

# CSS Properties That Landed / Gained Major Support After 2025

This PDF lists 20 CSS properties (and related shorthand) that were introduced, standardized, or gained significant cross-browser support after 2025. Each entry includes: Property name, Year (intro / support), Syntax, Purpose / Use case, Small working example, Browser support notes, and Reference links (MDN / W3C / Chrome docs). Notes: Dates and support information were checked against MDN, Chrome.dev, W3C drafts, and CanIUse as of early 2026.

## **position-anchor — 2026 (*major support rolling out 2026*)**

**Syntax:** position-anchor: auto | none | --anchor-name;

**Purpose / Use Case:** Associate a positioned (absolute/fixed) element with a named anchor element so it can be tethered to that anchor.

**Small working example:**

```
/* anchor element */ .btn { anchor-name: --menu-anchor; } .tooltip { position: absolute; position-anchor: --menu-anchor; position-area: bottom; }
```

**Browser Support Notes:** Chrome (125+), Firefox (from ~147+), Safari partial; check MDN/CanIUse for current details.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/position-anchor>

<https://www.w3.org/TR/css-anchor-position-1/>

## **anchor-name — 2026 (*standardized in Anchor Positioning spec; support 2026+*)**

**Syntax:** anchor-name: --my-anchor;

**Purpose / Use Case:** Mark an element as an anchor by giving it one or more anchor names that other elements can reference with position-anchor.

**Small working example:**

```
header .menu { anchor-name: --menu-anchor; } .popup { position-anchor: --menu-anchor; }
```

**Browser Support Notes:** Implemented in Chromium-based browsers and being adopted in other engines in 2026.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/anchor-name>

<https://www.w3.org/TR/css-anchor-position-1/>

## **anchor-scope — 2026**

**Syntax:** anchor-scope: none | subtree | document;

**Purpose / Use Case:** Limit the scope in which an anchor association is valid (e.g., restrict anchor lookup to a subtree).

**Small working example:**

```
dialog { anchor-scope: subtree; }
```

**Browser Support Notes:** Marked Baseline 2026 on MDN; check CanIUse for exact engine coverage.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/anchor-scope>

## **position-area — 2025–2026 (*introduced 2025, broader support 2026*)**

**Syntax:** position-area: top | bottom | left | right | center | span-x-start | span-all etc;

**Purpose / Use Case:** Place an anchor-positioned element into one or more cells of an implicit 3x3 grid around its anchor (convenient for tooltips/popovers).

**Small working example:**

```
.anchor { anchor-name: --a; } .popup { position: absolute; position-anchor: --a; position-area: top; }
```

**Browser Support Notes:** Chrome first, other browsers adding support during 2025–2026.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/position-area>

<https://developer.chrome.com/blog/anchor-positioning-api>

## **position-try (shorthand) — 2026 (*shorthand built from new anchor fallback properties*)**

**Syntax:** position-try: ;

**Purpose / Use Case:** Shorthand to define order and fallback positions the UA should attempt when anchoring (helps auto-flip popover fallback).

**Small working example:**

```
.popup { position: absolute; position-try: normal flip-block, right; }
```

**Browser Support Notes:** Listed Baseline 2026 on MDN; Chrome and others introducing support.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/position-try>

<https://www.w3.org/TR/css-anchor-position-1/>

## **position-try-fallbacks — 2025–2026**

**Syntax:** position-try-fallbacks: none | ;

**Purpose / Use Case:** Provide a list of fallback position options to try if the initial anchor placement would overflow the container/viewport.

**Small working example:**

```
.popup { position-try-fallbacks: flip-block, most-width; }
```

**Browser Support Notes:** Experimental/Baseline rolling out 2026; check MDN/CanIUse.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/position-try-fallbacks>

<https://developer.chrome.com/blog/anchor-positioning-api>

## **position-try-order — 2026**

**Syntax:** position-try-order: normal | most-width | most-height | custom;

**Purpose / Use Case:** Define preference ordering when trying fallback positions for anchor-positioned elements.

**Small working example:**

```
.popup { position-try-order: most-width; position-try-fallbacks: flip-block; }
```

**Browser Support Notes:** Limited availability; see MDN and CanIUse.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/position-try-order>

## interest-delay — 2025 (*support expanding into 2026*)

**Syntax:** interest-delay: /\* or shorthand interest-delay: \*/

**Purpose / Use Case:** Control the delay between user 'interest' (hover/focus/peek) in an interest-invoker element and firing the interest/loseinterest events—useful for subtle hover previews.

**Small working example:**

```
.item { interest-delay: 200ms 500ms; /* start 200ms, end 500ms */ }
```

**Browser Support Notes:** Experimental; Chrome shipped in 2025 and broader engine support grew through 2026. Check MDN/caniuse.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/interest-delay>

[https://caniuse.com/mdn-css\\_properties\\_interest-delay\\_normal](https://caniuse.com/mdn-css_properties_interest-delay_normal)

## interest-delay-start — 2025

**Syntax:** interest-delay-start: ;

**Purpose / Use Case:** Set delay between showing interest and firing the interest event (part of interest-delay shorthand).

**Small working example:**

```
.item { interest-delay-start: 150ms; }
```

**Browser Support Notes:** Experimental; Chrome/Chromium engines first, follow-up support in other engines during 2026.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/interest-delay-start>

## interest-delay-end — 2025

**Syntax:** interest-delay-end: ;

**Purpose / Use Case:** Set delay between losing interest and firing the loseinterest event.

**Small working example:**

```
.item { interest-delay-end: 400ms; }
```

**Browser Support Notes:** Experimental; check MDN/CanIUse for current engine coverage.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/interest-delay-end>

## corner-shape — 2025

**Syntax:** corner-shape: round | bevel | notch | scoop | squircle | ...;

**Purpose / Use Case:** Create nonstandard corner shapes (beyond border-radius) such as squircles, notches, scoops—enables richer UI shapes without images/SVG.

**Small working example:**

```
.card { border-radius: 12px; corner-shape: squircle; }
```

**Browser Support Notes:** Experimental (Chrome first in 2025); multiple longhands exist (corner-top-left-shape etc.). See MDN/CanIUse.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/corner-shape>

[https://caniuse.com/?search=corner-shape](#)

## corner-top-left-shape — 2025

**Syntax:** corner-top-left-shape: ;

**Purpose / Use Case:** Control the specific corner style for top-left corner when using corner-shape family.

**Small working example:**

```
.btn { corner-top-left-shape: notch; }
```

**Browser Support Notes:** Part of corner-shape family; experimental—see MDN.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/corner-top-left-shape>

## corner-top-right-shape — 2025

**Syntax:** corner-top-right-shape: ;

**Purpose / Use Case:** Set top-right corner style separately (bevel/notch/squircle/etc.).

**Small working example:**

```
.btn { corner-top-right-shape: scoop; }
```

**Browser Support Notes:** Experimental; see MDN/caniuse.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/corner-top-right-shape>

## corner-bottom-left-shape — 2025

**Syntax:** corner-bottom-left-shape: ;

**Purpose / Use Case:** Set bottom-left corner style separately.

**Small working example:**

```
.panel { corner-bottom-left-shape: bevel; }
```

**Browser Support Notes:** Experimental; check MDN.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/corner-bottom-left-shape>

## corner-bottom-right-shape — 2025

**Syntax:** corner-bottom-right-shape: ;

**Purpose / Use Case:** Set bottom-right corner style separately.

**Small working example:**

```
.panel { corner-bottom-right-shape: round; }
```

**Browser Support Notes:** Experimental; check MDN.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/corner-bottom-right-shape>

## scroll-target-group — 2025–2026

**Syntax:** scroll-target-group: none | auto;

**Purpose / Use Case:** Declare an element as a scroll marker group container (useful for CSS carousels/navigation markers).

**Small working example:**

```
.carousel { overflow-x:auto; scroll-target-group: auto; }
```

**Browser Support Notes:** Experimental; Chrome implemented many scroll markers features in 2025; check CanIUse/MDN.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/scroll-target-group>

[https://caniuse.com/mdn-css\\_properties\\_scroll-target-group](https://caniuse.com/mdn-css_properties_scroll-target-group)

## scroll-marker-group — 2025

**Syntax:** scroll-marker-group: none | before | after;

**Purpose / Use Case:** Generate a ::scroll-marker-group pseudo-element for a scroll container to host generated markers (enables JS-free carousels).

**Small working example:**

```
.gallery { scroll-marker-group: after; } li::scroll-marker { width:1em; height:1em; }
```

**Browser Support Notes:** Available in Chromium-based browsers; check MDN for exact status.

**Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/scroll-marker-group>

## scroll-timeline-name — 2025

**Syntax:** scroll-timeline-name: --my-timeline | none;

**Purpose / Use Case:** Name a scroll-driven timeline that can be referenced by animation-timeline so animations follow scroll progress.

### **Small working example:**

```
:root { scroll-timeline-name: --page; } .box { animation: fade 1s linear; animation-timeline: --page; }
```

**Browser Support Notes:** Limited availability (rolling out across engines through 2025–2026).

### **Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/scroll-timeline-name>

<https://caniuse.com/css-scroll-timeline>

## **scroll-timeline-axis — 2025**

**Syntax:** scroll-timeline-axis: block | inline | x | y;

**Purpose / Use Case:** Choose which scrollbar axis drives a named scroll progress timeline (vertical/horizontal).

### **Small working example:**

```
.scroller { scroll-timeline-axis: x; }
```

**Browser Support Notes:** Experimental/limited; check MDN/CanIUse.

### **Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/scroll-timeline-axis>

## **animation-timeline (aka animation-timeline / animation-timeline property) — 2025–2026**

**Syntax:** animation-timeline: | scroll() | none;

**Purpose / Use Case:** Specify which timeline (time-based, scroll-driven, or view-driven) should drive an animation (enables scroll-driven animations).

### **Small working example:**

```
.hero { animation: fly 1s linear; animation-timeline: scroll(--scroller); }
```

**Browser Support Notes:** Limited availability; Chrome first, other engines working toward support in 2026.

### **Reference(s):**

<https://developer.mozilla.org/en-US/docs/Web/CSS/animation-timeline>

[https://developer.mozilla.org/en-US/docs/Web/CSS/Guides/Scroll-driven\\_animations](https://developer.mozilla.org/en-US/docs/Web/CSS/Guides/Scroll-driven_animations)

Generated by ChatGPT — source checks done against MDN, Chrome.dev, W3C drafts and CanIUse (Jan–Feb 2026). For the most up-to-date compatibility, please check the MDN / CanIUse pages linked above.