Modern CSS

Survey Presentation

Information Architecture and Web Usability
Winter Term 2025/26

Group 1 - Christian Burtscher, Jonas Glaser, Marcus Gugacs, Eva Haring

02 Dec 2025

Outline

- What is Modern CSS

- Features of Modern CSS:

  - Layouts

  - Properties

  - Pseudo-class

- Future Work

- References

  - @-Rules

  - Functions
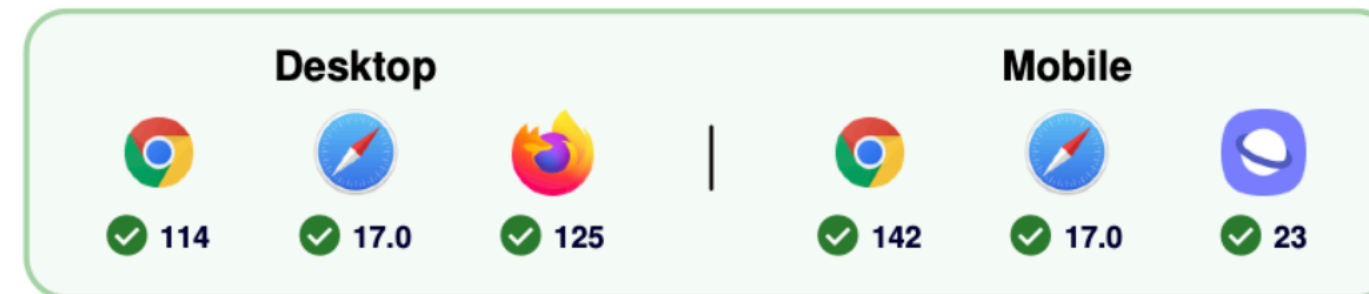
  - Other

What is Modern CSS

- Preprocessors like Sass emerged over 19 years ago.

  - Aim to add features missing in CSS.

- Frequent updates to CSS over past decade resulted in what we now know as Modern CSS. [1]

- Modern CSS added most features of preprocessors:

  - Including variables, nesting, functions, conditionals.

  - Important but missing for now are only mixins.

[1] https://beyond-the-cascade.com/do-we-still-need-css-preprocessors-in-2024/

Features of Modern CSS

Layouts: popover-open

- Baseline 2024

- Controlled by HTML or JS.

- Draw user's attention to specific information.

- Combination with CSS pseudo-class to change appearance of popover element.

Example Code

```css
[popover] {
  position: fixed;
  width: fit-content;
  height: fit-content;
  color: canvastext;
  background-color: canvas;
  inset: 0em;
  margin: auto;
  padding: 0.25em;
  overflow: auto;
}
:popover-open {
  background-color: hsl(25 100%
    50%);
  color: hsl(300 50% 3% / .85);
  padding: 1.5rem;
  width: 25ch;
}
```

2

Example Result

Say Hello

👋

1 https://caniuse.com/?search=popover-open

2 https://css-tricks.com/almanac/pseudo-selectors/p/popover-open/

Layouts: Masonry Layouts

- Experimental

- Dynamic layout method without JS.

- Automatic fill up of grid gaps.

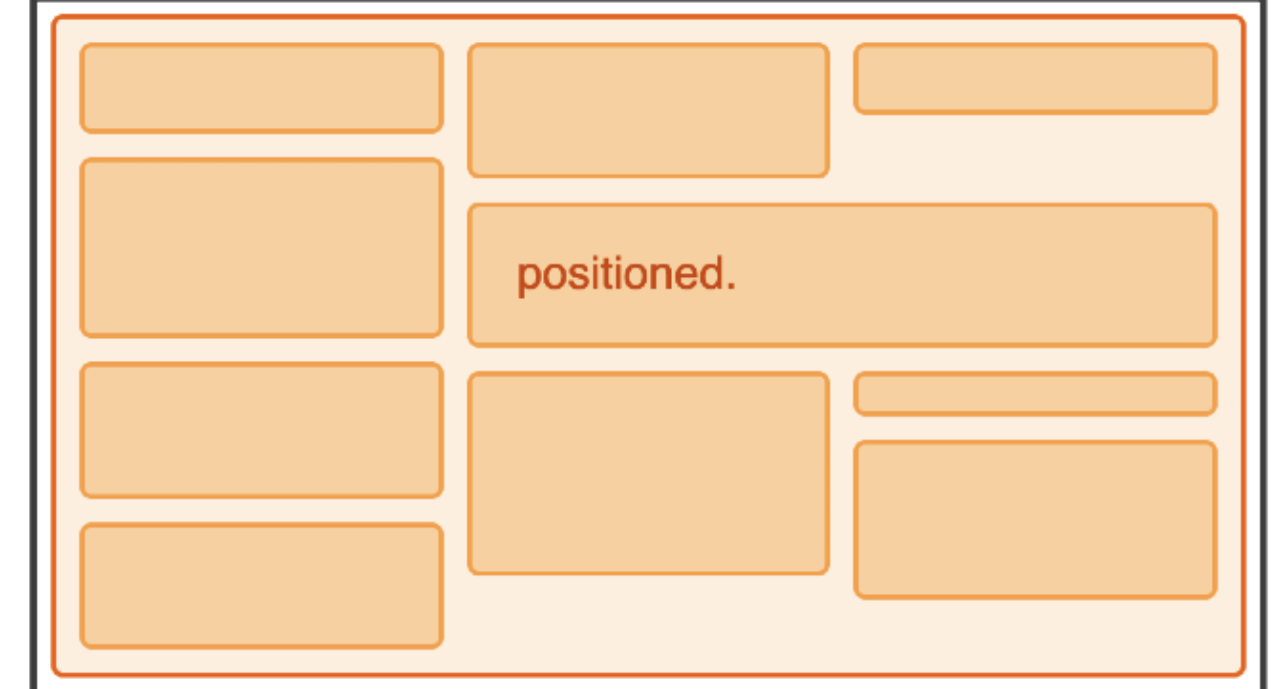- One strict axis and one masonry axis.

1

Example Code

```css
.grid {
  display: grid;
  gap: 0.65rem;
  grid-template-columns: repeat(
    auto-fill, minmax(7.5rem, 1fr)
    );
  grid-template-rows: masonry;
}

.positioned {
  padding: 1em;
  grid-column: 2 / 4;
}
```
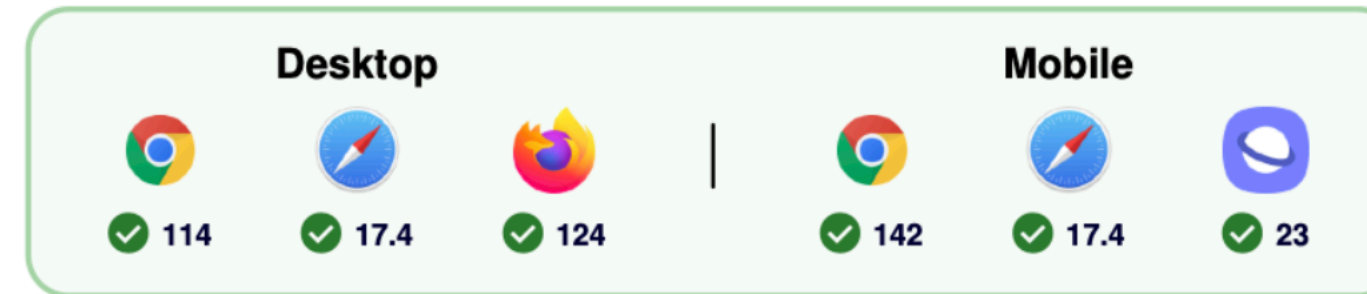
2

Example Result

positioned.

Properties: text-wrap

- Baseline 2024 [1]

- Offers multiple wrapping modes:
  `wrap`, `pretty`, `stable`, `balance`, `nowrap`.

- Simplifies inclusion of typographical content.

- Improves readability.

[1] https://caniuse.com/?search=text-wrap

[2] https://developer.mozilla.org/en-US/docs/Web/CSS/Reference/Properties/text-wrap

Example Code

```
.wrap {
    text-wrap: wrap;
}
.nowrap {
    text-wrap: nowrap;
}
.pretty {
    text-wrap: pretty;
}
.balance {
    text-wrap: balance;
}
h2 {
    font-size: 2rem;
    font-family: sans-serif;
}
```
[2]

Example Result

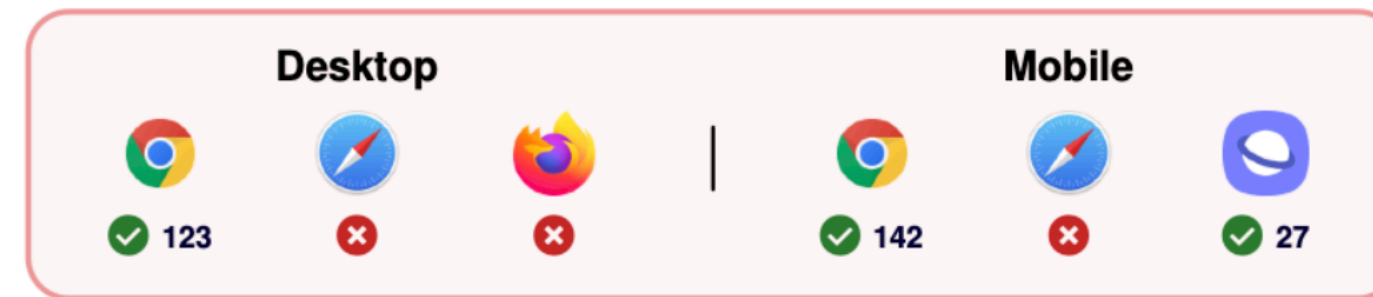**The default behavior; the text
in the heading wraps
"normally"**

In this case the text in the hea

**In this case the text in the
heading is nicely
pretty across the lines**

**In this case the text in
the heading is nicely
balanced across lines**

Properties: field-sizing

- Experimental

- Automatically resize form controls.

- Size is based on their content.

- Grow or shrink respectively.

Example Code

```
input,
textarea {
  field-sizing: content;
  min-width: 3.25rem;
  max-width: 22rem;
}

label {
  width: 9.50rem;
  margin-right: 1.25rem;
  text-align: right;
}
```

2

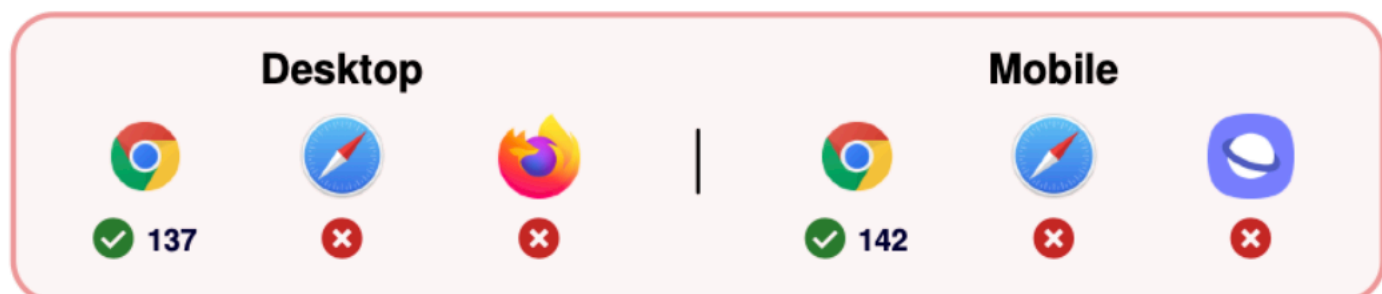**Example Result**

Enter name:

Enter email: e.g. a@b.com

Enter comment: crazy

1

[1] https://caniuse.com/?search=field-sizing

[2] https://developer.mozilla.org/en-US/docs/Web/CSS/Reference/Properties/field-sizing

Properties: reading-flow

- Experimental

- Accessibility feature.

- Control logical order of elements within layout.

- Specify navigation of assistive technologies when visual order differs from DOM order.

Example Code

```
.wrapper {
  display: grid;
  grid-template-columns: repeat
    (4, 1fr);
  grid-auto-rows: 6.25rem;
}
.wrapper a:nth-child(2) {
  grid-column: 3;
  grid-row: 2 / 4;
}
.wrapper a:nth-child(5) {
  grid-column: 1 / 3;
  grid-row: 1 / 3;
}
.wrapper {
  reading-flow: grid-rows;
}
```
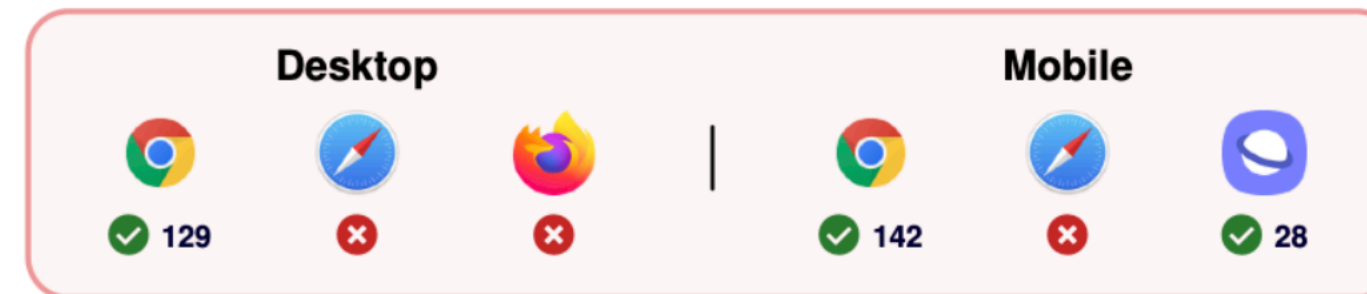
Example Result

[1] https://caniuse.com/?search=reading-flow

[2] https://developer.chrome.com/blog/reading-flow

Properties: interpolate-size

- Experimental

- Same as function calc-size(auto, …).

- Known as "Animate to Auto".

- Animate elements between fixed size values and intrinsic sizes.

Desktop

Chrome ✓ 129   Safari ✗   Firefox ✗ | Chrome ✓ 142   Safari ✗   ✓ 28

Mobile

1

Example Code

```
.demo-1 {
  interpolate-size:
    allow-keywords;

  .content {
    &.expanded {
      height: auto;
    }
  }
}
```

2

Example Result

Animate height to auto

`interpolate-size: allow-keywords;`

Toggle

Lorem ipsum dolor sit amet
consectetur adipisicing elit. Sint a

Animate height to auto

`calc-size(auto, size)`

¹ https://caniuse.com/?search=interpolate-size%3A+allow-keywords
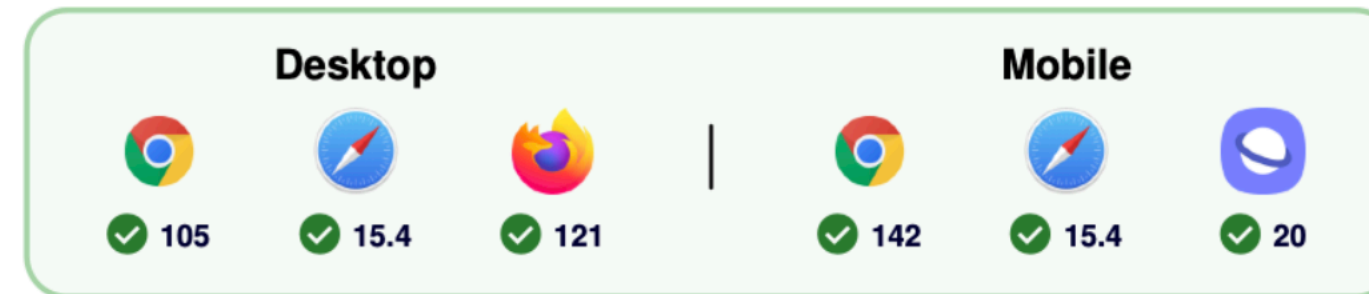
² https://frontendmasters.com/blog/what-you-need-to-know-about-modern-css-2025-edition/

Pseudo-class: has()

- Baseline 2023 [1]

- Select elements based on descendant elements.

- Apply styling rules to parent or ancestor element.

- Is able to select upwards in the DOM tree.

Example Code

```css
button:has(.icon) {
  display: flex;
  gap: 0.5rem;
}

nav li:has(ul) > a::after {
  content: "+";
  margin-inline: 0.5rem;
}
```
[2]

Example Result

[1] https://caniuse.com/?search=has%28%29

[2] https://bejamas.com/hub/guides/learn-css-has-selector-by-examples-top-use-cases
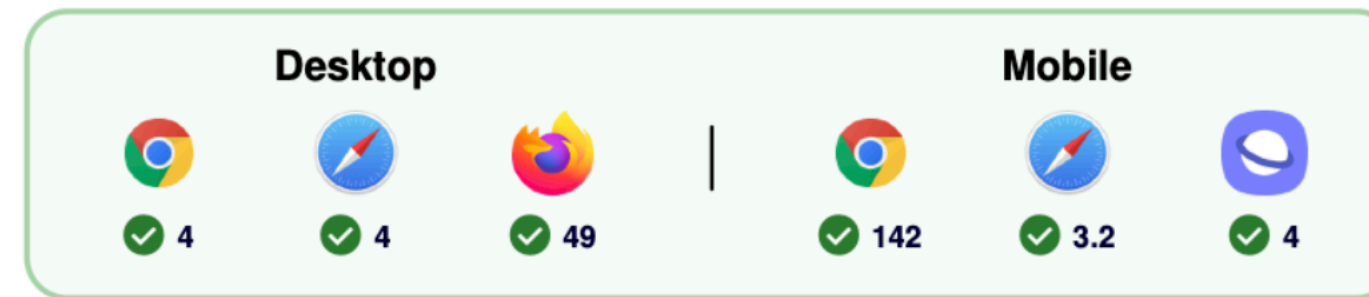
@-Rules: @supports

- Baseline 2015

¹

- Test browser feature support.

- Applies styles conditionally depending on support.

- Enhances progressive enhancement.

Example Code

```
.flex-container > * {
  padding: 0.3em;
  list-style-type: none;
  text-shadow: 0 0 0.125rem red;
  float: left;
}

@supports (display: flex) {
  .flex-container > * {
    text-shadow: 0 0 0.125rem
    blue;
    float: none;
  }

  .flex-container {
    display: flex;
  }
}
```
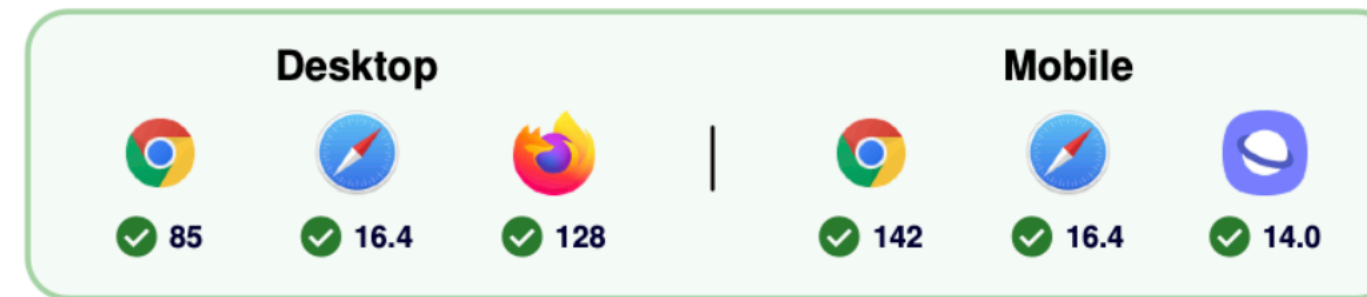
²

¹ https://caniuse.com/?search=%40supports

² https://developer.mozilla.org/en-US/docs/Web/CSS/Reference/At-rules/@supports

@-Rules: @property

- Baseline 2024 **1**

- Custom properties without JS.

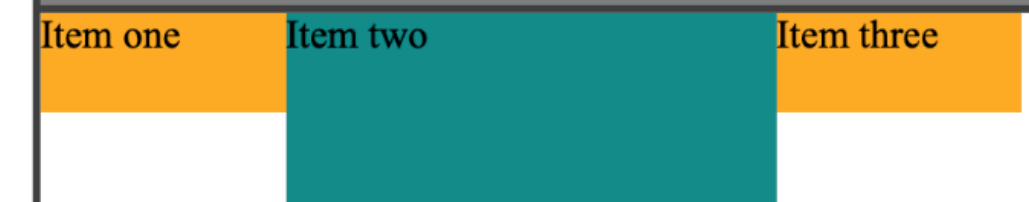- Supports typed variables, default values, and inheritance.

```
Example Code

@property --item-size {
  syntax: "<percentage>";
  inherits: true;
  initial-value: 40%;
}
.container {
  /* ... parent property value */
  --item-size: 20%;
}
.two {
  --item-size: initial;
  background-color: teal;
}
.three {
  /* invalid value */
  --item-size: 62rem;
}
```
2

Example Result

Item one   Item two   Item three

1 https://caniuse.com/?search=%40property

2 https://developer.mozilla.org/en-US/docs/Web/CSS/Reference/At-rules/@property
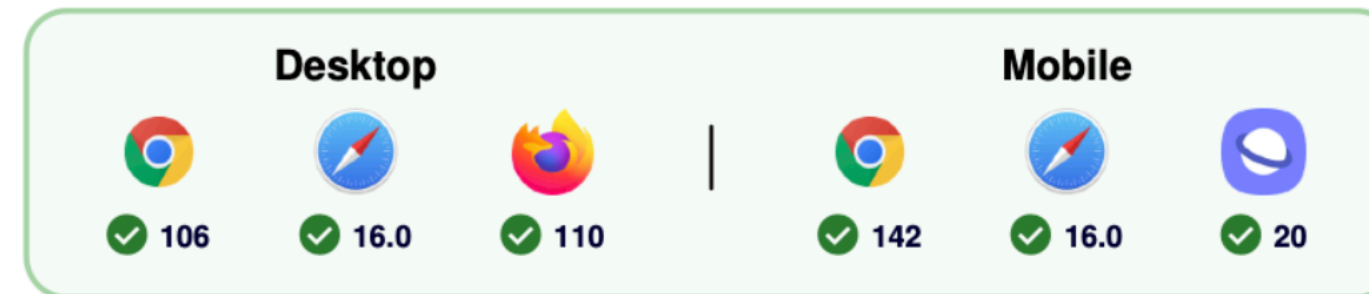
@-Rules: Container Size Queries

- Baseline 2025 [1]

- Size-based adjustment.

- Apply styles based on the size of a container.

- Conditional styles that activate when a container meets specific characteristics.
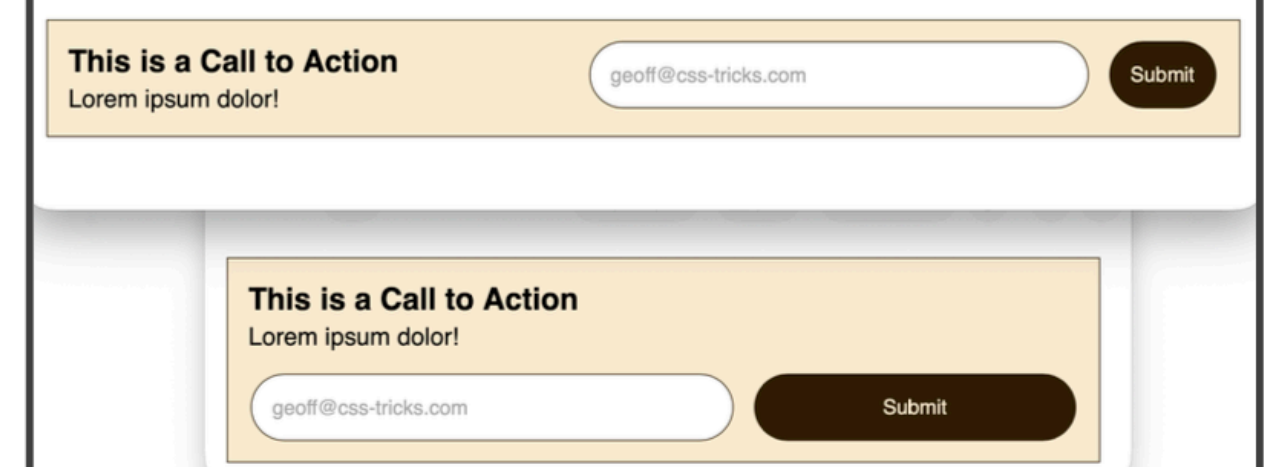
Example Code

```
container cta (width > 50rem) {
  .cta-layout {
    justify-content:
    space-between;
  }
}

@container cta (width < 50rem) {
  .cta-layout {
    flex-direction: column;
  }

  .cta-layout input,
  .cta-layout button {
    flex: 1 1 auto;
  }
}
```
[2]

Example Result

This is a Call to Action
Lorem ipsum dolor!
geoff@css-tricks.com   Submit

This is a Call to Action
Lorem ipsum dolor!
geoff@css-tricks.com   Submit

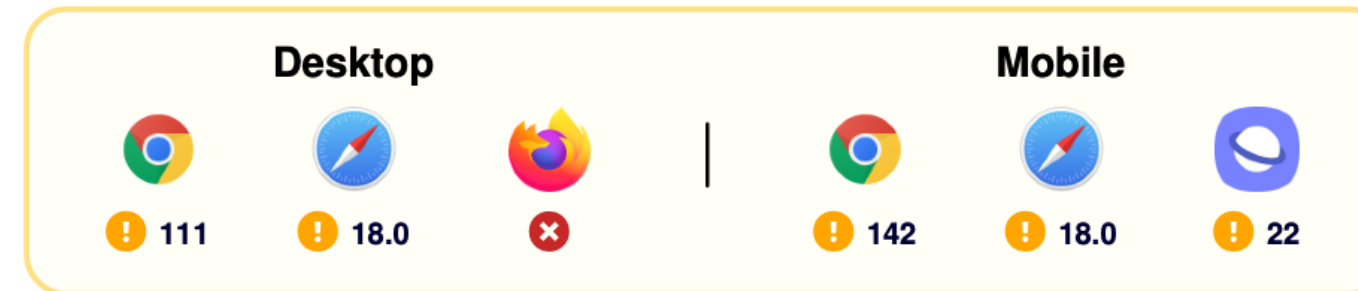[1] https://caniuse.com/?search=container+size+queries

[2] https://css-tricks.com/css-container-queries/#container-queries-properties-amp-values

@-Rules: Container Style Queries

- Experimental

- Style-based adjustment.

- Creates context-sensitive components.

- Allows relations to characteristics like background-color.

Desktop / **Mobile** [1]

Example Code

```
@container style(--theme: green)
    or style(--theme: blue) or
    style(--theme: red) {
  output {
    color: var(--theme);
  }
}

@container style(--theme: red) {
  output {
    font-weight: bold;
  }
}
```
[2]

Example Result

Change the value of --theme

1. ○ --theme: red;
2. ○ --theme: green
3. ○ --theme: blue
4. ○ Other color: [currentColor]

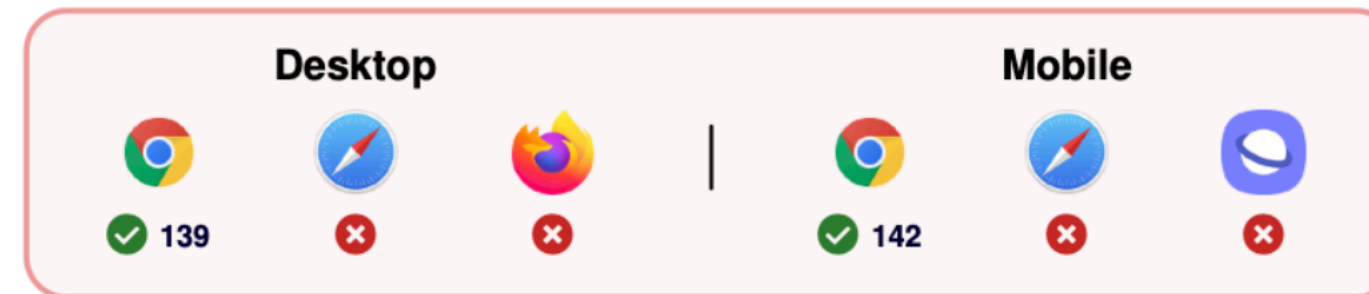**I change colors**

[1] https://caniuse.com/?search=container+style+queries

[2] https://developer.mozilla.org/en-US/docs/Web/CSS/Guides/Containment/Container_size_and_style_queries

@-Rules: @function

- Experimental

- Define custom, reusable functions.

- Perform calculations or logic.

- Return single value.

- Can be used in any CSS property.

Example Code

```css
@function --double(--value) {
  result: calc(var(--value) * 2);
}

p {
  --base-spacing: 0.625rem;
  border-radius: var(
    --base-spacing);
  padding: --double(var(
    --base-spacing));
  width: 50%;
  background-color: wheat;
}
```
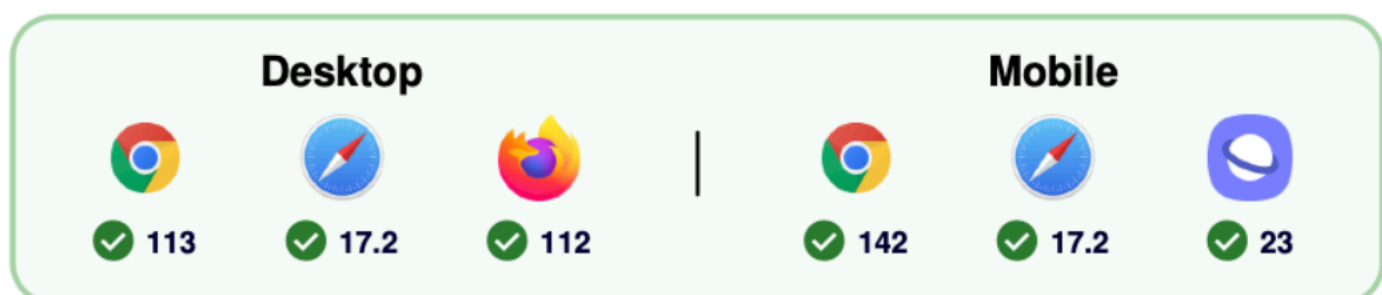
2

Example Result

Some content

[1] https://caniuse.com/?search=%40function

[2] https://developer.mozilla.org/en-US/docs/Web/CSS/Reference/At-rules/@function

Functions: linear()

- Baseline 2023

 1

- Used to control timing of CSS animations and transitions.

- Creates a transition curve.

- Allows definition of multiple intermediate points.
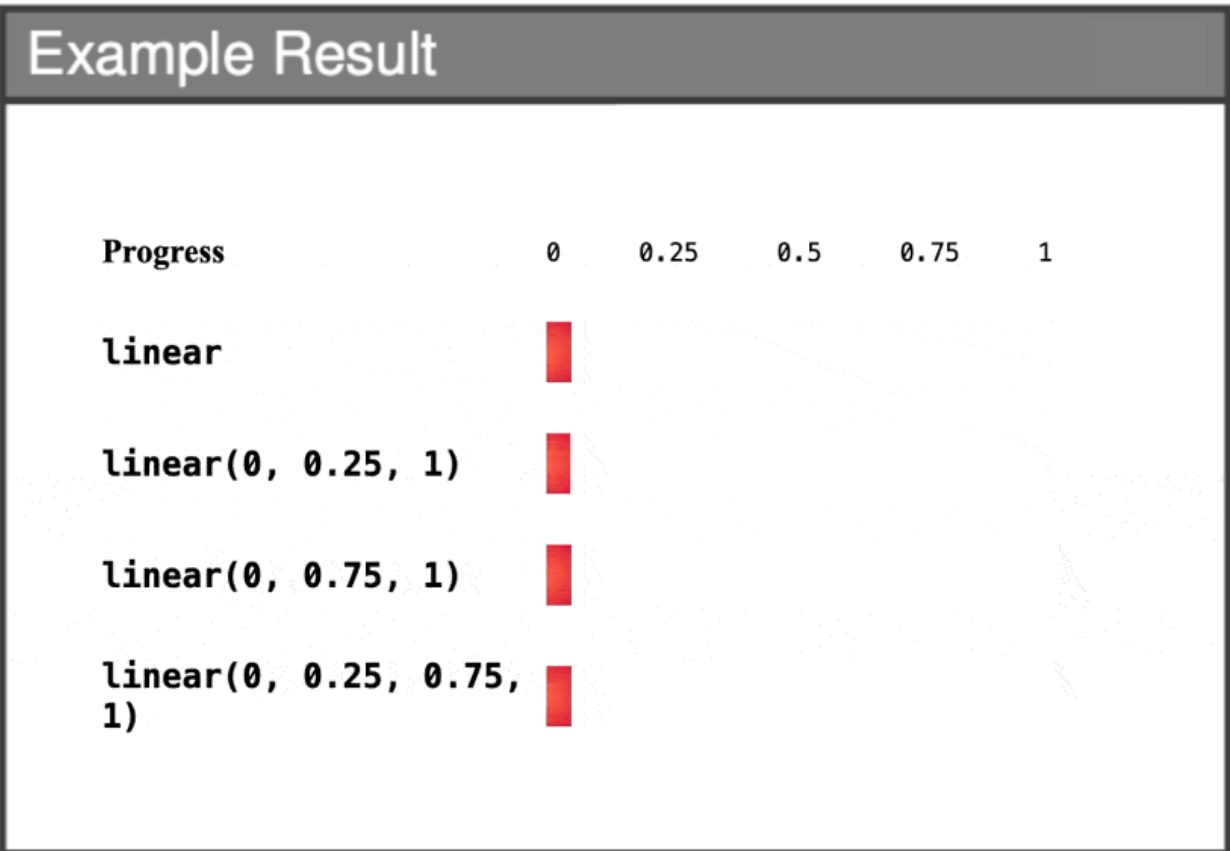

Example Code

```
.bar-1 {
  animation-timing-function:
    linear(0, 0.25, 1);
}

.bar-2 {
  animation-timing-function:
    linear(0, 0.75, 1);
}

.bar-3 {
  animation-timing-function:
    linear(0, 0.25, 0.75, 1);
}
```
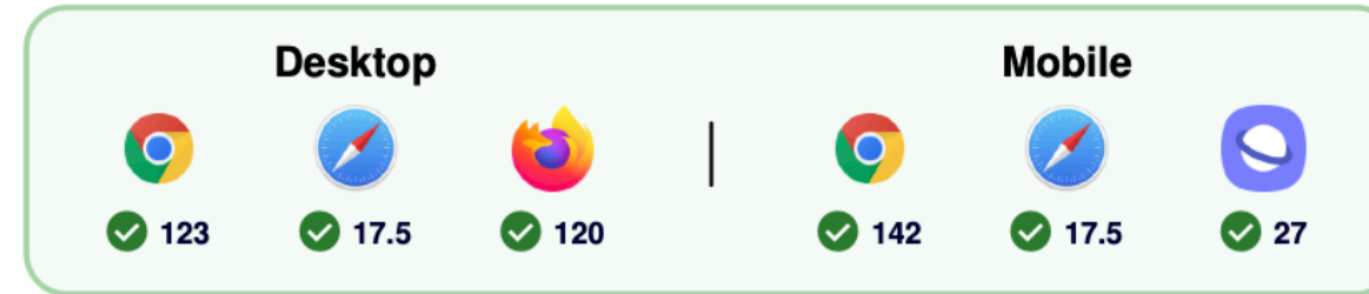**2**

Example Result

Progress	0	0.25	0.5	0.75	1
linear					
linear(0, 0.25, 1)					
linear(0, 0.75, 1)					
linear(0, 0.25, 0.75, 1)					

1 https://caniuse.com/?search=linear%28%29

2 https://css-tricks.com/almanac/functions/l/linear/

Functions: light-dark()

- Baseline 2024

- Change between light and dark modes.

- Properties can be assigned 2 different colors.

- Reduces effort for developers.

Example Code

```
:root {
  color-scheme: light dark;
  --light-bg: ghostwhite;
  --light-color: darkslategray;
  --dark-bg: darkslategray;
  --dark-color: ghostwhite;
}
* {
  background-color: light-dark(
    var(--light-bg), var(--dark-bg
    ));
  color: light-dark(var(
    --light-color), var(
    --dark-color));
}
.light { color-scheme: light; }
.dark { color-scheme: dark; }
```

2

Example Result

light-dark() CSS function

Automatic

This section will react to the users system or user agent setting.

Light

This section will be light due to the color-scheme: light;.

Dark

This section will be dark due to the color-scheme: dark;.

[1] https://caniuse.com/?search=light-dark

[2] https://developer.mozilla.org/en-US/docs/Web/CSS/Reference/Values/color_value/light-dark

Functions: if()

- Experimental ¹

- Mimics familiar if-else behaviour in CSS blocks.

- Sets values based on conditional test.

- Enables dynamic style logic.

- Requires fallback options and allows nesting.

Example Code

```
color: if(
  style(--scheme: ice):
    if(
      media(prefers-color-scheme:
      dark): #caf0f8;
      else: #03045e;
    );
  style(--scheme: fire):
    if(
      media(prefers-color-scheme:
      dark): #ffc971;
      else: #621708;
    );
  else: black
);
```
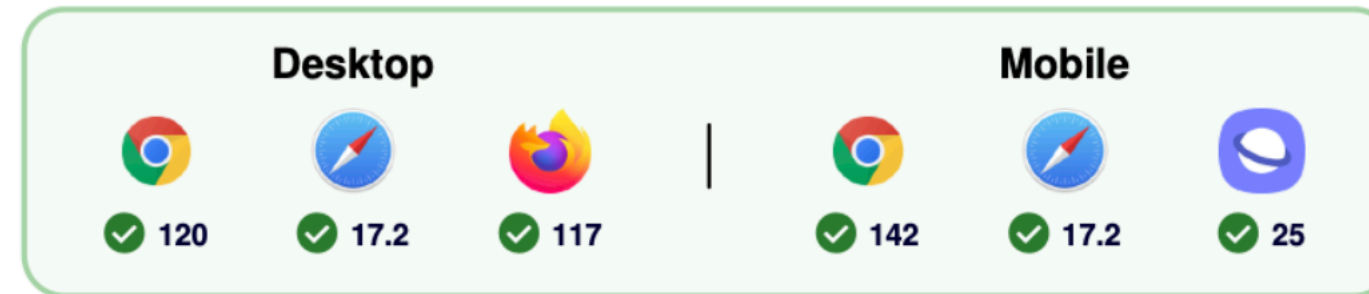²

¹ https://caniuse.com/?search=css+if%28%29

² https://developer.mozilla.org/en-US/docs/Web/CSS/Reference/Values/if

Other: CSS Nesting

- Baseline 2023

- Nesting of rules within on another.

- Write code more compact and intuitive.

- Increased readability, modularity, and maintainability of code.
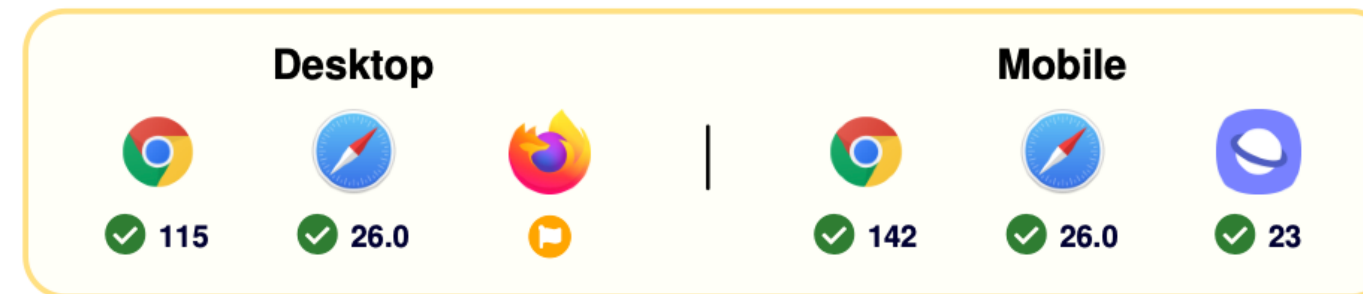
Example Code

```
/* Without CSS Nesting: */
.foo {
  background-color: red;
}
.foo .bar {
  font-size: 2.0rem;
}

/* Now it is possible to express
   the same logic more compactly:
   */
.foo {
  background-color: red;
  .bar {
    font-size: 2.0rem;
  }
}
```

[1] https://caniuse.com/css-nesting

Other: Scroll-Driven Animations

1

- Experimental

- Traditionally implemented with JavaScript.

- Animations linked directly to user's scroll progress.

- New possibilities to interactive web design.
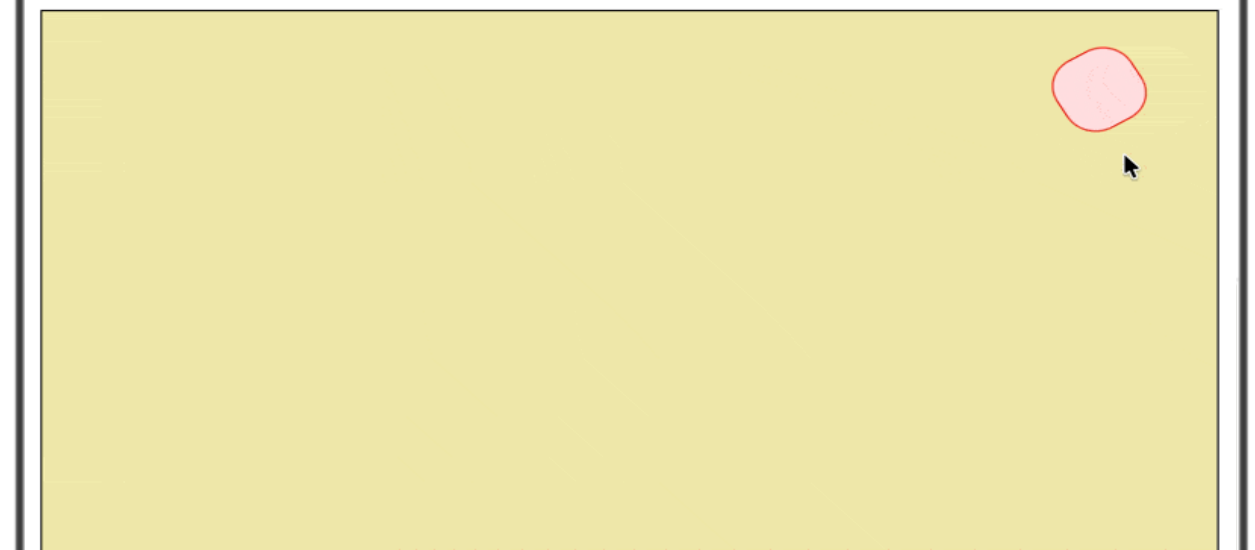
```
main {
    scroll-timeline:
        --main-timeline;
}

div {
    animation: background-animation
        linear;
    animation-timeline: scroll(
        nearest inline);
}

div::after {
    animation: shape-animation
        linear;
    animation-timeline:
        --main-timeline;
}
```
2

Example Result

1 https://caniuse.com/?search=scroll+driven+animations

2 https://developer.mozilla.org/en-US/docs/Web/CSS/Guides/Scroll-driven_animations

NoNewNews Demo

Demo of NoNewNews

Showcase Video

Christian Burtscher, Jonas Glaser, Marcus Gugacs, Eva Haring

24.11.2025

Information Architecture and Web Usability WS 2025/2026

Group 1

Using Chrome Canary Version 144.0.7553.0 on MacOS Tahoe 26.1

Youtube: https://youtu.be/-WVDrBNgazs

Github: https://github.com/gugacs/nonewnews

Future Work

- Web ecosystem is frequently evolving.

- Update & include further features.

- Extend No New News by:

  ○ Implementing additional (sub-)routes.

  ○ Making it more accessible (extending aria-label usage).

Questions?