

Browser support
and the rapidly
changing web

Rachel
Andrew

Content lead for Chrome
Developer Relations

CSS Working Group member

rachelandrew.co.uk

**It's an exciting time to be a
front-end developer**

appearance, Array at(), ::backdrop, background-attachment, BroadcastChannel, cascade layers, containment, <dialog>, :focus-visible, hwb(), dynamic-range media query, prefers-contrast media query, Streams API, contain: style, Individual transforms, Array findLast() and findLastIndex(), :modal pseudo-class, Permissions API, overscroll-behavior, overflow property, offset property, forced-colors media query, Inline-size containment, ic unit, grid animation, font-palette CSS property, small, large, and dynamic viewport units, WebGL OES_draw_buffers_indexed extension, MathML, :autofill pseudo, container queries, color-gamut media query, CSS trigonometric functions, font-variant-alternates, translate attribute, OPFS, Wasm SIMD, Imperative slot assignment, serverTiming, Screen orientation, Resource size, requestAnimationFrame() in workers, Push messages, CSS outline shorthand, OffscreenCanvas API, messageerror, media query range syntax, input (event), Import maps, Form-associated custom elements, font-synthesis properties, Fetch metadata, Constructed stylesheets, inert attribute, Canvas roundRect(), Canvas createConicGradient(), WebRTC SCTP information, New CSS color spaces, nth-child() of <selector>, Compression Streams, CSS color-mix(), CSS color(), window.print(), JavaScript modules in workers, calc() constants, Array by copy, animation-composition property, Two-values of CSS Display, dirname attribute, CSS Grid subgrid, font-variant-position Update frequency media query, Overflow media queries, <link rel="modulepreload">, image-set(), CSS hyphens property, Device orientation events, CSS @counter-style, contain-intrinsic-style, HTML <search> element, Web Authn easy public key access, isWellFormed() and toWellFormed(), ARIA attribute reflection, :user-valid and :user-invalid, clip-path boxes, User activation API, lh unit, Storage access, URL.canParse(), Scripting media query, CSS mask property, CSS exponential functions, :dir pseudo-class, background-clip: text, preservesPitch property, rel="preload" for responsive images, CSS Nesting, CSS linear() easing, counter-set, CSS cap unit, Canvas reset(), loading="lazy" for and <iframe>, :has(), Array.fromAsync, AVIF, Declarative shadow DOM, Transferable ArrayBuffer, Promise.withResolvers(), checkVisibility(), Array grouping, white-space-collapse, text-wrap: nowrap, AbortSignal.any(), CSS transform-box, Intl.Segmenter, Vertical form controls, text-wrap: balance, light-dark(), zoom, :state(), round(), mod(), rem(), Set methods, Gradient interpolation, Async clipboard API, Resizable ArrayBuffers, CSS Relative colors, @property, Unsanitized HTML parsing methods, alt text for generated content, font-size-adjust, transition-behavior, mutually exclusive details elements, content-visibility, Canvas willReadFrequently, backdrop-filter





Newly interoperable features for the web platform 2022-2024

Interoperable

Features available in the major browser engines.

The Interop project

Working together to make the web more interoperable.

Active Focus Areas					INTEROP
Accessibility	95.6%	95.6%	95.6%	97.3%	95.4%
CSS Nesting	97.4%	97.4%	97.4%	66.5%	66.5%
Custom Properties	98.4%	98.4%	99.0%	98.1%	97.6%
Declarative Shadow DOM	100%	100%	100%	100%	100%
font-size-adjust	100%	100%	100%	100%	100%
HTTPS URLs for WebSocket	92.8%	92.8%	100%	100%	92.8%
IndexedDB	98.8%	99.2%	99.9%	95.7%	93.9%
Layout	99.8%	99.7%	95.4%	92.7%	90.0%
Pointer and Mouse Events	93.5%	89.3%	91.9%	55.3%	52.0%
Popover	100%	100%	100%	97.8%	97.8%
Relative Color Syntax	14.8%	14.8%	14.9%	99.6%	14.4%
requestVideoFrameCallback	100%	100%	100%	86.1%	86.1%
Scrollbar Styling	100%	100%	100%	48.2%	48.2%
@starting-style & transition-behavior	93.0%	97.2%	95.8%	98.6%	88.8%
Text Directionality	99.0%	99.0%	99.8%	94.4%	94.4%
text-wrap: balance	100%	100%	100%	100%	100%
URL	95.5%	85.5%	89.6%	99.7%	85.3%
TOTAL	92.8%	92.2%	92.9%	90.0%	82.5%

<https://wpt.fyi/interop-2024>

Interop 2024 Dashboard

STABLE

EXPERIMENTAL

83

INTEROP

36

INVESTIGATIONS

93



Chrome

92



Edge

93



Firefox

90



Safari

Interop 2024 stable scores as of 13 Nov 2024.

cascade layers, **containment**, `<dialog>`, **hwb()**, `contain: style`, **individual transforms**, `overscroll-behavior`, **offset property**, `inline-size containment`, **font-palette CSS property**, `container queries`, **translate attribute**, OffscreenCanvas API, **form-associated custom elements**, constructed Stylesheets, **inert attribute**, new CSS color spaces, **nth-child()** of `<selector>`, CSS grid subgrid, **:user-valid** and **:user-invalid**, CSS mask property, **CSS nesting**, `:has()`, **Declarative Shadow DOM**, `text-wrap: balance`, **round()**, **mod()**, **rem()**, CSS relative colors, **@property**, `font-size-adjust`, `transition-behavior`

Some of the features helped to interoperability by Interop

How do I know when a feature is **ready to use?**

Is it in all browsers? How long should I wait to use it once it's in browsers?

This is **not a new problem**

Rapid release cycles have made it harder. In the past we often tied our version of the web to which version of Internet Explorer things worked in.



Baseline

<https://web-platform-dx.github.io/web-features/>



Newly available

Supported in:

- Apple Safari (macOS and iOS)
- Google Chrome (desktop and Android)
- Microsoft Edge (desktop)
- Mozilla Firefox (desktop and Android)



Widely available

A feature is classed as **Widely available** once 30 months have passed since it became Newly available.



Limited availability

A feature is classed as **Limited availability** if it is unavailable in one or more of the core browsers.

appearance, Array at(), ::backdrop, background-attachment, BroadcastChannel, cascade layers, containment, <dialog>, :focus-visible, hwb(), dynamic-range media query

prefers-contrast media query, Streams API, contain: style, Individual transforms, Array findLast() and findLastIndex(), :modal pseudo-class, Permissions API, overscroll-behavior, overflow property, offset property, forced-colors media query, Inline-size containment, ic unit, Grid animation, font-palette CSS property, Small, large, and dynamic viewport units, WebGL OES_draw_buffers_indexed extension, MathML, :autofill pseudo, Container queries, color-gamut media query, CSS trigonometric functions, font-variant-alternates, translate attribute, OPFS, Wasm SIMD, Imperative slot assignment, serverTiming, Screen orientation, Resource size, requestAnimationFrame() in workers, Push messages, CSS outline shorthand, OffscreenCanvas API, messageerror, Media query range syntax, input (event), Import maps, Form-associated custom elements, font-synthesis properties, Fetch metadata, Constructed stylesheets, Inert attribute, Canvas roundRect(), Canvas createConicGradient(), WebRTC SCTP information, New CSS color spaces, nth-child() of <selector>, Compression Streams, CSS color-mix(), CSS color(), window.print(), JavaScript modules in workers, calc() constants, Array by copy, animation-composition property, Two-values of CSS Display, dirname attribute, CSS Grid subgrid, font-variant-position Update frequency media query, Overflow media queries, <link rel="modulepreload">, image-set(), CSS hyphens property, Device orientation events, CSS @counter-style, contain-intrinsic-style, HTML <search> element, Web Authn easy public key access, isWellFormed() and toWellFormed(), ARIA attribute reflection, :user-valid and :user-invalid, clip-path boxes, User activation API, lh unit, Storage access, URL.canParse(), Scripting media query, CSS mask property, CSS exponential functions, :dir pseudo-class, background-clip: text, preservesPitch property, rel="preload" for responsive images, CSS Nesting, CSS linear() easing, counter-set, CSS cap unit, Canvas reset(), loading="lazy" for and <iframe>, :has(), Array.fromAsync, AVIF, Declarative shadow DOM, Transferable ArrayBuffer, Promise.withResolvers(), checkVisibility(), Array grouping, white-space-collapse, text-wrap: nowrap, AbortSignal.any(), CSS transform-box, Intl.Segmenter, Vertical form controls, text-wrap: balance, light-dark(), zoom, :state(), round(), mod(), rem(), Set methods, Gradient interpolation, Async clipboard API, Resizable ArrayBuffers, CSS Relative colors, @property, Unsanitized HTML parsing methods, Alt text for generated content, font-size-adjust, transition-behavior, Mutually exclusive details elements, content-visibility, Canvas willReadFrequently, backdrop-filter

Baseline Newly available 2022-2024

**The mission of the WebDX
Community Group is to facilitate
coordinated approaches to improve
the overall experience of
developing for the web platform.**

WebDX Community Group Charter
<https://github.com/web-platform-dx/admin/blob/main/charter.md>

Baseline <year>

For the first time we can view an annual release of the web platform.

appearance, Array at(), ::backdrop, background-attachment, BroadcastChannel, cascade layers, containment, <dialog>, :focus-visible, hwb(), dynamic-range media query, prefers-contrast media query, Streams API, contain: style, Individual transforms, Array findLast() and findLastIndex(), :modal pseudo-class, Permissions API, overscroll-behavior, overflow property, offset property, forced-colors media query, Inline-size containment, ic unit, grid animation, font-palette CSS property, small, large, and dynamic viewport units, WebGL OES_draw_buffers_indexed extension,

MathML, :autofill pseudo, Container queries, color-gamut media query, CSS trigonometric functions, font-variant-alternates, translate attribute, OPFS, Wasm SIMD, Imperative slot assignment, serverTiming, Screen orientation, Resource size, requestAnimationFrame() in workers, Push messages, CSS outline shorthand, OffscreenCanvas API, messageerror, Media query range syntax, input (event), Import maps, Form-associated custom elements, font-synthesis properties, Fetch metadata, Constructed stylesheets, Inert attribute, Canvas roundRect(), Canvas createConicGradient(), WebRTC SCTP information, New CSS color spaces, nth-child() of <selector>, Compression Streams, CSS color-mix(), CSS color(), window.print(), JavaScript modules in workers, calc() constants, Array by copy, animation-composition property, Two-values of CSS Display, dirname attribute, CSS Grid subgrid, font-variant-position Update frequency media query, Overflow media queries, <link rel="modulepreload">, image-set(), CSS hyphens property, Device orientation events, CSS @counter-style, contain-intrinsic-style, HTML <search> element, Web Authn easy public key access, isWellFormed() and toWellFormed(), ARIA attribute reflection, :user-valid and :user-invalid, clip-path boxes, User activation API, lh unit, Storage access, URL.canParse(), Scripting media query, CSS mask property, CSS exponential functions, :dir pseudo-class, background-clip: text, preservesPitch property, rel="preload" for responsive images, CSS Nesting, CSS linear() easing, counter-set, CSS cap unit, Canvas reset(), loading="lazy" for and <iframe>, :has()

Array.fromAsync, AVIF, Declarative shadow DOM, Transferable ArrayBuffer, Promise.withResolvers(), checkVisibility(), Popover, Array grouping, white-space-collapse, text-wrap: nowrap, AbortSignal.any(), CSS transform-box, Intl.Segmenter, Vertical form controls, text-wrap: balance, light-dark(), zoom, :state(), round(), mod(), rem(), Set methods, gradient interpolation, Async clipboard API, Resizable ArrayBuffers, CSS Relative colors, @property, Unsanitized HTML parsing methods, Alt text for generated content, font-size-adjust, transition-behavior, mutually exclusive details elements, content-visibility, Canvas willReadFrequently, backdrop-filter



align-content for block layout

Vertically center an item without using flex or grid

```
.center {  
  display: flex;  
  align-content: center;  
  block-size: 300px;  
}
```



```
.center {  
  align-content: center;  
  block-size: 300px;  
}
```

I am centered vertically



@property

Define type, default values, and inheritance for CSS custom properties

```
@property --bgcolor {  
  syntax: "<color>";  
  inherits: false;  
  initial-value: #ee4b6a;  
}
```

```
.box {  
  background-color: var(--bgcolor);  
}
```

This box uses the initial value for --bgcolor.

```
.box2 {  
  --bgcolor: #0f7173;  
  background-color: var(--bgcolor);  
}
```

This box resets the value of
--bgcolor.

```
.box3 {  
  --bgcolor: 1;  
  background-color: var(--bgcolor);  
}
```

This box tries to use an integer as the value for `--bgcolor`.

@property: Next-gen CSS val x +

https://web.dev/blog/at-property-baseline

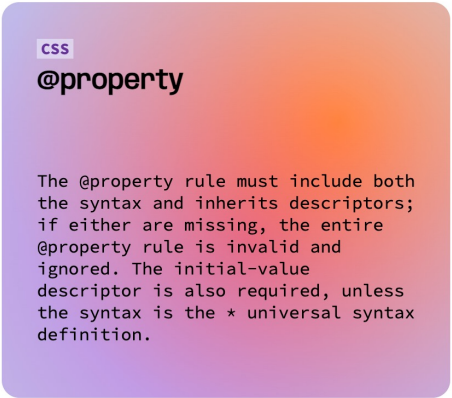
web.dev About HTML CSS JavaScript Blog Learn Explore More Search English Sign in

Demo: Twinkling gradient background

One of the neat applications of `@property` is the smooth animation of properties that were previously impossible to transition, like gradients, which are perceived as images by the browser. However, if you give variables syntactic meaning via `@property`, these can be used in a gradient statement. Now you are describing an animation between two declared values within the gradient, enabling animation. Take the following example: a card with a subtle background animation that consists of two radial gradients which shift color on different timelines:

CSS @property animated bg card

HTML CSS Result EDIT ON CODEPEN



CSS

@property

The `@property` rule must include both the syntax and inherits descriptors; if either are missing, the entire `@property` rule is invalid and ignored. The initial-value descriptor is also required, unless the syntax is the `*` universal syntax definition.

Resources 1x 0.5x 0.25x Rerun

Using @property to style animate colors in a background gradient

On this page

- The benefits of `@property`
- Create advanced custom properties
- Demo: Twinkling gradient background
- Conclusion
- Additional Reading

https://web.dev/blog/at-property-baseline



Size container queries

Query the size of the container instead of the viewport.



G-CIND at the Bristol Balloon Fiesta 2022

After two years the Bristol Balloon Fiesta was back. Most of Bristol could be found trekking up the hill at 5.30am to worship their strange hot-air demanding gods.

<https://codepen.io/web-dot-dev/pen/dyxXxjV>



G-CIND at the Bristol Balloon Fiesta 2022

After two years the Bristol Balloon Fiesta was back. Most of Bristol could be found trekking up the hill at 5.30am to worship their strange hot-air demanding gods.

```
.wrapper {
  container-type: inline-size;
}

@container (min-width: 500px) {
  .card {
    display: grid;
    grid-template-columns: 0.8fr 1fr;
    gap: 10px;
  }
}
```



:has()

A parent selector—and much more



G-CIND at the Bristol Balloon Fiesta 2022

After two years the Bristol Balloon Fiesta was back. Most of Bristol could be found trekking up the hill at 5.30am to worship their strange hot-air demanding gods.

After two years the Bristol Balloon Fiesta was back. Most of Bristol could be found trekking up the hill at 5.30am to worship their strange hot-air demanding gods.

```
@container (min-width: 500px) {  
  .card:has(figure + .content) {  
    display: grid;  
    grid-template-columns: 0.8fr 1fr;  
    gap: 10px;  
  }  
}
```



G-CIND at the Bristol Balloon
Fiesta 2022

After two years the Bristol Balloon Fiesta was back. Most of Bristol could be found trekking up the hill at 5.30am to worship their strange hot-air demanding gods.

After two years the Bristol Balloon Fiesta was back. Most of Bristol could be found trekking up the hill at 5.30am to worship their strange hot-air demanding gods.



text-wrap: balance

Making headlines more beautiful

```
h2 {  
  text-wrap: balance;  
}
```

<https://codepen.io/web-dot-dev/pen/vYoXZeO>

Bell bok broccoli bush cabbage

The heading in this card is not balanced. We might end up with a single word on the last line.

Bell bok broccoli bush cabbage

The heading in this card is balanced. This means we avoid having a single word on the last line.



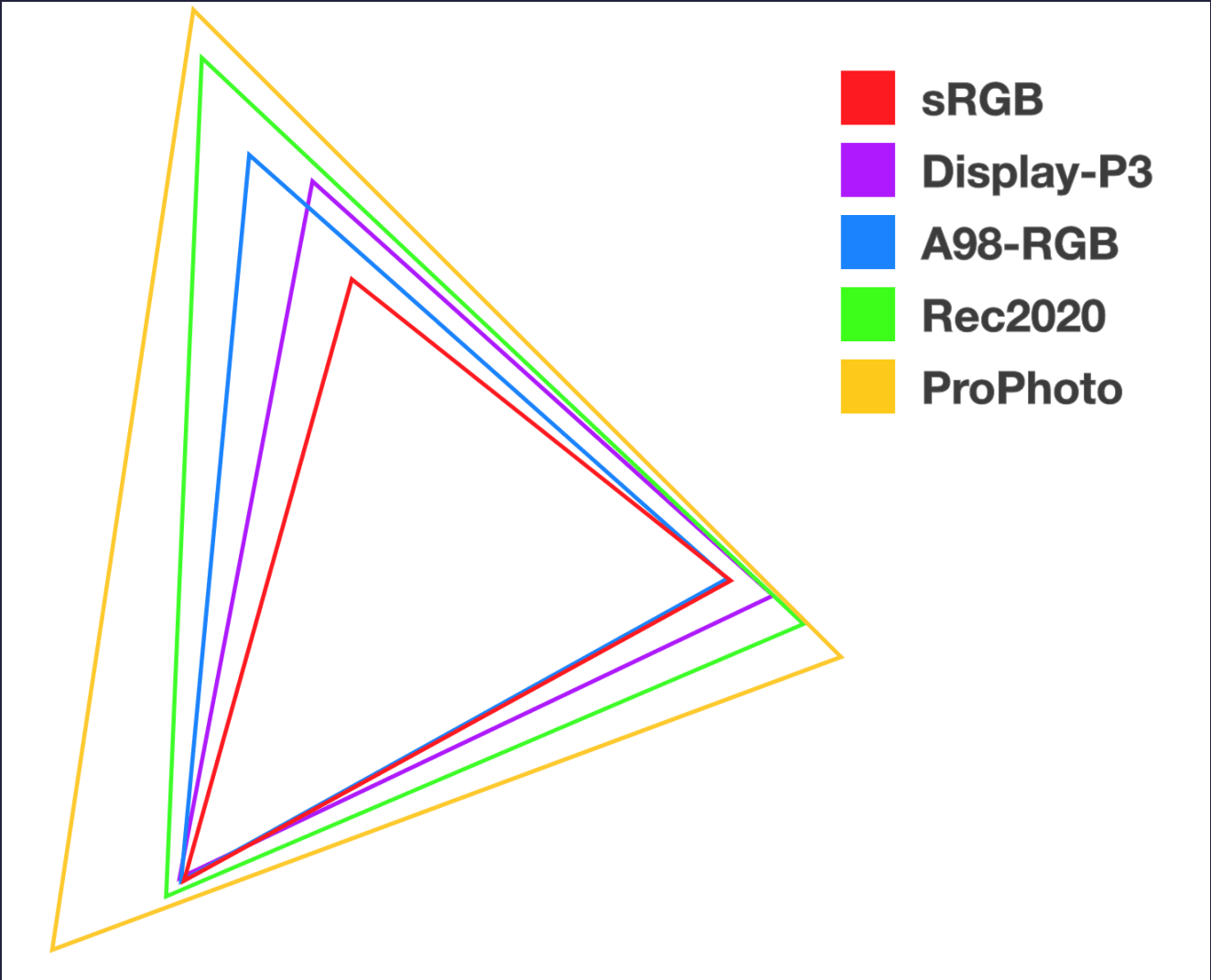
New color spaces

sRGB Linear, LCH, okLCH, LAB, okLAB, Display p3, Rec2020,
a98 RGB, ProPhoto RGB, XYZ, XYZ d50, XYZ d65

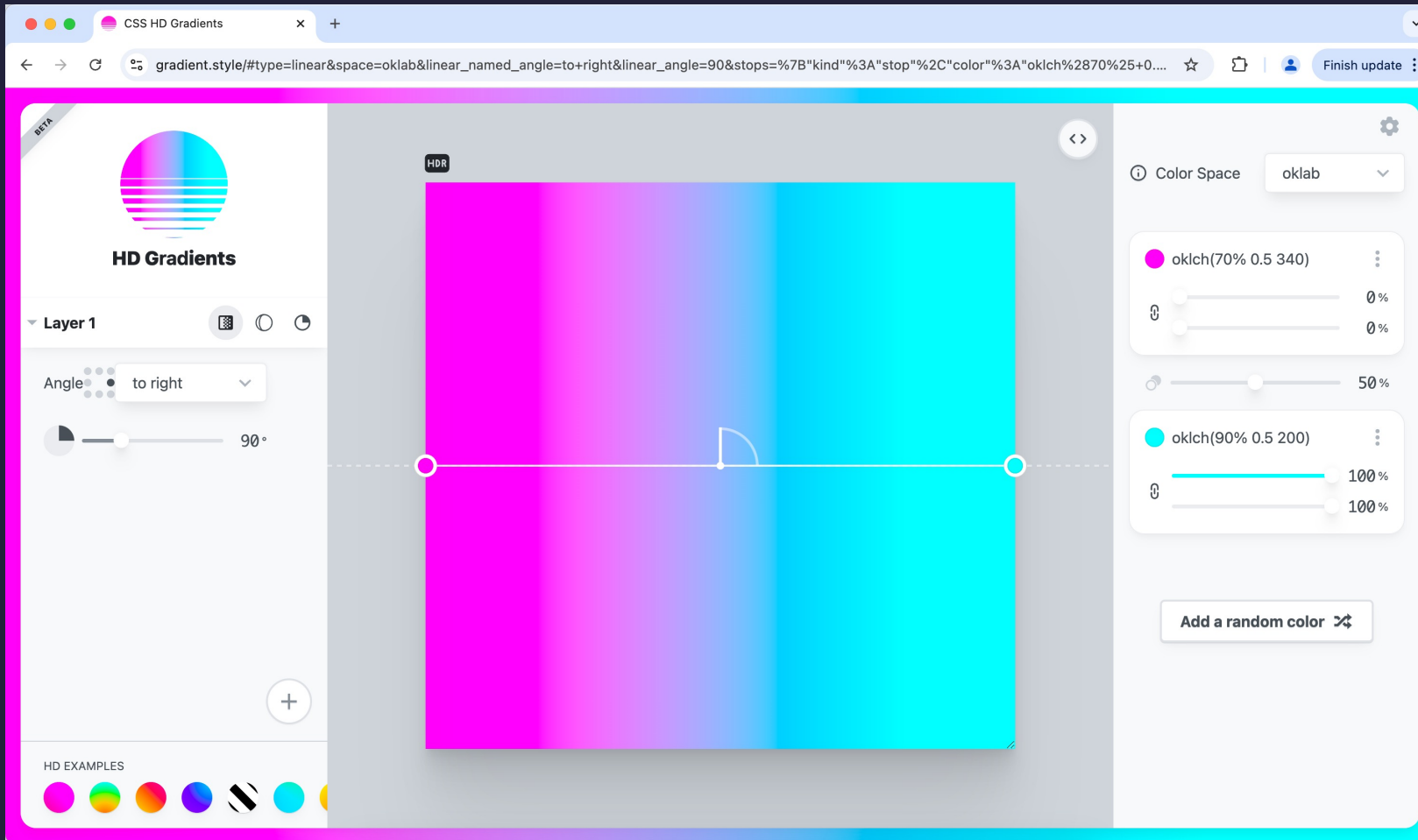


New color functions

`color()`, `lch()`, `lab()`, `oklch()`, `oklab()`



<https://developer.chrome.com/docs/css-ui/access-colors-spaces>



gradient.style



color-mix()

Mix two colors in CSS using any color space.


```
li {  
  background-color:  
    color-mix(in srgb, #34c9eb 10%, white);  
}
```

#34c9eb 10%

#34c9eb 40%

#34c9eb 70%

#34c9eb 90%

How do we know what's in Baseline?

Introducing web features: mapping the web platform as a series of features.

```
Code Blame 60 lines (60 loc) · 2.34 KB Raw Copy Download Edit View Source
```

```
1 name: Grid
2 description: CSS grid is a two-dimensional layout system, which lays content out in rows and columns.
3 spec: https://drafts.csswg.org/css-grid-3/
4 group: grid
5 caniuse: css-grid
6 status:
7   compute_from: css.properties.grid
8 compat_features:
9   - css.types.flex
10  - css.properties.align-content.grid_context
11  - css.properties.align-items.grid_context
12  - css.properties.align-self.grid_context
13  - css.properties.justify-content.grid_context
14  - css.properties.justify-items.grid_context
15  - css.properties.justify-self.grid_context
16  - css.properties.display.grid
17  - css.properties.display.inline-grid
18  - css.properties.grid
19  - css.properties.grid-area
20  - css.properties.grid-auto-flow
21  - css.properties.grid-auto-flow.column
22  - css.properties.grid-auto-flow.dense
23  - css.properties.grid-auto-flow.row
24  - css.properties.grid-column
25  - css.properties.grid-column-end
26  - css.properties.grid-column-start
```



grid-auto-flow - CSS: Cascad x +

developer.mozilla.org/en-US/docs/Web/CSS/grid-auto-flow#browser_compatibility

mdn web docs | References | Guides | Plus | Curriculum | Blog | Tools | Theme | Log in | Sign up for free

References > CSS > grid-auto-flow | English (US)

Browser compatibility

[Report problems with this compatibility data on GitHub](#)

Filter

- grid-*
 - grid
 - grid-area
 - grid-auto-columns
 - grid-auto-flow**
 - grid-auto-rows
 - grid-column
 - grid-column-end
 - grid-column-start
 - grid-row
 - grid-row-end
 - grid-row-start
 - grid-template
 - grid-template-areas
 - grid-template-columns
 - grid-template-rows
 - hanging-punctuation

	🖥️					📱						
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
grid-auto-flow	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
column	57	16	52	44	10.1	57	52	43	10.3	6.0	57	10.3
dense	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
row	57	79	52	44	10.1	57	52	43	10.3	7.0	57	10.3

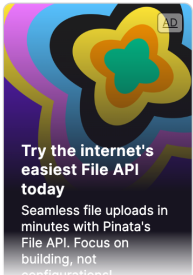
Tip: you can click/tap on a cell for more information.

- ✓ Full support

See also

In this article

- Try it
- Syntax
- Formal definition
- Formal syntax
- Examples
- Specifications
- Browser compatibility**
- See also



Try the internet's easiest File API today
Seamless file uploads in minutes with Pinata's File API. Focus on building, not configurations!

https://developer.mozilla.org/docs/Web/CSS/grid-auto-flow#browser_compatibility

grid-auto-flow - CSS: Cascad x +

developer.mozilla.org/en-US/docs/Web/CSS/grid-auto-flow#browser_compatibility






mdn web docs | References | Guides | Plus | Curriculum | Blog | Tools | Theme | Log in | Sign up for free

References > CSS > grid-auto-flow | English (US)

Filter

- grid-*
- grid
- grid-area
- grid-auto-columns
- grid-auto-flow**
- grid-auto-rows
- grid-column
- grid-column-end
- grid-column-start
- grid-row
- grid-row-end
- grid-row-start
- grid-template
- grid-template-areas
- grid-template-columns
- grid-template-rows
- hanging-punctuation

grid-auto-flow

 **Baseline** Widely available    

The **grid-auto-flow** CSS property controls how the auto-placement algorithm works, specifying exactly how auto-placed items get flowed into the grid.

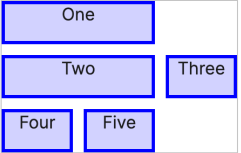
Try it

CSS Demo: grid-auto-flow RESET

```
grid-auto-flow: row;
```

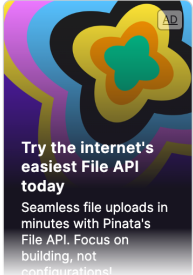
```
grid-auto-flow: column;
```

```
grid-auto-flow: row dense;
```



In this article

- Try it
- Syntax
- Formal definition
- Formal syntax
- Examples
- Specifications
- Browser compatibility
- See also



Try the internet's easiest File API today
Seamless file uploads in minutes with Pinata's File API. Focus on building, not configurations!

https://developer.mozilla.org/docs/Web/CSS/grid-auto-flow#browser_compatibility

Explain browser support to stakeholders

“Everything in <project> will be Baseline Widely available”

Browser support at Clearleft x +

browsersupport.clearleft.com

Clearleft

Browser Support

Features, not browsers

We don't base our browser support on specific browser names and numbers. Instead, our support policy is based on the capabilities of those browsers.

The [Baseline](#) initiative is a joint effort by Google, Microsoft, Apple, and Mozilla to categorise browser support for web standards. Baseline provides clear information about which web standards features are ready to use in websites. It designates new features into two categories:

1. **Newly available** – a feature is supported by the latest versions of all core browsers
2. **Widely available** – a feature has been supported across browsers for at least 30

<https://browsersupport.clearleft.com/>

Adopt a Baseline year

My shiny framework v.1 uses Baseline 2022, v.2 moves to Baseline 2023.

Use Baseline to see what's new

Find out what features have come to the platform, create a list to learn about.

Web features explorer

Release notes Newly available Widely available Limited availability All Groups

Widely available

These features have been supported across all browsers for at least 30 months, making them **Baseline widely available**. To learn more, see [Baseline](#).

appearance

Widely available

TAGS: CSS GROUPS: CSS

The appearance CSS property controls the appearance of form controls. Using appearance: none disables any default native appearance and allows the elements to be styled with CSS.

[Learn more](#)

Chrome 84 <small>Released on 2020-07-27</small>	Chrome Android 84 <small>Released on 2020-07-27</small>	Edge 84 <small>Released on 2020-07-16</small>	Firefox 80 <small>Released on 2020-08-25</small>	Firefox for Android 80 <small>Released on 2020-08-31</small>	Safari 15.4 <small>Released on 2022-03-14</small>	Safari on iOS 15.4 <small>Released on 2022-03-14</small>
---	---	---	--	--	---	--

Array at()

Widely available

TAGS: JAVASCRIPT GROUPS: JAVASCRIPT > COLLECTIONS > ARRAYS JAVASCRIPT > COLLECTIONS > TYPED ARRAYS

The at() method of arrays and typed arrays returns the item at an index, including negative indices for getting items relative to the end of an array. Also known as the relative indexing method.

[Learn more](#)

<https://web-platform-dx.github.io/web-features-explorer/>

Web Platform Status

webstatus.dev

Log in

Features

Baseline 2023

Report an issue

About

Features overview

523 features

Filter by ...

Columns

Export to CSV

Feature	Baseline	Chrome	Edge	Firefox	Safari
backdrop-filter	✓ Newly available	99.5%	95.9%	98.5%	6.0%
willReadFrequently	✓ Newly available	---	---	---	---
content-visibility	✓ Newly available	97.3%	96.1%	95.2%	93.8%
Mutually exclusive <details> elements	✓ Newly available	100%	100%	100%	100%
transition-behavior	✓ Newly available	100%	100%	100%	100%
font-size-adjust	✓ Newly available	94.1%	94.1%	94.1%	94.5%
Alt text for generated content	✓ Newly available	---	---	---	---
Growable SharedArrayBuffer	✓ Newly available	ⓘ	ⓘ	ⓘ	ⓘ
Unsanitized HTML parsing methods	✓ Newly available	---	---	---	---
Registered custom properties	✓ Newly available	---	---	---	---
Relative colors	✓ Newly available	---	---	---	---
Resizable ArrayBuffer	✓ Newly available	ⓘ	ⓘ	ⓘ	ⓘ
Resizable buffers	✓ Newly available	ⓘ	ⓘ	ⓘ	ⓘ

<https://webstatus.dev>

Show feature status in talks, articles, and blog posts

Use the Baseline status component on posts or logos on slides

web.dev

About HTML CSS JavaScript Blog Learn Explore More

Search / English Sign in

Home > Blog Was this helpful?

CSS font-size-adjust is now in Baseline

Rachel Andrew

Published: July 23, 2024

Celebration: This web feature is now available in all three major browser engines, and becomes **Baseline Newly Available** as of July 23, 2024.

font-size-adjust

Baseline 2024 NEWLY AVAILABLE

The CSS `font-size-adjust` property lands in Chrome today, and becomes part of Baseline Newly Available. This property can help prevent shifting of your layout when fallback fonts are loaded, and ensure legibility of fallback fonts at smaller font sizes. The `font-size-adjust` property is part of [Interop 2024](#), so this is another win for the effort to improve the interoperability of the web platform.

On this page

- The problem
- How font-size-adjust helps

<https://web.dev/blog/font-size-adjust>
















GitHub - web-platform-dx/ba x +

github.com/web-platform-dx/baseline-status

README Apache-2.0 license

<baseline-status> web component

A widget displaying Baseline status of a web feature based on <https://github.com/web-platform-dx/web-features> data

Array (initial support)	 Baseline Widely available	   	▼
Anchor positioning	 Limited availability	   	▼
animation-composition	 Baseline 2023 NEWLY AVAILABLE	   	▼

Example

Show Baseline status widget for [anchor-positioning](#):

Languages

- JavaScript 100.0%

<https://github.com/web-platform-dx/baseline-status>

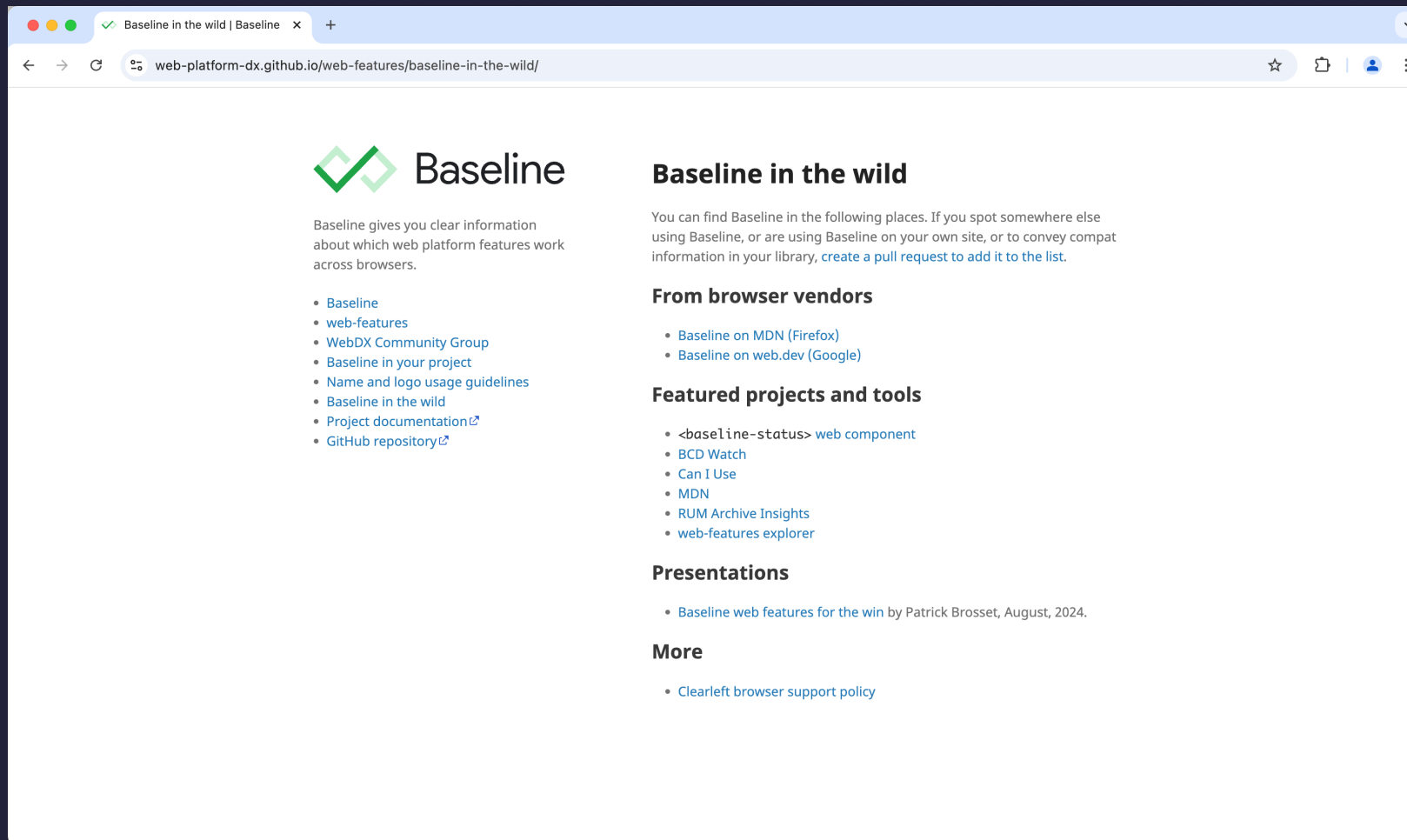
```
<baseline-status featureId="font-size-adjust"></baseline-status>
```

<https://github.com/web-platform-dx/baseline-status>



Use the Baseline logos!

<https://web-platform-dx.github.io/web-features/name-and-logo-usage-guidelines/>



<https://web-platform-dx.github.io/web-features/baseline-in-the-wild/>

Baseline in developer tools

Where can we use this data?

RUM Archive - RUM Insights x +


rumarchive.com/insights/#baseline

Baseline Support

The [Web Platform Baseline project](#) aims to track when exactly Web features became available in all major browsers (Chrome, Edge, Firefox, Safari). This is important since not all browsers implement all features at the same time; using cutting edge feature X on browser A might mean you leave some of your users on browser B out in the cold (or having to implement a costly fallback).

However, you also don't want to only use features from Baseline 2017, "just to be safe", because that would mean missing out on many already widely available features. In practice, you want to utilize features that **almost all of your users** will benefit from. To help you figure out which features those are and which Baseline year you should target, below we show how many global visitors observed by the RUM Archive **on the first Tuesday of August 2024** had support for ALL the features in a particular Baseline year.

Baseline support Desktop

 Baseline year	Supported by % of users of Tracked browsers [?]	Details
2017	99.97%	Click on a row in the table to get more information per Baseline year
2018	99.96%	
2019	99.43%	
2020	98.93%	
2021	98.61%	
2022	95.84%	
2023	85.18%	
2024	82.28%	

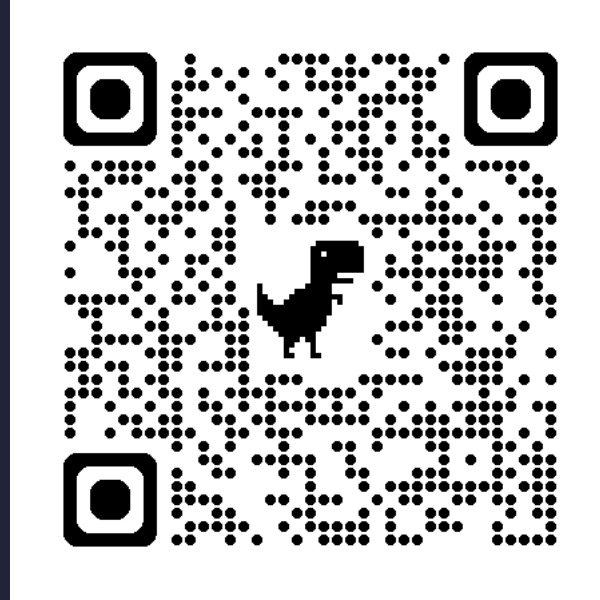
Baseline support Mobile

<https://rumarchive.com/insights/#baseline>

Get involved!

<https://web-platform-dx.github.io/web-features/>

Thank
you



Rachel Andrew

<https://noti.st/rachelandrew/PGuk8T/>