

# CSS Properties Introduced or Widely Supported After 2025

## 1. position-area

Year of Introduction / Standardization  
2024 — CSS Anchor Positioning Module

Syntax

```
position-area: bottom;
```

Purpose / Use Case

Positions an element relative to its anchor using a predefined placement grid. Used for tooltips, popovers, and UI overlays.

Small Working Code Example

```
.tooltip {  
  position-area: bottom;  
}
```

Browser Support Information

Experimental support in Chromium browsers.

Reference Source

MDN, W3C CSS Anchor Positioning Draft

---

## 2. transition-behavior

Year  
2024

Syntax

```
transition-behavior: allow-discrete;
```

Purpose

Enables transitions for discrete properties like display.

Example

```
dialog {  
  transition-behavior: allow-discrete;  
}
```

Support  
Chrome 117+

Reference  
MDN

### 3. text-autospace

Year  
2024

Syntax

`text-autospace: ideograph-alpha;`

Purpose  
Improves spacing in mixed-language text layouts.

Example

`p { text-autospace: ideograph-alpha; }`

Support  
Chromium experimental

Reference  
CSS Text Level 4 Spec

---

### 4. white-space-collapse

Year  
2024

Syntax

`white-space-collapse: preserve;`

Purpose  
Controls whitespace collapsing behavior.

Example

`pre { white-space-collapse: preserve; }`

Support  
Experimental

Reference  
CSS Text Module Level 4

---

### 5. text-box-edge

Year  
2024

Syntax

`text-box-edge: cap alphabetic;`

Purpose  
Controls vertical alignment of text boxes.

## Example

```
h1 { text-box-edge: cap alphabetic; }
```

## Support

Experimental

## Reference

CSS Inline Layout Module

---

## 6. animation-timeline

### Year

2024

## Syntax

```
animation-timeline: scroll();
```

### Purpose

Links animations to custom timelines like scrolling.

## Example

```
.box {  
  animation-timeline: scroll();  
}
```

## Support

Chrome, Edge

## Reference

Scroll-driven Animations Spec

---

## 7. scroll-timeline

### Year

2024

## Syntax

```
scroll-timeline: --page block;
```

### Purpose

Defines timeline for scroll-linked animations.

## Example

```
.container {  
  scroll-timeline: --page block;  
}
```

## Support

Chromium browsers

---

Reference  
W3C Scroll Animations

---

## 8. caret-animation

Year  
2024

Syntax

`caret-animation: smooth;`

Purpose  
Controls text caret blinking behavior.

Example

```
input { caret-animation: smooth; }
```

Support  
Experimental

Reference  
CSS UI Module

---

## 9. position-try-fallbacks

Year  
2024

Syntax

`position-try-fallbacks: flip-block;`

Purpose  
Defines fallback positions if preferred placement fails.

Example

```
.tooltip {  
  position-try-fallbacks: flip-block;  
}
```

Support  
Experimental

Reference  
Anchor Positioning Spec

---

## 10. position-visibility

Year  
2024

## Syntax

position-visibility: anchors-visible;

## Purpose

Controls element visibility based on anchor availability.

## Example

```
.popover {  
  position-visibility: anchors-visible;  
}
```

## Support

Experimental

## Reference

[CSS Anchor Positioning](#)

---

## 11. anchor-scope

### Year

2024

## Syntax

anchor-scope: all;

## Purpose

Limits scope for anchor references.

## Example

```
.container { anchor-scope: all; }
```

## Support

Experimental

## Reference

[W3C Draft](#)

---

## 12. ruby-align

### Year

Updated support mid-2020s

## Syntax

ruby-align: center;

## Purpose

Aligns ruby annotations relative to base text.

## Example

```
ruby { ruby-align: center; }
```

Support  
Modern browsers

Reference  
MDN

---

### 13. box-decoration-break

Year  
Widespread support mid-2020s

Syntax

```
box-decoration-break: clone;
```

Purpose  
Controls border rendering across line breaks.

Example

```
span { box-decoration-break: clone; }
```

Support  
All modern browsers

Reference  
MDN

---

### 14. font-variant-emoji

Year  
2024

Syntax

```
font-variant-emoji: text;
```

Purpose  
Controls emoji rendering style.

Example

```
p { font-variant-emoji: text; }
```

Support  
Experimental

Reference  
CSS Fonts Module

---

### 15. input-security

Year

2024

## Syntax

`input-security: disc;`

### Purpose

Masks input text similar to password fields.

### Example

```
input { input-security: disc; }
```

### Support

Experimental

### Reference

CSS UI Spec

---

## 16. print-color-adjust

### Year

2023–2025

## Syntax

`print-color-adjust: exact;`

### Purpose

Controls color rendering during printing.

### Example

```
img { print-color-adjust: exact; }
```

### Support

Modern browsers

### Reference

MDN

---

## 17. zoom

### Year

Re-standardized mid-2020s

## Syntax

`zoom: 1.2;`

### Purpose

Scales elements visually.

### Example

```
.container { zoom: 1.2; }
```

Support  
Chrome, Edge, Safari

Reference  
CSS Viewport Module

---

## 18. image-rendering

Year  
Widely adopted mid-2020s

Syntax

`image-rendering: pixelated;`

Purpose  
Controls scaling algorithm for images.

Example

```
img { image-rendering: pixelated; }
```

Support  
All modern browsers

Reference  
MDN

---

## 19. border-image-source

Year  
Expanded support mid-2020s

Syntax

`border-image-source: url(border.png);`

Purpose  
Uses image as border graphic.

Example

```
.box {  
  border-image-source: url(border.png);  
}
```

Support  
All modern browsers

Reference  
MDN

---

## 20. aspect-ratio

Year  
2021–2025 widespread adoption

Syntax

aspect-ratio: 16 / 9;

Purpose  
Maintains consistent width-height ratio.

Example

```
.video {  
  aspect-ratio: 16 / 9;  
}
```

Support  
All modern browsers

Reference  
MDN

---

## CONCLUSION :

The evolution of CSS after 2025 highlights a shift toward more declarative and component-driven design. Features such as scroll-linked animations, advanced typography controls, and anchor positioning significantly reduce reliance on JavaScript while improving performance and maintainability. As browser support continues to expand, these properties will become essential tools for modern front-end development.

---

**References :**  
(MDN / W3C / Official Drafts)

---

Submitted By: Kaushal D. Rajmandai

---

Roll No: 150096725111