

Glaze for WP (and for BricksBuilder)

“ Update 1.2.0: Preview of the Presets directly in the WP backend now possible.
Update 1.3.0: Preview improved (more reliable + scroll preview)

What It Is

Glaze for WP / Bricks brings Tailwind-style, attribute-driven animations to any WordPress site (with special affinity for Bricks Builder) using GSAP under the hood. Instead of writing JavaScript for every animated element, you simply add a `data-animate` attribute (or a CSS class, if you prefer class-based syntax) containing a compact animation string, and the plugin's runtime script parses it and drives GSAP automatically.

The Technology Behind It

GSAP (GreenSock Animation Platform) is the actual animation engine — it handles all the tweening, easing, timelines, and (optionally) scroll-triggered behavior via its ScrollTrigger plugin. GSAP itself knows nothing about your markup; it's just invoked with the right parameters.

GlazeJS is a small ES module that sits between your HTML and GSAP. It scans the DOM for elements carrying the configured attribute (default `data-animate`) or class prefix (default `glaze-`), parses the pipe-separated animation string on each one (e.g. `from:opacity-0|y-30|duration-0.6`), and translates that into the equivalent GSAP `.from()`/`.to()`/`timeline` calls — including support for staggered children (`[\&>]:...`), nested selectors (`[\&_path]:...`), transform origins, easing functions, and scroll-trigger properties.

This plugin is the WordPress "glue" layer: it loads GSAP and Glaze (from CDN or a locally hosted copy), lets you configure breakpoints, global defaults, reusable presets, and automatic class-to-animation mappings through an admin UI, and injects the resulting configuration into the front end via a small runtime script (`runtime.js`). Everything is stored in flat JSON/files (`/config/settings.json`, `/lib/*.js`) rather than the options table, and is fully removed on uninstall.

Getting Started

1. After activation, go to the **Glaze** menu item in the WordPress admin sidebar (superhero icon).
2. The admin screen has five tabs: **Libraries**, **General**, **Breakpoints & Defaults**, **Presets**, and **Mappings**.
3. Make your changes and click **Save changes** in the header. **Reset** restores all settings to plugin defaults (with a confirmation prompt).

Tab-by-Tab Guide

Libraries

Controls how GSAP, ScrollTrigger, and GlazeJS are loaded on the front end.

- **Library mode:** `CDN` (jsDelivr/esm.sh — fastest to set up, but relies on external requests) or `Local` (serves self-hosted copies from the plugin's `/lib` folder — no third-party requests, but you must download each library first).
- **Use existing global GSAP:** Enable if another plugin or your theme already loads GSAP globally, so Glaze reuses `window.gsap` instead of loading a duplicate copy.
- **Local libraries** section: For each library (GSAP core, ScrollTrigger, GlazeJS), you can **Download** (fetch a fresh local copy), **Refresh** (re-download an existing one, e.g. after an update), or **Remove** (delete the local copy, falling back to CDN until re-downloaded).

General

- **Data attribute:** The HTML attribute Glaze scans for animation strings (default `data-animate`). Only change this if it conflicts with another plugin/theme.
- **Class prefix:** Prefix for the optional class-based syntax (default `glaze`), e.g. `class="glaze-from:opacity-0"`.
- **Enable ScrollTrigger:** Registers GSAP's ScrollTrigger plugin so animations can react to scroll position (`scrollTrigger.*` properties).
- **Force GPU (force3D):** Forces animated elements onto GPU-accelerated layers for smoother motion (may occasionally blur very small text).
- **Watch DOM & re-animate:** Watches for elements added after page load (AJAX content, Bricks-injected components) and automatically animates them.
- **Watch debounce (ms):** Delay before re-scanning after a DOM change is detected (only relevant when Watch DOM is on).

- **Run inside Bricks builder:** Plays animations live while editing in the Bricks canvas (off by default, since constant replay can be distracting while building).
- **Verbose console logging:** Logs parsed animation strings and GSAP calls to the console — useful for debugging, should stay off in production.

Breakpoints & Defaults

- **Breakpoints:** Named media queries used for responsive prefixes like `@tablet:from:opacity-0`. The name is what you reference in animation strings; the value is the actual CSS media query. `default` applies when no `@breakpoint:` prefix is used.
- **Global defaults:** Baseline settings applied automatically everywhere.
 - **Element defaults:** Fallback properties for every animated element (e.g. `duration-0.6|ease-power2.out`) — overridden by anything set directly in an element's own attribute.
 - **Timeline defaults:** Applied to every internal timeline Glaze creates (every element is wrapped in one), useful for a global easing or default `scrollTrigger.trigger`.

Presets

Reusable animation strings referenced by name inside `data-animate` (e.g. `fadeIn`) or as a class (`glaze-preset-fadeIn`). Categories only organize the admin UI — they have no front-end effect, and **preset names must be unique across all categories** since they're flattened into one list at runtime.

- **+ Add category / + Preset:** create new groups/rows.
- **Preview** button on each preset row: opens a modal that plays the animation live on a sample layout (single box, staggered row, grid, or SVG shape) using the same GSAP/Glaze pipeline as the front end — handy for tuning a string without leaving wp-admin.
- Advanced users can bypass the UI and hand-edit `/config/settings.json` directly to add many more presets at once; a `glaze-opus-docs-2.0.md` file in the plugin directory documents the full animation-string syntax for AI-assisted ("Vibe Coding") editing.

Mappings

Automatically attaches an animation to every element matching a CSS class or selector, without needing to add `data-animate` manually — useful for markup you don't control directly (e.g. icon wrappers Bricks generates).

- Left field: CSS class or selector to match (e.g. `.fv-a-icon` or `fv-a-icon`).
- Right field: raw animation string or `preset-name` to apply.

Writing Animation Strings

Animation strings are pipe-separated (|) key-value pairs, e.g.:

```
from:opacity-0|y-30|duration-0.6|ease-power2.out
```

Common patterns:

- `from:` / `to:` define the animated state (properties before the tween vs. the target values).
- Numeric values follow the property name with a hyphen: `y-30`, `scale-0.8`, `rotate-[-45]` (brackets allow negative/complex values).
- `duration-`, `delay-`, `ease-`, `stagger-`, `repeat-`, `yoyo-` control timing/behavior.
- `[&>*]:` prefix applies the animation with staggering to direct children (useful for lists/grids).
- `[&_path]:` / `[&_line]:` prefix targets nested SVG elements.
- `scrollTrigger.trigger- [&]`, `scrollTrigger.start-[top_85%]`, `scrollTrigger.scrub-1`, etc. wire the animation to scroll position (requires **Enable ScrollTrigger**).
- `@breakpoint:` prefix (e.g. `@tablet:from:opacity-0`) restricts a rule to a given breakpoint's media query.

Practical Notes

- Changing the **library mode** to Local without first downloading a library will simply cause the plugin to fall back to CDN for that library until you download it.
- **Watch DOM** uses a Firefox-safe, childList-only `MutationObserver` — it intentionally never listens for attribute changes, avoiding an infinite re-trigger loop that can occur when GSAP rewrites inline styles.
- If **Run inside Bricks builder** is left off, animations simply won't fire while editing in the canvas; they still work normally on the live front end.
- All plugin data lives in flat files (`/config/settings.json`, `/lib/*.js`) — nothing is stored in the WordPress options table, and everything is deleted automatically on uninstall.

Revision #11

Created 2026-07-17 19:00:56 UTC by art10m

Updated 2026-07-22 09:49:04 UTC by art10m