

Lottie Animations — Workbook

This workbook turns the course into one finished, interactive Lottie you could ship today. You will animate a small piece inside the Bodymovin-supported subset of After Effects, export and read the JSON, optimise it with dotLottie and the LottieFiles optimiser, embed it on a page with the lottie-player web component, wire up a hover or scroll trigger, and host and hand it off through LottieFiles. Work through one section per module and finish with a tiny, sharp, reactive animation and a clean handoff note.

What Lottie Is and How the Pipeline Works

Decide whether your idea is a Lottie at all, then map it onto the After Effects to Bodymovin to player pipeline.

Exercise: Is It a Lottie? Format Decision Drill

Take three animation ideas you actually want to build (for example a loading spinner, an onboarding illustration, and a product photo flythrough) and decide for each whether Lottie is the right format, justifying it against a GIF and a video.

- For each idea, state whether it is shapes and lines (Lottie territory) or photographic or footage (not a Lottie).
- Estimate how it would be used: small icon, full-screen hero, looping loader, or one-shot illustration.
- Decide if it needs to react to the user (hover, click, scroll); if yes, that pushes strongly toward Lottie.
- Write the final call for each: Lottie, GIF, or video, and one sentence on why.

Worksheet: Pipeline Map

Lock the pipeline for the one animation you will build in this workbook, so every later step traces back to a single plan.

Animation name / purpose

Authoring tool (After Effects)

Export plugin (Bodymovin)

Primary destination (website / app / Figma prototype)

Player to ship with (lottie-player web component / lottie-web / native)

Output format intended (.json / .lottie)

Will it be interactive? (hover / click / scroll / none)

Checklist: Concept-Ready Pass

-] The idea is vector shapes and lines, not photos or video footage
-] I can say in one sentence why Lottie beats a GIF or video here
-] I know where it will be used and at roughly what size
-] I have decided whether it needs to be interactive
-] I know which player will load the file on the destination
-] I have a single, named animation chosen to build through the whole workbook

Animating for a Clean Bodymovin Export

Build the animation inside the supported subset on shape layers, lean on Trim Paths, and keep the comp tidy and export-safe.

Exercise: Shape-Only Build Drill

Create your chosen animation using only shape layers and supported properties. If you start from Illustrator art, convert it; if you start in After Effects, draw with shape tools and the Pen.

- Convert any imported vector art with Create Shapes from Vector Layer, and confirm the layers are now native shapes.
- Animate using only position, scale, rotation, opacity, anchor point, path, stroke, fill, and Trim Paths.
- Build at least one move with Trim Paths (a draw-on stroke or a spinning loader End and Offset).
- List anything you wanted to use but could not (3D, glow, blur) and describe the shape-based alternative you chose instead.

Worksheet: Supported-Feature Audit

Audit your comp against the Bodymovin-supported subset before exporting, so nothing gets silently dropped. Fill one row per significant layer or effect.

Layer / effect name

Type (shape / mask / matte / raster / 3D / effect)

Supported by Bodymovin? (Y / N / test it)

Animated properties used (position / scale / rotation / opacity / path / trim)

Action (keep / convert to shapes / remove / bake expression)

Verified in export? (Y/N)

Exercise: Comp Hygiene and Bake Pass

Get the composition itself clean and predictable before the first export: right size, sane frame rate, baked expressions, tidy layers.

- Set the comp to a deliberate size and frame rate (for example 512 by 512 at 30 fps) and note both.
 - Bake every expression to keyframes with Convert Expression to Keyframes, working on a copy if you want to keep the live version.
-

- Rename every layer clearly and delete hidden, empty, or guide layers.
-
- Trim the work area so the in and out points match the animation exactly, then state the final duration in seconds.
-

Checklist: Export-Ready Pass

- Everything visible is a shape layer, with no stray rasters
- No 3D layers, cameras, blurs, glows, or unsupported effects remain
- All expressions are baked to keyframes
- Layers are named and the timeline is tidy
- Comp size and frame rate are deliberate and modest (30 fps where possible)
- Point counts are as low as the shapes allow
- The work area is trimmed to the exact animation length

Exporting and Optimising the File

Run Bodymovin, read the output honestly, then shrink the file with dotLottie and the LottieFiles optimiser and confirm it still plays everywhere.

Exercise: First Export and Read-the-JSON Drill

Export your animation with Bodymovin and inspect the result, so you understand the file you made rather than treating it as a black box.

- Render from the Bodymovin panel, choosing standard JSON, and note the destination and file size.
-
- Preview it immediately by dragging it onto lottiefiles.com and watching the full loop.
-
- Open the JSON in a text editor and find fr (frame rate), ip and op (in and out frames), w and h, and the assets array.
-
- If the assets array is large or the file is surprisingly big, identify what slipped in (raster, points, frame rate) and plan the fix.
-

Worksheet: Optimisation Ledger

Record each optimisation step and its effect on size, so you keep only the cuts that do no visible harm. Fill one row per step.

Step (remove raster / simplify paths / lower fps / convert to dotLottie / run optimiser)

File size before (KB)

File size after (KB)

Still looks identical? (Y/N)

Kept this change? (Y/N)

Exercise: Cross-Player Sanity Test

Prove the optimised file behaves outside After Effects, on real players and backgrounds, before you build it into a page.

- Preview on lottiefiles.com on both a light and a dark background and confirm transparency is correct.
-
- Load it in the player you will ship with and watch the whole loop for missing or frozen pieces.

-
- Check the loop point: confirm the last frame meets the first with no visible jump.
 - Note the final size, frame rate, whether it loops, and any matte or feature you were unsure about.
-

Checklist: Optimised-File Gate

- [] The file previews correctly and matches the After Effects comp
- [] Embedded rasters are removed or externalised
- [] Frame rate is no higher than the motion needs
- [] The file is converted to dotLottie where it helps
- [] The LottieFiles optimiser has been run and the result still looks identical
- [] Transparency is correct on light and dark backgrounds
- [] The loop is seamless and the final size is recorded

Playing, Triggering, and Hosting on the Web

Embed the Lottie with the web component, make it react with a real trigger, then host and hand it off through LottieFiles.

Exercise: Web-Component Embed Drill

Get the animation playing on an actual page using the lottie-player web component, controlling how it plays with attributes only.

- Add the lottie-player tag, set src to your file, and set autoplay and loop appropriately for the use case.
 - Set background to transparent and size it with style width and height; confirm it stays sharp.
 - Try a non-default speed (for example 0.5 or 1.5) and note how it changes the feel.
 - Confirm it plays in place on the page and looks identical to the LottieFiles preview.
-

Worksheet: Interactivity Plan

Plan how the animation reacts to the user before wiring it, mapping each interaction to a specific segment. Fill one row per interaction.

Interaction (hover / click / cursor / play-on-scroll / scrub-on-scroll)

Segment name and frame range (e.g. open = 0 to 30)

What it should do (play forward / reverse / advance state / scrub to frame)

Trigger element (button / icon / illustration container)

Tested in the real player? (Y/N)

Exercise: Make It React Drill

Add one genuine interaction using a segment and the LottieFiles interactivity approach, so the file does something a GIF never could.

- Define at least one segment in your animation by frame range (for example open and close states).
- Wire a hover or click to play the relevant segment forward, and the reverse on mouse-out or second click.
- If you chose scroll, decide play-on-scroll (start on entry) versus scrub-on-scroll (frame follows scroll) and

implement one.

- Confirm the interaction is smooth, lands on a clean state, and respects users who prefer reduced motion.
-

Checklist: Ship-and-Handoff Gate

- The Lottie embeds and plays correctly on a real page via the web component
- Loop, autoplay, speed, and background are set deliberately for the use case
- At least one interaction (hover, click, or scroll) works and lands on a clean state
- The final optimised file is uploaded to a LottieFiles workspace as the single source of truth
- A shared project link carries the file, preview, and embed together
- The handoff note states use, interactivity, segments, size, frame rate, and anything to test
- Reduced-motion users are respected with a pause or simplified fallback

Your Action Plan

1. Pick one shape-based idea and confirm it is genuinely a Lottie job, not a GIF or video.
2. Build it in After Effects on shape layers using only supported properties, including at least one Trim Paths move.
3. Audit the comp against the Bodymovin-supported subset, bake all expressions, and tidy and name every layer.
4. Set a deliberate, modest comp size and frame rate (for example 512 by 512 at 30 fps) and trim the work area to the animation.
5. Export with Bodymovin, note the file size, preview it on LottieFiles, and read the JSON to recognise fr, ip, op, and assets.
6. Optimise from the source down: remove rasters, simplify paths, convert to dotLottie, and run the LottieFiles optimiser, previewing after each step.
7. Embed it on a page with the lottie-player web component and set src, loop, autoplay, speed, and a transparent background.
8. Plan segments and add at least one interaction (hover, click, or scroll) mapped to a frame range.
9. Test across players and on light and dark backgrounds, confirm a seamless loop, and add a reduced-motion fallback.
10. Upload the final .lottie to a LottieFiles workspace, share the project link, and write a one-paragraph handoff note.

