Glaze Cheat Sheet

Tailwind-style animations for HTML. Describe GSAP animations with data attributes—no JS animation code.

---

1. Core Concept

Glaze is a **~3kb syntax layer on top of GSAP**. You write animation strings in HTML attributes; Glaze parses them into GSAP calls.

```
<div data-animate="from:opacity-0|y-50|duration-1"></div>
<!-- becomes -->
gsap.from(element, { opacity: 0, y: 50, duration: 1 })
```

Requirements: GSAP must be installed and passed to Glaze. (GSAP has its own license—check compliance for commercial use. Glaze itself is MIT.)

Features: responsive breakpoints (via GSAP `matchMedia`), timelines, dot notation for nested props, full GSAP access (ScrollTrigger, easing, stagger), library-agnostic design.

---

2. Install & Setup

```
npm i glazejs   # or yarn add / pnpm add / bun add glazejs
```

```
import gsap from "gsap";
import glaze from "glazejs";

glaze({
  lib: { gsap: { core: gsap } },   // ONLY required option
});
```

CDN:

```
import glaze from "https://esm.sh/glazejs";
import gsap from "https://esm.sh/gsap";
glaze({ lib: { gsap: { core: gsap } } });
```

Glaze auto-finds all `data-animate` elements and animates them.

---

3. Full Configuration

```
glaze({
  lib: { gsap: { core: gsap } },          // required

  element: document.querySelector("#app"), // search scope (default: document)
  dataAttribute: "data-move",              // custom attr (default: "data-animate")
  className: "animate",                    // enable class-based syntax instead

  breakpoints: {                           // responsive (GSAP matchMedia)
    default: "(min-width: 1px)",           // runs when no bp specified
    sm: "(min-width: 640px)",
    md: "(min-width: 768px)",
    lg: "(min-width: 1024px)",
  },

  defaults: {                              // global default settings
    tl: "ease-power2.inOut",               // applies to every timeline
    element: "duration-1",                 // applies to every element
  },

  presets: {                               // reusable animation shortcuts
    fadeIn: "from:opacity-0|duration-1",
    slideUp: "from:y-50|opacity-0",
  },

  watch: { debounceTime: 500 },            // watch DOM & re-animate (default: false)
});
```

---

4. Anatomy of an Animation String

```
[breakpoint][selector]:state:properties
```

Example: `@lg:[&>div]:from:opacity-0|y-50|stagger-0.1|duration-0.5|ease-power2.out`

---

5. States *(every animation needs one)*

Syntax	GSAP equivalent	Meaning
`from:...`	`gsap.from`	Start at these values → animate to natural state
`to:...`	`gsap.to`	Animate from current state → these values
`from:... to:...`	`gsap.fromTo`	Animate between two explicit states

```
<div data-animate="from:opacity-0"></div>
<div data-animate="to:xPercent-50"></div>
<div data-animate="from:opacity-0.5 to:opacity-1"></div>
```

---

6. Properties

- List after state, separated by **pipes** `|`.
- Parsed by splitting at the **dash** `-`: first part = property name, second = value.
- Values auto-convert to number, boolean, or string.

```
<div data-animate="to:opacity-1|yPercent-10|duration-0.5"></div>
```

Need	Syntax	Result
Nested props (dots)	`to:scale.x-2|scale.y-2`	`{ scale: { x: 2, y: 2 } }`
Negative values (brackets)	`to:xPercent-[-50]`	`{ xPercent: -50 }`
Values with spaces (underscores)	`to:boxShadow-[0_0_50px_20px_red]`	`0 0 50px 20px red`

7. Breakpoints

Prefix with `@breakpoint:` — animation only runs at that screen size.

```
<div data-animate="@sm:from:opacity-0"></div>          <!-- sm only -->
<div data-animate="@lg:from:opacity-0|rotate-180"></div>
```

Stack multiple with spaces; larger breakpoints add to/override smaller:

```
<div data-animate="@sm:from:opacity-0|y-50 @lg:from:scale-0.5"></div>
```

`default` breakpoint runs when none specified (defaults to `(min-width: 1px)`).

---

8. Selectors *(target children)*

Bracket notation `[selector]:`. The `&` = the parent element.

```
<div data-animate="[&>h1]:to:opacity-1|stagger-0.25">
  <h1>One</h1><h1>Two</h1><h1>Three</h1>
</div>
```

Combine with breakpoints: `@sm:[&>h1]:to:opacity-1|stagger-0.25`

---

9. Alternative Input Modes

Custom attribute: set `dataAttribute: "data-move"` → use `<div data-move="from:opacity-0">`.

Class-based: set `className: "animate"` → write:

```
<div class="animate-from:opacity-0|duration-1"></div>
```

---

10. Timelines (`tl`)

Add `tl` keyword to make an element a **timeline container**. All child elements are automatically included in its scope.

```
<div data-animate="tl defaults:ease-elastic|duration-1">
  <div data-animate="to:rotate-360"></div>
  <div data-animate="to:rotate-360"></div>
</div>
```

Timeline defaults: `tl defaults:ease-elastic|duration-4 yoyo-true`

Timing control with `tl:[...]` on children:

Value	Meaning
`tl:[+=1]`	Start 1s **after** previous ends
`tl:[-=1]`	Start 1s **before** previous ends (overlap)
`tl:[<]`	Start **at same time** as previous

Responsive timelines: mix breakpoints inside children:

```
<div data-animate="tl defaults:power2.inOut|duration-2">
  <div data-animate="to:rotate-360 @lg:to:xPercent-[50]"></div>
  <div data-animate="tl:[-=1] to:rotate-360 @lg:to:xPercent-[-50]"></div>
</div>
```

Named timelines — reference from anywhere in the DOM:

```
<div data-animate="tl/main defaults:power2.inOut|duration-2">
  <div data-animate="to:rotate-360"></div>
</div>
<div data-animate="tl:main-[-=1] to:scale-1.5"></div>
```

`tl/name` = create named timeline · `tl:name` = join it.

> **Internally, Glaze creates a timeline for every element** even without `tl`, so `tl` defaults apply globally.

---

11. ScrollTrigger

No special syntax—it's just the GSAP `scrollTrigger` property via dot notation.

Setup:

```
import ScrollTrigger from "gsap/ScrollTrigger";
gsap.registerPlugin(ScrollTrigger);
```

```
<div data-animate="from:opacity-0|y-50|scrollTrigger.trigger-[&]"></div>
```

Option	What it does
`scrollTrigger.trigger-[&]`	Element that triggers animation (`[&]` = this element)
`scrollTrigger.start-[top_center]`	When to start (element pos + viewport pos)
`scrollTrigger.end-[bottom_top]`	When to end
`scrollTrigger.scrub=true`	Link progress to scroll position
`scrollTrigger.markers=true`	Show debug markers

> Use underscores for spaces: `[top_center]` not `[top center]`.

---

12. Defaults (behavior)

```
defaults: {
  tl: "defaults:ease-power2.inOut scrollTrigger.trigger-[&]", // every timeline
  element: "duration-1",                                      // every element
}
```

Because every element gets an internal timeline, `tl` defaults effectively apply everywhere (great for global ScrollTrigger/easing).

Override per element:

```
<div data-animate="from:opacity-0"></div>           <!-- default 1s -->
<div data-animate="from:opacity-0|duration-0.5"></div> <!-- overrides -->
```

13. Presets

Define reusable strings in config; reference with `preset-`.

```
presets: {
  fadeIn: "from:opacity-0|duration-1",
  slideUp: "from:y-50|opacity-0|duration-0.5",
  helicopter: "from:rotate-2160|duration-5",
}
```

```
<div data-animate="preset-fadeIn"></div>
<div class="animate-preset-helicopter"></div>        <!-- class mode -->
<div data-animate="preset-fadeIn scrollTrigger.trigger-[&]"></div> <!-- combine -->
```

Presets simply expand into the full string before processing.

---

14. Quick Syntax Reference

Token	Purpose
`from:` `to:`	states (fromTo = use both)
`|`	separates properties
`-`	splits property name / value
`.`	nested property (`scale.x-2`)
`[-50]`	negative / literal value
`[0_0_50px_red]`	value with spaces (underscores)
`@bp:`	breakpoint prefix
`[&>sel]:`	child selector (`&` = parent)
`tl`	timeline container
`tl/name`	named timeline
`tl:name` / `tl:[+=1]`	join timeline / timing offset
`preset-name`	apply preset
`defaults:`	timeline-scoped defaults

Token	Purpose
`scrollTrigger.*`	ScrollTrigger options
`[&]`	"this element" (trigger/self)

Full complex example:

```
<div data-animate="@lg:[&>div]:from:opacity-0|y-50|stagger-0.1|duration-0.5|ease-power2.out">
  <div>Card 1</div><div>Card 2</div><div>Card 3</div>
</div>
```

On large screens, each child div fades in from 50px below, staggered by 0.1s.

Revision #2
Created 2026-07-18 10:01:10 UTC by art10m
Updated 2026-07-18 10:07:23 UTC by art10m