1.8MB。ポスター画像は0.4秒時点のコマです。音声はありません。

この方式の限界：人物やキャラクターの手足そのものは動きません。動いて見えるのは、状態の変化・カメラの寄り引き・光と粒子です。手首を動かすような芝居が必要な場合は、同じ開始・終了画像を動画生成AIに渡す方法が候補になります（今回は未実施）。

CHAPTER 10

絵柄を安定させた指示

「何を変え、何を保つか」を分けます。以下は今回実際に使ったプロンプトの抜粋です。全文は制作フォルダの src/prompts/images.json に保存しています。

研ぎ直し前：状態だけを変える

```
Reproduce the reference image exactly and pixel-aligned: the same knife at the same position, angle, size and framing, same outline. Change ONLY its condition to a neglected, worn knife: the blade is dull and grey with patches of orange-brown rust, dark tarnish, water stains and fine scratches, the mirror shine is gone, and there are a few small chips along the cutting edge near the tip and the middle.
```

動画の終了画像：刃だけを磨き上げる

```
Reproduce the reference image exactly and pixel-aligned: same camera, framing, lighting, whetstone, workbench, background and the same mascot character in the same kneeling pose at the same position holding the same white cloth. Change ONLY the knife blade's condition: all rust, tarnish and stains are gone and the whole blade is now freshly sharpened, clean, mirror-bright polished stainless steel.
```

試作で起きた問題 → 修正の考え方

- 別の絵でキャラや包丁の見た目が変わる → 基準画像を参照として渡し、特徴を文章でも書く。
- Before／Afterがずれる → 「同じ位置・角度・大きさ・輪郭」を明記し、変えるのは状態だけにする。
- 画像の一部が白く抜ける → 「画面全体を不透明の写真に」と明記して作り直す。
- ロゴや文字が勝手に入る → 文字・刻印・ロゴなしを毎回指定し、文字はHTMLで足す。
- シートの並びが指示と違う → 作り直す前に、CSS側の対応で吸収できないか検討する。

別案件では素材に合わせて書き換えます。禁止事項を増やすだけでなく、主語・動作対象・変化する範囲を具体化するのがポイントです。同じ指示でも毎回同じ結果になる保証はありません。

CHAPTER 11

自然言語で制作を進める方法

一度に「全部かっこよく」ではなく、確認できる単位に分けます。

最初の依頼テンプレート

「添付素材をもとに、［商品・サービス］の動くLPを制作してください。目的は［問い合わせ／購入］、対象は［顧客］です。まず導入・比較・CTAの3シーンを作成。［基調色］を基調に、人物・商品・背景・文章を分離してください。スマホ優先で、動きを止めても全情報が読めるようにし、公開前に未接続のリンクを一覧化してください。」

キャラ画像の依頼テンプレート

「参照キャラの形・色・ハチマキ・手足を保ち、背景透過で全身を作成。右向きのしゃがみ／空中／着地、左向きの同3ポーズを、同じ縮尺・足元基準で3列×2行に配置。各セルに余白を確保し、枠線・番号・文字は入れないでください。」

動きの修正テンプレート

「職人セクションに入ったとき、右下のキャラが左下へ1.3秒程度でジャンプ。しゃがみ→空中→着地の順に切り替え、着地後に一度手を振ってください。本文とCTAには重ねず、スマホは高さを控えめに。高速スクロール時は最新の目的地だけを採用してください。」

フィードバックは場所・症状・期待で伝える

例：「スマホの料金セクションで、キャラが金額に重なる。左下へ移し、本文を読む間は止めたい」。画面幅、セクション名、スクロール方向、再現手順を添えると、AIが修正対象を絞れます。

毎回の完了条件

変更点を説明 → 実画面で確認 → PCとスマホ幅を比較 → 逆スクロール・再読み込み・停止状態を確認 → 採用版を保存。この繰り返しを制作スキルにします。

CHAPTER 12

制作ファイルと、再現の進め方

このLPはビルド不要の静的サイト。別案件でも同じ構成から始められます。

基本のフォルダ構成

```
site/
  index.html          本文・リンク・動画
  styles.css          共通デザイン・導入〜比較
  motion.js           スクロールと比較の演出
  extension.css/js    商品・職人・細部の比較
  completion.css/js   料金・時間・流れ・FAQ
  opening.css/js      導入動画の演出
  mascot.css/js       キャラクターの状態と移動
  atmosphere.css/js   背景の質感・帯・輪郭文字
  assets/             画像・動画・ポスター
  guide/              この制作ガイド
src/
  prompts/images.json   画像生成の指示（全17枚）
  masters/              生成した制作マスター（PNG）
  generate_images.py    画像の生成
  export_assets.py      配信用への圧縮
  film/film.html        導入ムービーの画面
  render_film.py        ムービーの撮影・書き出し
  build_page.py         LPのHTMLを組み立て
  build_guide.py        このガイドを組み立て
```

01　静止状態で完成させる

文章、見出し、商品、CTA、スマホレイアウトを先に確認。JavaScriptがなくても主な情報が読める構造を目指します。

02　演出を1種類ずつ追加

登場 → 比較 → 固定区間 → 動画のつなぎ → キャラ移動の順で追加。一度に変更する範囲を小さくすると、不具合の原因を追いやすくなります。

03　ローカルで確認する

例：Pythonがある環境なら、siteフォルダで python3 -m http.server 8766 を実行し、http://127.0.0.1:8766/ を開きます。これは自分のPC用で、外部共有URLではありません。

別の商品に差し替えるとき

コピーは src/build_page.py の文章を書き換えて実行。画像は src/prompts/images.json の指示を商品に合わせて書き換え、生成 → 圧縮の順に実行します。色は styles.css 冒頭の --accent（アクセント色）と --ink（地の色）が起点です。比較画像の縦横比やズーム位置は、商品の形に合わせて調整が必要です。

引き継ぎに残すもの

ソース一式、採用素材、素材の利用条件、生成プロンプト、シーン表、公開手順、直前の正常版、未完了事項をまとめます。

CHAPTER 13

公開と更新を、確実に届ける

手元のブラウザでの成功だけでは、外部公開版の更新完了にはなりません。

現在の状態

このサンプルは未公開です。確認はすべて制作PC上（ローカル）で行いました。外部URLでの表示確認は、公開先を決めてから行います。

公開までの順番

① 使用許諾・掲載文言・リンクを確認 → ② 公開対象フォルダ（site）を確定 → ③ ホスティングへアップロード → ④ デプロイ成功と版を確認 → ⑤ 外部URLを別ブラウザ・スマホで開く → ⑥ CTAまで操作して確認。

反映されないときに見る順番

まずURL。localhost／127.0.0.1はローカル、公開ドメインは配信版です。次に公開履歴と対象ファイル。アップロード失敗や別フォルダの公開がないか確認します。それでも同じ版が出ない場合に、キャッシュを確認します。キャッシュだけを原因と決めつけません。

版管理と戻せる仕組み

Gitで変更を保存し、公開時の版と確認日時を記録。画像・動画名に版を付けると更新を配信しやすくなります。直前の正常版に戻す手順を準備し、公開用フォルダにAPIキーなどの秘密情報や不要な元データを含めないようにします。

未接続のもの

LINEボタン（2か所）はデモ用の案内表示で、押しても登録や送信は行われません。実案件では正しい登録先への接続が別作業になります。検索エンジンには載らない設定（noindex）にしています。

運用で見るもの

表示・リンク切れ、動画の読込、問い合わせ到達、アクセスと離脱、容量・通信費を定期確認。アクセス解析を導入する場合は目的に必要な範囲に絞り、プライバシー告知や同意要否も確認します。

CHAPTER 14

公開前に、体験と性能を確認

よく動くことと、快適に読めることを両立させます。

表示・操作

- iPhone Safari／Android Chrome／PCの実機で確認
- 横幅変更・縦横回転・逆スクロール・途中再読込を確認
- キャラが本文・料金・ボタンを隠さない
- キーボードでボタンを操作でき、フォーカスが見える
- FAQと比較操作、最終CTAが正しい行き先につながる

動き・動画

- 「動きを止める」とprefers-reduced-motionに対応
- 自動再生を拒否されても静止画と本文が見える
- 音声を勝手に流さない
- タブ非表示・画面外で不要な再生を抑える
- 読込失敗時の代替表示とキャラ素材のフォールバックを確認

容量・速度

LP本体の配信ファイル（この制作ガイドを除く）は合計約11MB、うちPC用動画が約5.5MB、スマホ用動画が約1.8MBです。これは初回通信量の実測値ではありません。画面外の画像は遅延読込にしています。大きなぼかしや同時アニメーションは実機で負荷を確認します。

表示内容・権利

生成画像はイメージであることをページ内に明示しています。実案件では、実際の仕上がり・料金・納期の条件を事業者が確認します。参考サイトの表現を研究しても、キャラ・音楽・素材を無断転載しないでください。AIサービスの商用利用条件と参照素材の権利を確認します。

今回確認した範囲

PCの画面幅（1024／1280／1440px）とスマホ幅（375px）の表示を、制作PCのブラウザで確認しました。導入の切り替え、比較スライダー、3か所のズーム、時計・流れの連動、FAQ、6ポーズの切り替

えが動作し、エラー表示はありません。実機での検証、性能スコアの測定、本番の問い合わせ導線は未実施です。

CHAPTER 15

費用は「契約・生成・工数」で考える

動きが3倍になっても、費用が一律3倍になるわけではありません。

今回の画像生成で確認できた量

2026年11月20日の記録：画像生成APIの成功した呼び出しは18回（採用17枚＋作り直し1枚）。APIが返した利用量の合計は約57,000トークンで、1枚あたりの出力は1,372トークン、参照画像を渡した回は入力が約1,500〜3,300トークン増えました。

金額は未確認

トークン単価をこの制作では確認していないため、円換算の金額は記載していません。実際の請求額はAPIの利用明細で確認してください。Claude Codeの利用料、制作工数も未集計です。

動画にかかった生成費

導入ムービーは静止画3枚から手元のPCで書き出したため、動画生成AIの費用は発生していません。3枚の画像生成分は上の18回に含まれます。

見積もりの簡単な例：工数の仮定

この構成を流用し、コピーと素材の差し替え・演出調整に仮に20時間、内部単価3,000円／時なら作業費60,000円。これに画像生成の実費、制作ツールの利用料、公開先・ドメイン費、税等を加えます。市場相場や今回の請求実績ではなく、予算を組む計算例です。

追加したい内容ごとの費用要因

- 移動・フェードの追加：主に実装と検証時間。
- 新しいキャラ動作：ポーズ生成・そろえ込み・遷移調整。
- 動画の追加：静止画の枚数、または動画生成AIの試行回数・尺・解像度。
- 訪問者増加：ホスティング・通信条件による。

CHAPTER 16

再利用するための記録と参照先

次のLPで同じ手順を使えるように、制作記録を残します。

残すと役立つ4つの台帳

- 素材台帳：ファイル名／用途／参照素材／権利／生成条件／採用版。
- 演出台帳：シーン／開始条件／動作／終了状態／スマホ差分。
- 費用台帳：サービス／モデル／試行回数／消費量／採否。
- 公開台帳：版／公開日時／URL／確認端末／未完了事項／戻し方。

今回の素材台帳（抜粋）

- キャラクター：立ち姿／まばたき／手を振る／案内／ジャンプ6ポーズのシート（計5枚）
- 商品：研ぎ直し後・前の三徳包丁、ペティナイフ、キッチンばさみ（計4枚）
- シーン：工房の背景、見立て、研ぎ、配送、梱包（計5枚）
- 動画用：開始、終了、決めカット（計3枚）

スキル化するなら、入力と合格条件を固定

入力は目的、対象者、原稿、素材、参考、CTA、予算。工程は構成→素材→3シーン→全体→実機確認→公開→外部URL確認。合格条件は本文の可読性、素材の一貫性、停止時の表示、リンク接続、公開版一致。案件固有のコピーや未確認の料金を固定ルールにしないようにします。

参照資料

- 一次資料：このLPのHTML／CSS／JavaScript・画像・動画、画像生成の指示（src/prompts/images.json）、制作時の画面確認。
- Web実装の公式リファレンス：https://developer.mozilla.org/en-US/docs/Web/API/Window/requestAnimationFrame
- 動画の自動再生：https://developer.mozilla.org/en-US/docs/Web/Media/Guides/Autoplay
- 画像生成API：https://platform.openai.com/docs/guides/image-generation

外部共有用 / 2026.11.20 制作ガイドです。TOGINOは制作サンプル用の架空のサービスで、実在の事業者・商品とは関係ありません。費用は実績・未確認・仮定を区別して記載しています。LPのLINEボタンはデモです。