

Styling the Intrinsic Web



Miriam Suzanne @ **An Event Apart Fall Summit**

October 11, 2021

The Present of CSS

The Past of CSS

CSS exists
For Two Reasons

1. *CSS makes*
Styles Responsive

*@media
min-width*

`@media
prefers-reduced-motion`

*@media
hover*

WorldWideWeb
Info ▶
Navigate ▶
Document ▶
Edit ▶
Find ▶
Links ▶
Style ▶
Print... p
Page layout...
Windows ▶
Services ▶
Hide h
Quit q

Info
HyperMedia Browser/Editor
An excercise in global information availability
by Tim Berners-Lee
Version 1.0
Alpha only
1990,91, CERN. Distribution restricted: ask for terms. TEST VERSION ONLY
text: Text which is not constrained to be linear.
media: Information which is not constrained linear... or to be text.

Welcome to the Universe of HyperText

File View

Home

Access to this information is provided as part of the [WorldWideWeb](#) project. The WWW project does not take responsibility for the accuracy of information provided by others.

How to proceed

References to other information are represented like [this](#). Double-click on it to jump to related information.

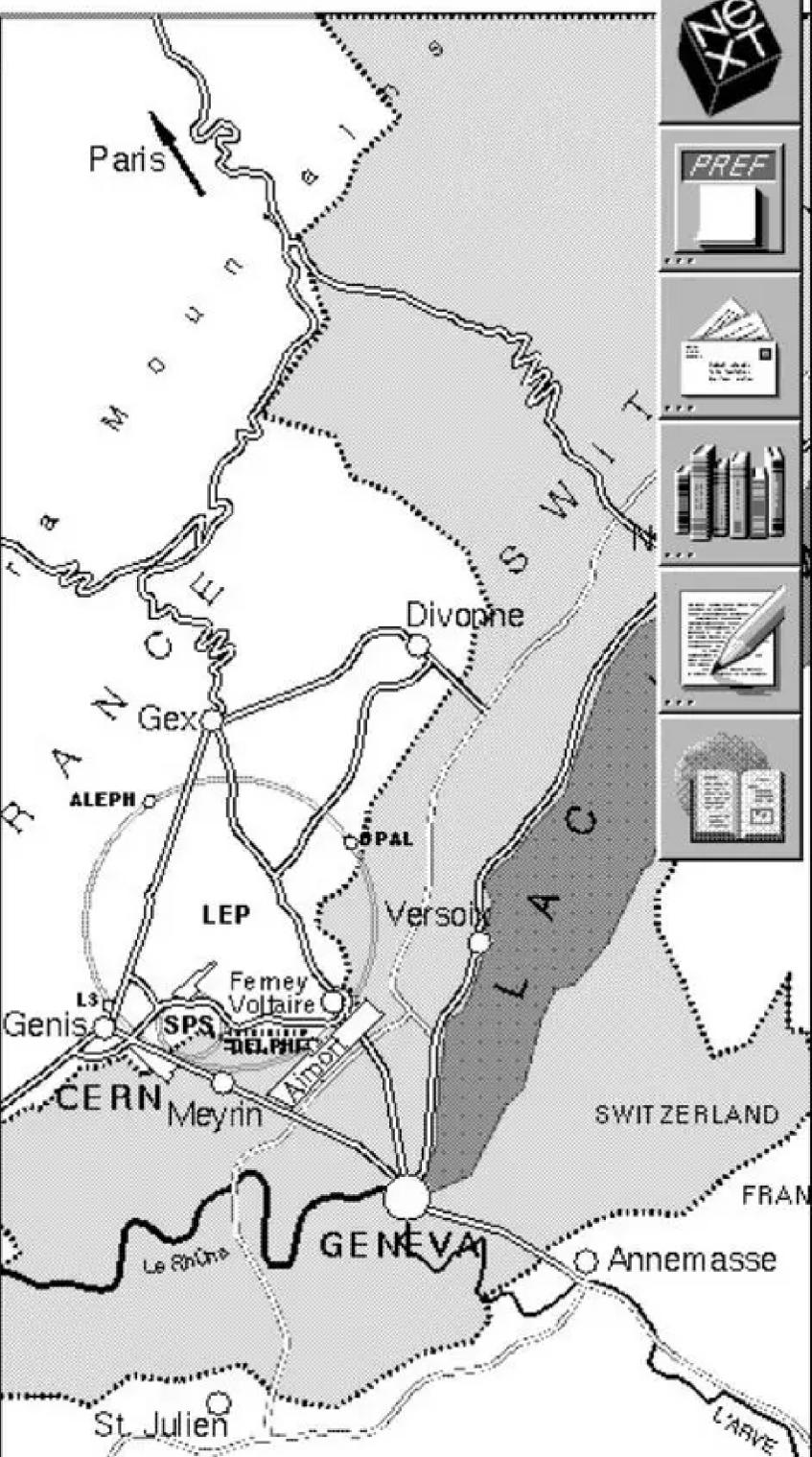
General CERN Information sources

Now choose an area in which you would like to start browsing. The system currently has access to three sources of information. With the indexes, you should use the keyword search option on your browser.

[CERN Information](#) A general keyword index of information made available by the computer centre, including CERN, Cray and IBM help files, "Writeups", and the Computer Newsletter (CNL). (This is the same data on CERNVM which is also available on CERNVM with the VM [FIND command](#)).

[Yellow Pages](#) A keyword index to the CERN telephone book by function.

You can access the internet news scheme ([See information for new users](#)). News articles are distributed typically CERN-wide or worldwide, and have a finite lifetime.





WORLD WIDE WEB

The WorldWideWeb (W3) is a wide-area hypermedia[1] information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary[2] of the project, Mailing lists[3] , Policy[4] , November's W3 news[5] , Frequently Asked Questions[6] .

What's out there?[7]Pointers to the world's online information, subjects[8] , W3 servers[9], etc.

Help[10] on the browser you are using

Software Products[11] A list of W3 project components and their current state. (e.g. Line Mode[12] ,X11 Viola[13] , NeXTStep[14] , Servers[15] , Tools[16] , Mail robot[17] , Library[18])

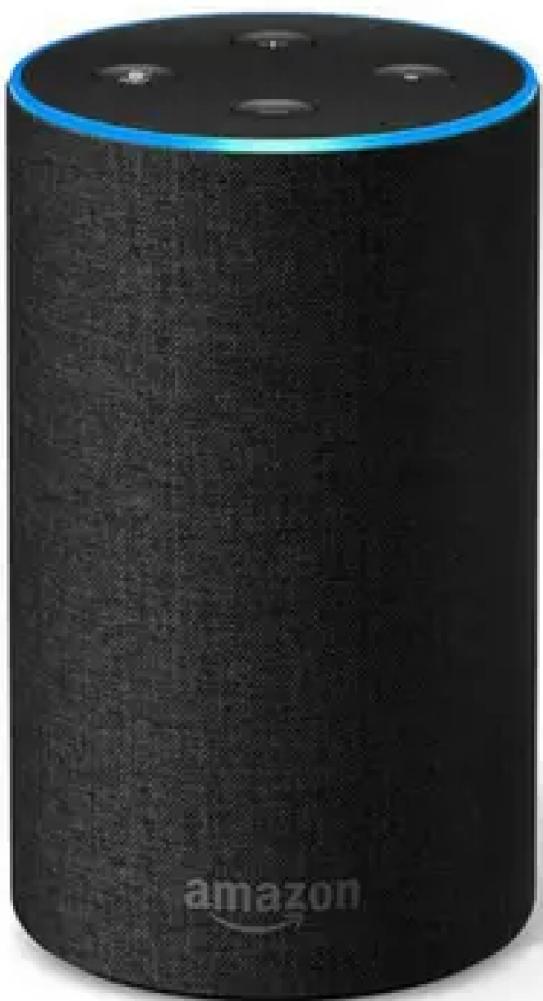
Technical[19] Details of protocols, formats, program internals etc

<ref.number>, Back, <RETURN> for more, or Help: |

" Web for all. Web on everything.

– W3C Mission Statement





" *There are too many variables to consider.*

– Keith J Grant

*" Graphic design of unknown content on an infinite
and unknown canvas, across operating systems,
interfaces, & languages...*

– me, on twitter...

"Provide hints that the browser may or may not use.

– Håkon Lie

The Cascade

" *An ordered list (cascade) of style sheets ... can be referenced from the same document.*

– Håkon Lie

" The user/browser specifies initial preferences and hands the remaining influence over to the document.

– Håkon Lie

Cascade Origins

-  **Author** (Document)
-  **User** Preferences
-  **User Agent** (Browser)

The Cascade

Resolves Merge Conflicts

1.  **Author** (Document)
2.  **User** Preferences
3.  **User Agent** (Browser)

" If conflicts arise the user should have the last word, but one should also allow the author to attach style hints.

– Håkon Lie

! important

A balance of power

1. !  User Agent **Important**
2. !  User **Important**
3. !  Author **Important**
4.  Author Styles
5.  User Preferences
6.  User Agent Defaults

2. *StyleSheets*

Make Styles Reusable

<P>...</P>

CSS Selectors

```
p {  
    color: blue;  
}
```

We use
Classes & Attributes

Combine them
To Compose Objects

Design Systems & Component Libraries



@stubbornella
Nicole Sullivan

2009 Object Oriented CSS ↗



@natbat
Natalie Downe

2008 [CSS Systems ↗](#)

A D T O M A

Issue №
306



Responsive Web Design

Technical Ingredients...

- Fluid grids (%)
- Flexible images (%)
- Media queries

Flexbox & Grids & ...

min-/max-content, fit-content, minmax(), clamp(), etc...



@jensimmons
Jen Simmons
Everything You Know About Web Design
Just Changed ↗

Intrinsic Web Design

Responsive & Reusable

Components

1. Fluid & Fixed
2. Stages of Squishiness
3. Truly Two-Dimensional Layouts
4. Nested Contexts
5. Expand & Contract Content
6. Media Queries, As Needed

” Our medium is not done. Our medium is still *going* through radical changes.

– Jen Simmons, *Designing with Grid*

What's Painful

About Intrinsic Design ?

CSS Selectors & Specificity



Conflict!

```
button          { background: gray; }
.action         { background: darkblue; }
[type="submit"] { background: darkgreen; }
#send          { background: maroon; }
```

Selector Specificity

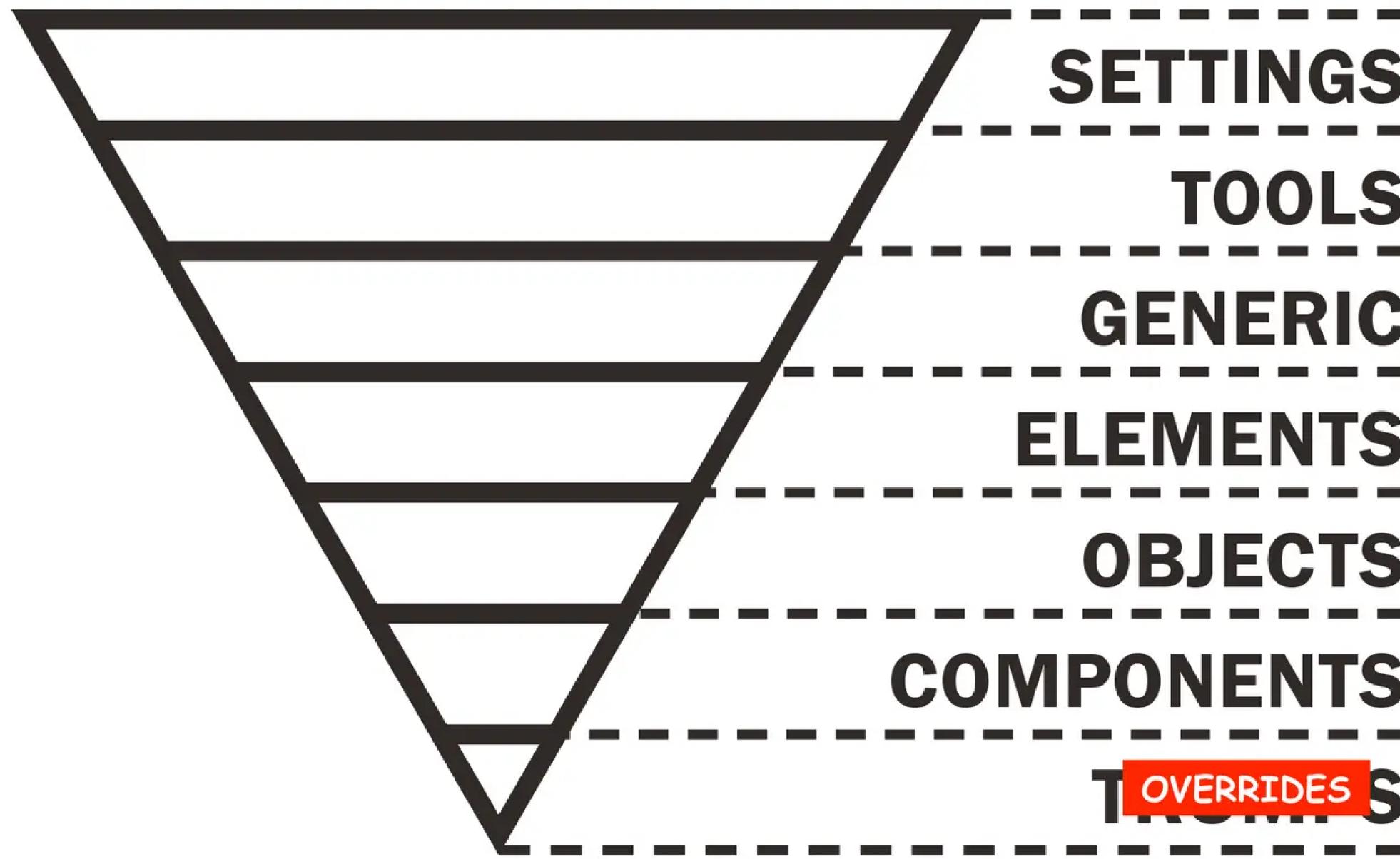
- * (**universal**)
- type
- .class & [attr]
- #IDs (**single-use**)

universal/type »
Global Defaults

attrs/classes »
Common Patterns

ID/style »
Singular Overrides

1. *Unique* #IDs
2. *Reusable* .classes & [attributes]
3. *Element* types
4. *Universal* *



Heuristics Can  Fail 

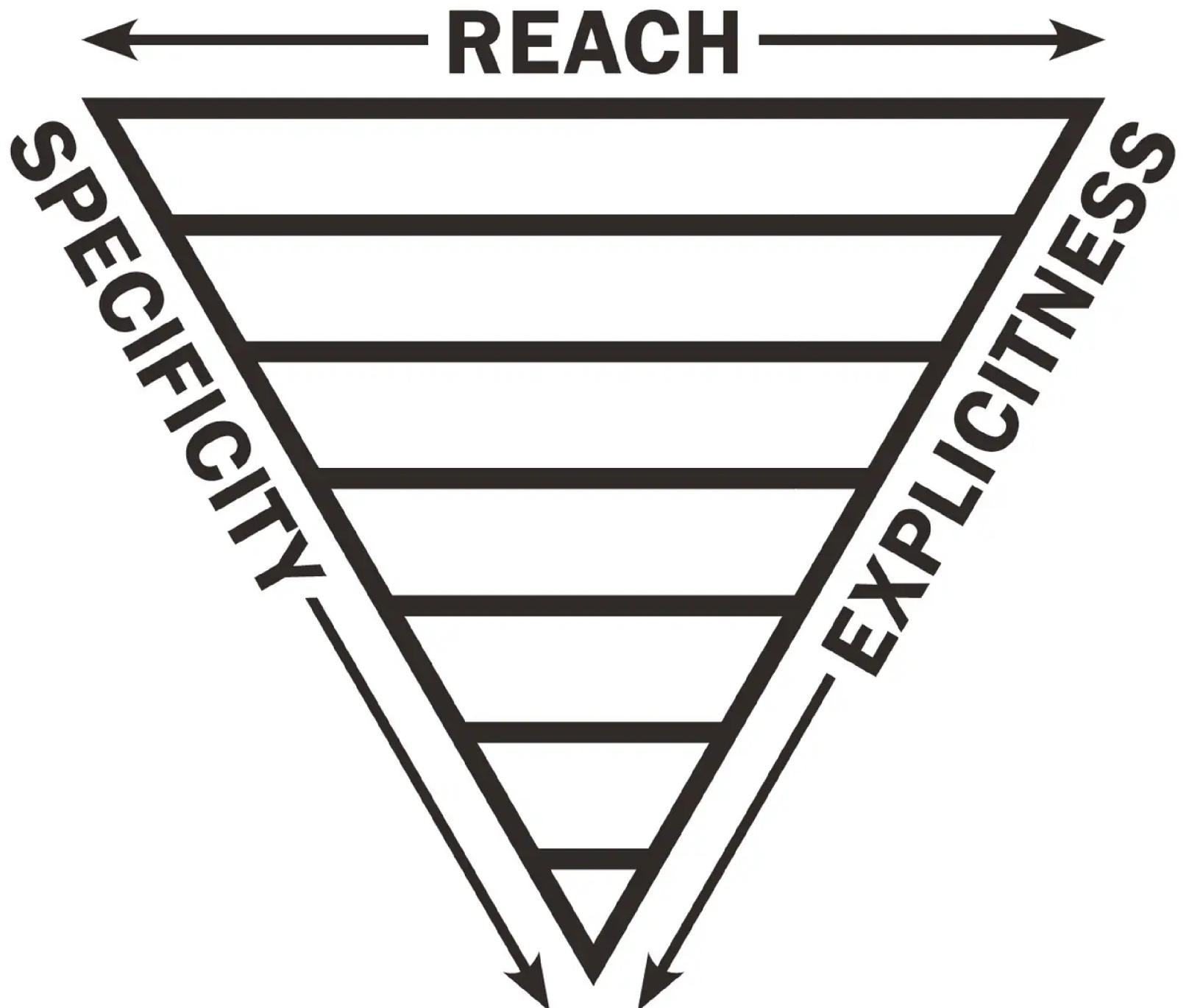
Especielly “At Scale”

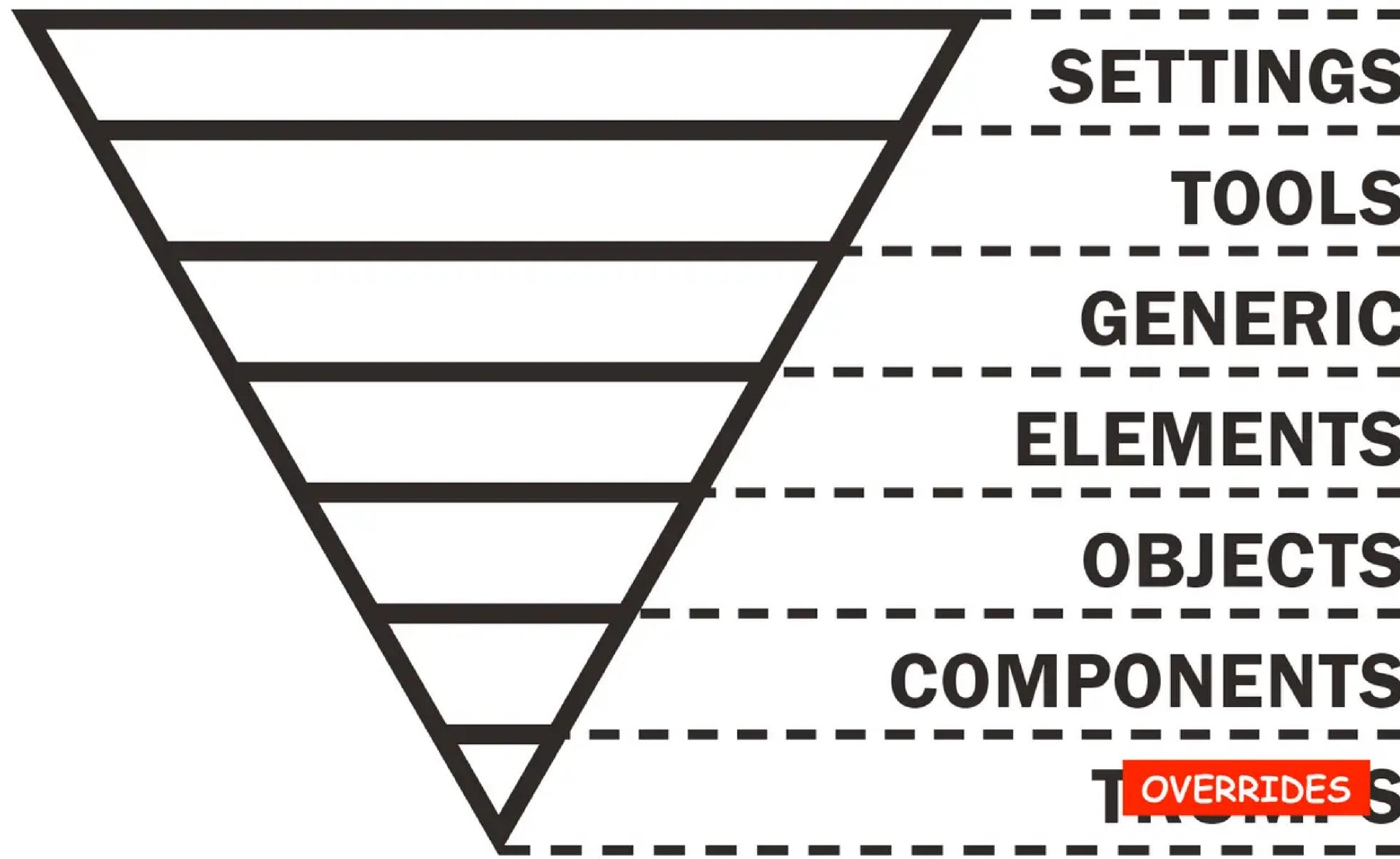
```
table[rules]:not([rules=""])> tr > td,  
table[rules]:not([rules=""])> * > tr > td,  
table[rules]:not([rules=""])> tr > th,  
table[rules]:not([rules=""])> * > tr > th,  
table[rules]:not([rules=""])> td,  
table[rules]:not([rules=""])> th  
{  
    border-width: thin;  
    border-style: none;  
}
```

```
[hidden] {  
    display: none !important;  
}
```

1. ~~Unique #IDs~~
2. *Reusable* .classes & [attributes]
3. ~~Element types~~
4. ~~Universal~~*

```
.block .element.modifier { /* 3 */ }
.block__element--modifier { /* 1 */ }
```



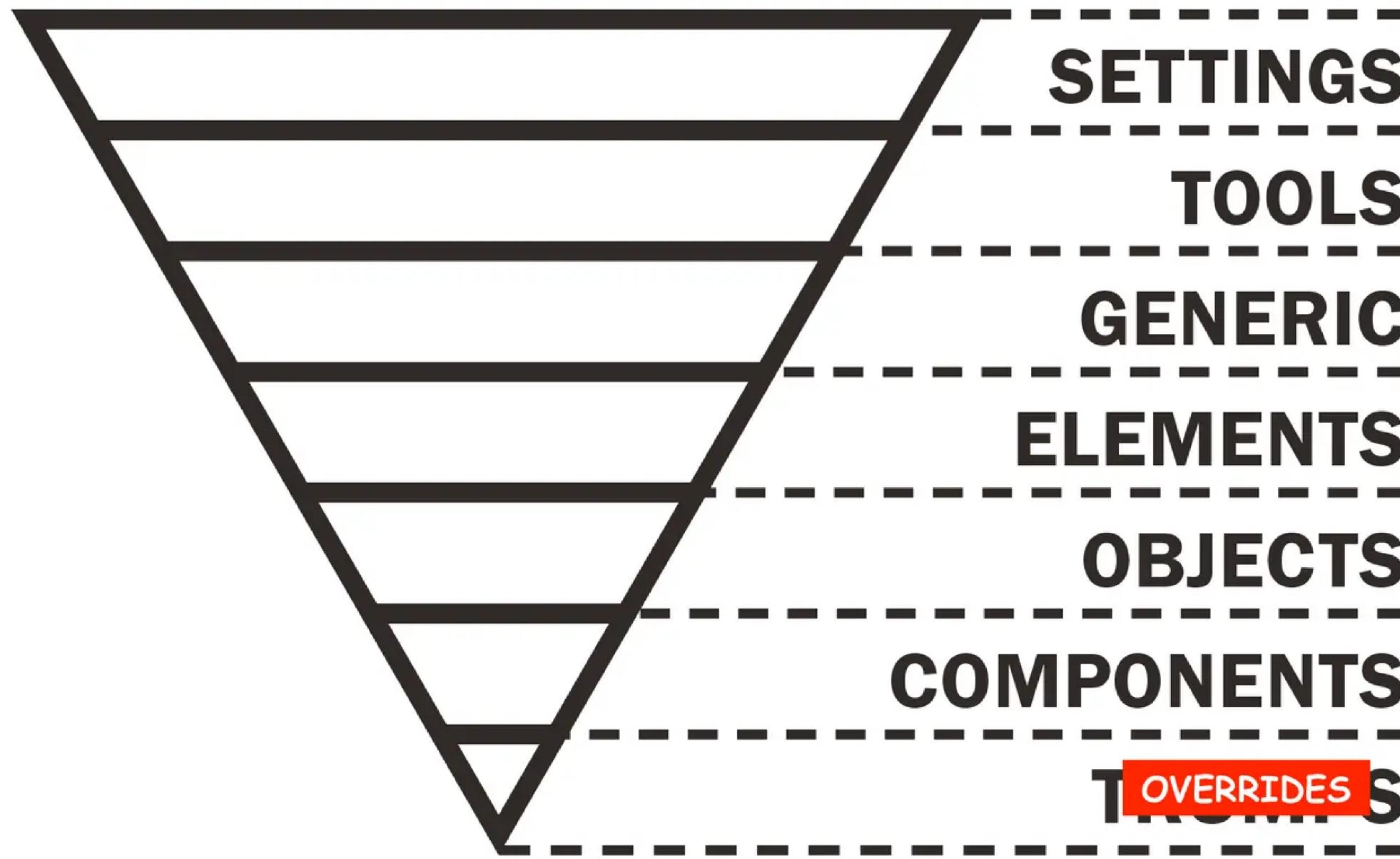


! important

A balance of power

```
.bootstrap-thing {  
    font-weight: bold !important;  
}
```

1. **Cascade Layers**
2. Scoped Styles
3. Container Queries



Stacked in Layers

- 1. Components?**
- 2. Themes?**
- 3. Frameworks?**
- 4. Resets?**

1. ! Important Resets
2. ! Important Themes
3. ! Important Components
4. Components
5. Themes
6. Resets

```
@import url(headings.css) layer(default);
```

```
@layer default {  
    audio[controls] { display: block; }  
    [hidden] { display: none !important; }  
}
```

```
@import url(headings.css) layer(default);  
  
@layer default {  
    audio[controls] { display: block; }  
    [hidden] { display: none !important; }  
}
```

```
@layer default { /* lowest */ }  
@layer theme { /* higher */ }  
@layer components { /* highest */ }
```

```
@layer framework {  
    .menu > .dropdown > .item {  
        background: whitesmoke;  
    }  
}  
}
```

```
@layer override {  
    .menu-item {  
        background: lightcyan;  
    }  
}
```

```
@layer default { /* ... */ }  
@layer theme { /* ... */ }  
  
/* still a lower layer than "theme" styles */  
@layer default { /* ... */ }
```

```
@layer default;  
@layer theme;  
@layer components;
```

```
@layer default {  
    * { box-sizing: border-box; }  
}
```

```
@layer default, theme, components;  
  
@layer default {  
    * { box-sizing: border-box; }  
}
```

```
@layer default, tools, theme, components;  
@import url(tools.css) layer(tools);
```

```
/* tools.css */
@layer theme { /* ... */ }
@layer components { /* ... */ }
```

```
@import url(tools.css) layer(tools);
```

```
@layer tools.theme { /* ... */ }
```

```
@layer tools.custom { /* ... */ }
```

```
@layer tools {  
    @layer custom {  
        /* tools.custom */  
    }  
}
```

More Cascade Control

Fewer Hacks

Unlayered styles

~~Default Lowest Priority~~

Default **Highest Priority**

Possible to Polyfill

Do we also need @supports (layers) conditional?

Cascade Layer Resources

- [Working Draft Spec ↗](#)
- [Full layers Explainer ↗](#)

BREAKING HEADLINES



Try Cascade Layers Today



[CodePen collection of layer demos ↗](#)

- **Firefox Nightly:**
» about:config » layout.css.cascade-layers.enabled
- **Chrome Canary:**
» chrome://flags » Enable CSS Cascade Layers
- **Safari Technology Preview:**
» Develop/Experimental Features (menu) » CSS Cascade Layers
» (#133 has unlayered styles reversed, likely fixed in #134)

```
1/* layers override unlayered styles */  
2@layer override {  
3  h1 {  
4    color: green;  
5  }  
6  
7  h1::before {  
8    content: "✓ ";  
9  }  
10}  
11  
12/* despite the higher specificity */  
13#h1 {  
14  color: red;  
15}  
16  
17#h1::before {  
18  content: "✗ ";  
19}  
20  
21/* for readability */  
22html {
```

JS Twilight (Dark Background) Console Display URL Assets

X Does This Browser Support @layer?

In **Firefox Nightly**, go to `about:config` and toggle the `layout.css.cascade-layers.enabled` feature flag.

In **Chrome Canary**, this currently requires a [run-time flag](#) (`--enable-blink-features=CSSCascadeLayers`) while opening the browser from the command line.

To get a more extensive overview of support in your current browser, see the Web Platform Tests for [basic @layer support](#), [layering with @import](#), and [layering @keyframes](#).

← BACK TO AGENDA

NOTES

MANAGE

An Event Apart
ONLINE TOGETHER

Spring Summit · April 19–21, 2021

Miriam Suzanne

Beyond Variables

48:03

(ONLINE) WEB DESIGN CONFERENCE TOOLS & FRONT-END EXPERTS

vimeo

The slide features a central portrait of Miriam Suzanne and a background illustration of hands holding laptops and phones, with purple flowers at the bottom.

Chat

Polls

Files

People



No polls found

This meeting does not have any polls

Beyond Variables

THEATER MODE



Miriam Suzanne
OddBird
Co-founder

⌚ 1:30 PM - 2:30 PM MDT on Monday, April 19

Add to Calendar ▾

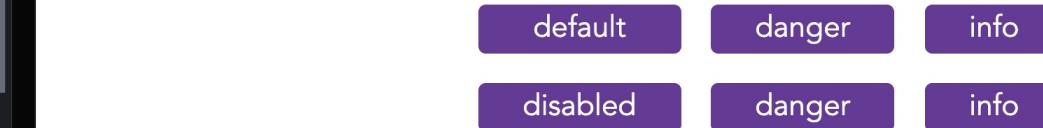
HTML

```
1 <main>
2   <button class="btn">default</button>
3   <button class="btn danger">danger</button>
4   <button class="btn info">info</button>
5   <button class="btn" disabled>disabled</button>
6   <button class="btn danger" disabled>danger</button>
7   <button class="btn info" disabled>info</button>
```

css

```
1 @layer theme, state;
2
3 .btn {
4   background: rebeccapurple;
5   color: white;
6 }
7
8 @layer state {
9   :disabled {
10   background: dimgray;
11 }
12 }
13
14 @layer theme {
15   .danger {
```

JS



github.com/w3c/csswg-drafts ↗

Starts with

The **Proposal ↗** on Github

Editor's Draft ↗

Working Draft ↗

Web-Platform-Tests ↗.org

wpt.live ↗ | wpt.fyi ↗

"One of the best ways a web developer can truly help make websites work the same in every browser is to help create more tests for WPT. We all want interoperability. Testing is a way to get there, and not regress. But we need more tests written to check on more things.

– Jen Simmons, [Twitter↗](#)

1. Cascade Layers
2. **Scoped Styles**
3. Container Queries

1. *Avoid*

Naming Conflicts

(across *large teams & projects*)

2. By

Expressing Membership

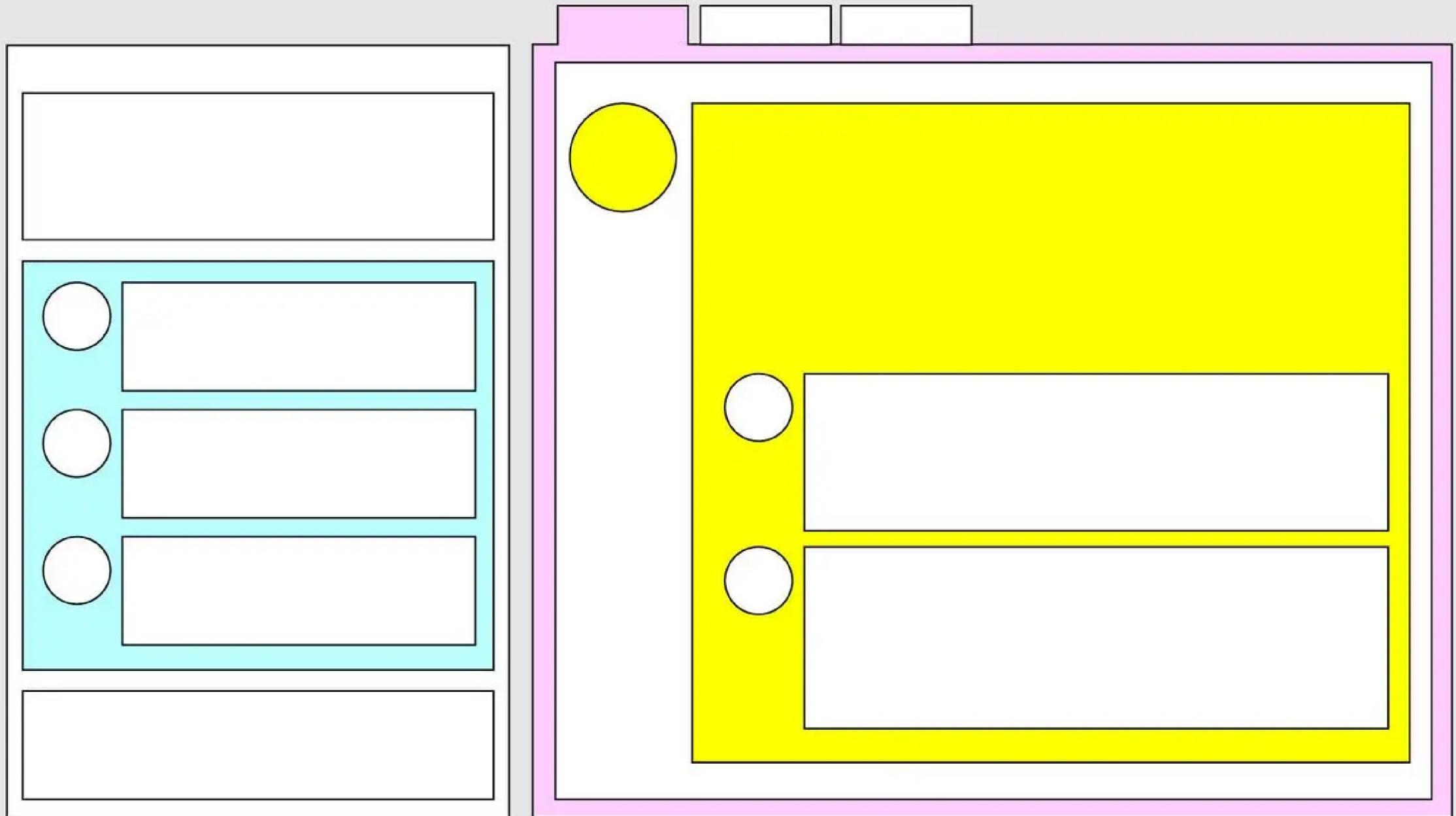
(through *lower boundaries* & *proximity*)

```
.title { /* global */ }  
.post .title { /* nested */ }
```

```
.title { /* global */ }  
.post .title { /* nested */ }  
  
.post__title { /* BEM */ }
```

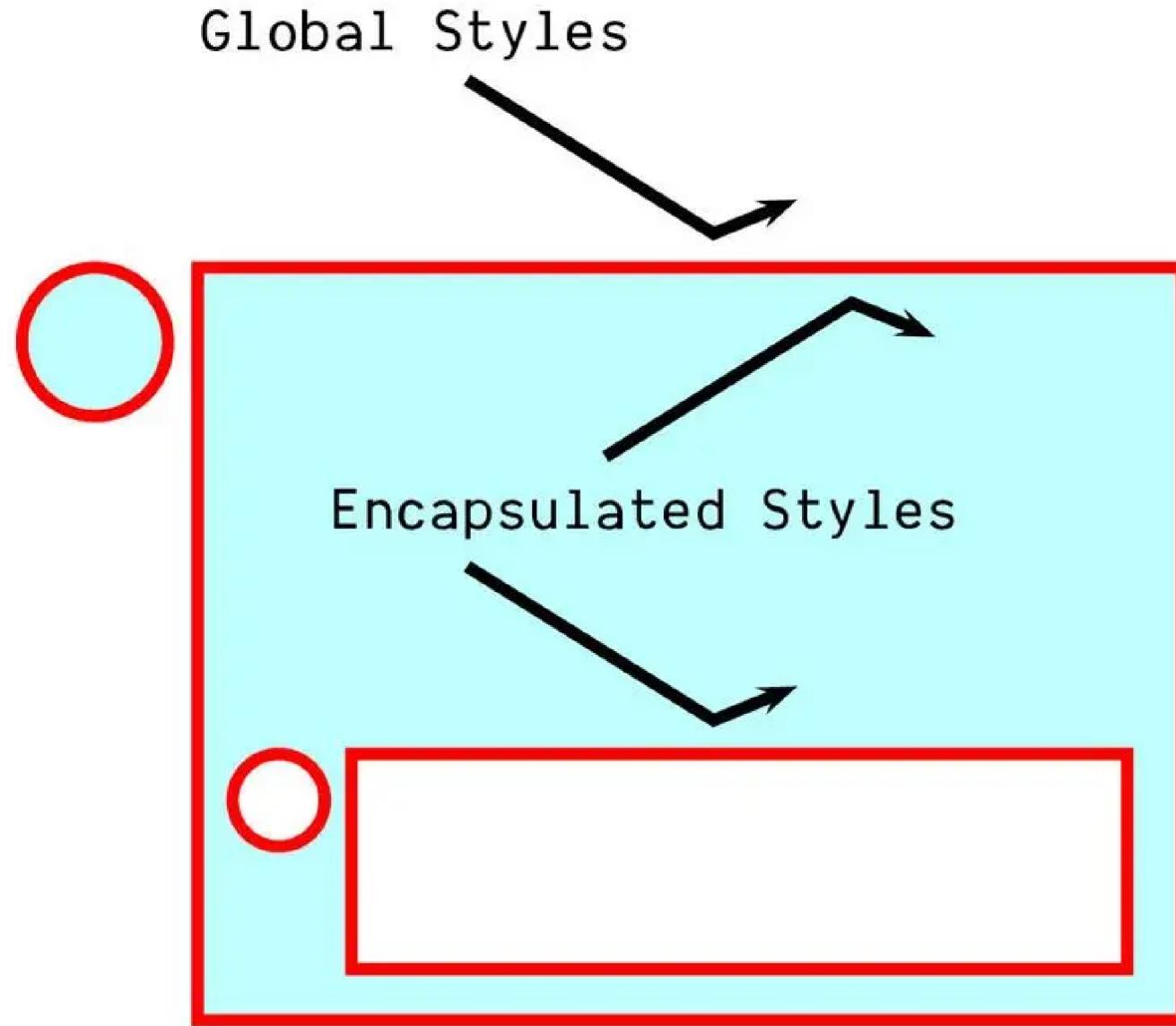
```
.post__title { /* BEM */ }
.title[data-JKGHJ] { /* Vue */ }
```

```
<h2 class="title post_title">
```



A “donut scope” with *lower boundaries* similar to shadow-DOM slots

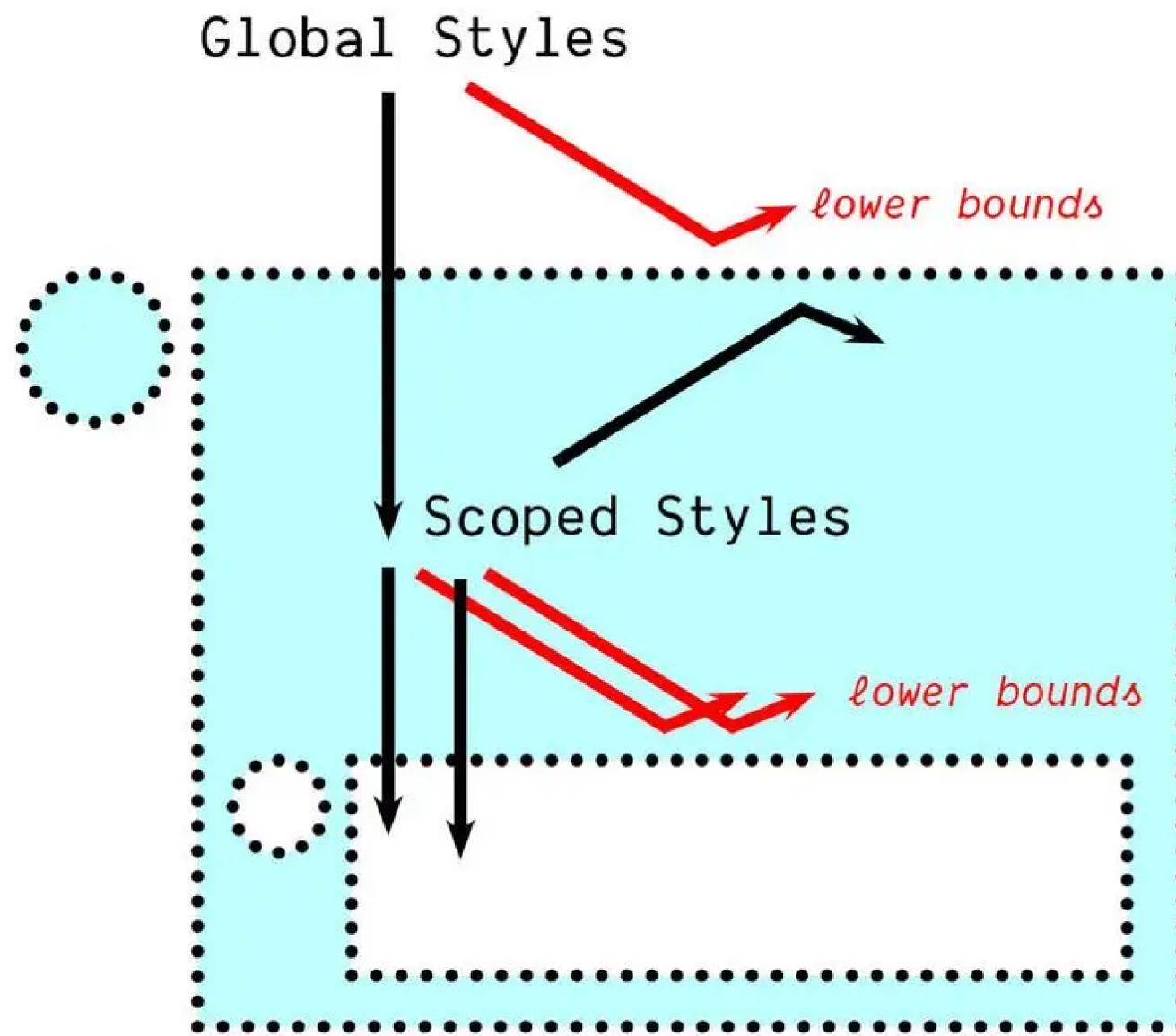
Different from
Shadow-DOM Encapsulation



Build-tools

Provide Scoped Styles

BEM, CSS Modules, Vue, JSX, Stylable, etc



```
@scope (.media) to (.content) {  
    img { /* only images that are "in scope" */ }  
}
```

```
.light-theme a { color: purple; }  
.dark-theme a { color: plum; }
```

Scope Proximity Issue

Light Theme

purple

Dark Theme

plum

Nested Light Theme

also plum???

```
@scope ([data-theme=light]) to ([data-theme]) {  
    a { color: purple; }  
}  
  
@scope ([data-theme=dark]) to ([data-theme]) {  
    a { color: plum; }  
}
```

```
@scope ([data-theme=light]) {  
    a { color: purple; }  
}  
@scope ([data-theme=dark]) {  
    a { color: plum; }  
}
```

A Donut Selector ?

```
@scope (.media) to (.content) {  
    img { border: red; }  
}  
  
/* as a selector, without proximity rules? */  
img:in(.media / .content) { border: red; }
```

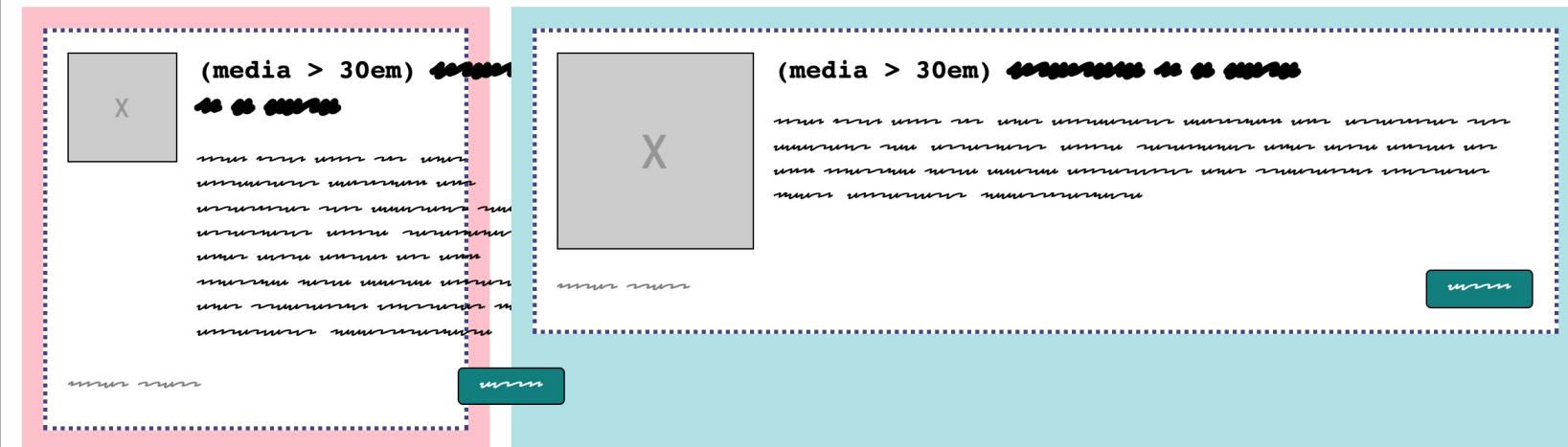
Scope Resources

- [Full scope Explainer ↗](#)
- [CSSWG Issue ↗](#)
- [Draft Spec ↗](#)

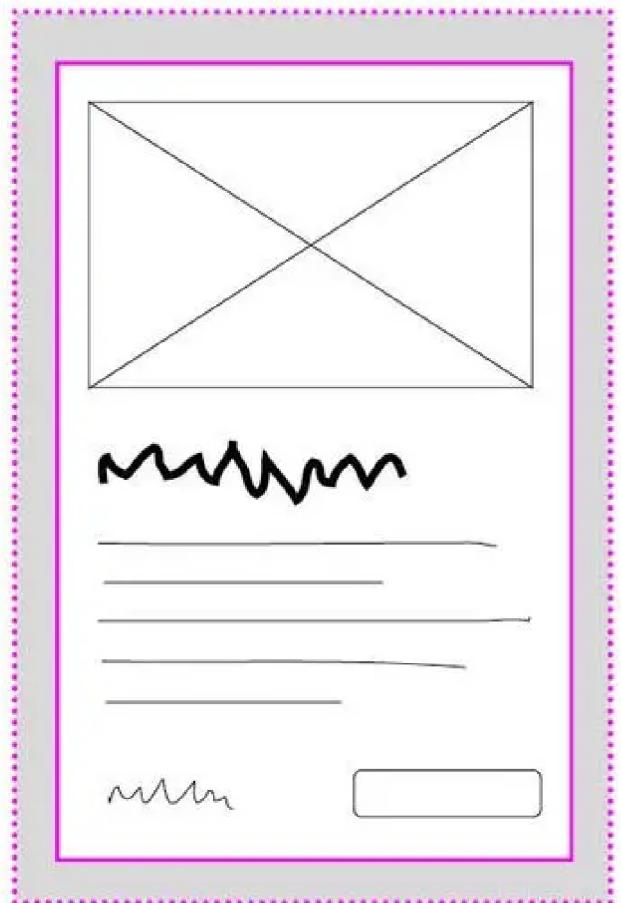
1. Cascade Layers
2. Scoped Styles
3. **Container Queries**

```
1 .root {  
2   display: grid;  
3   gap: 1em;  
4   margin: 1em;  
5 }  
6  
7 header {  
8   grid-column: 1 / -1;  
9 }  
10  
11 header,  
12 aside,  
13 main {  
14   background: gainsboro;  
15   padding: 1em;  
16 }  
17  
18 .card {  
19   --media: "small";  
20   font-family: "Redacted";  
21   display: grid;  
22   grid-template-columns: ;
```

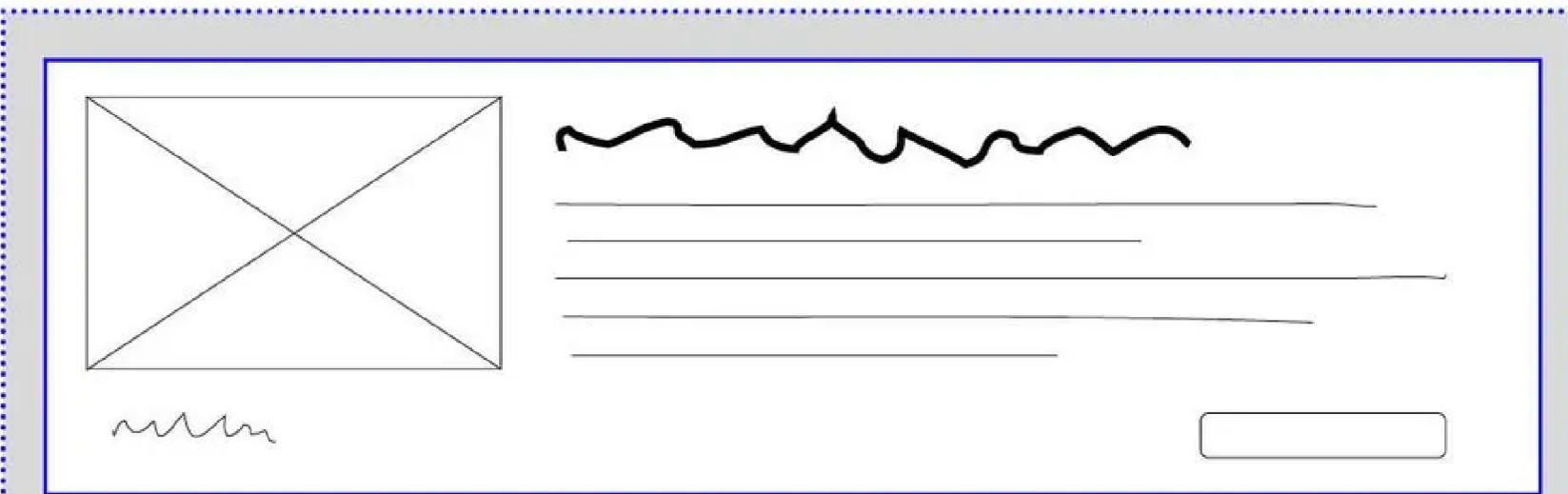
Media vs Container



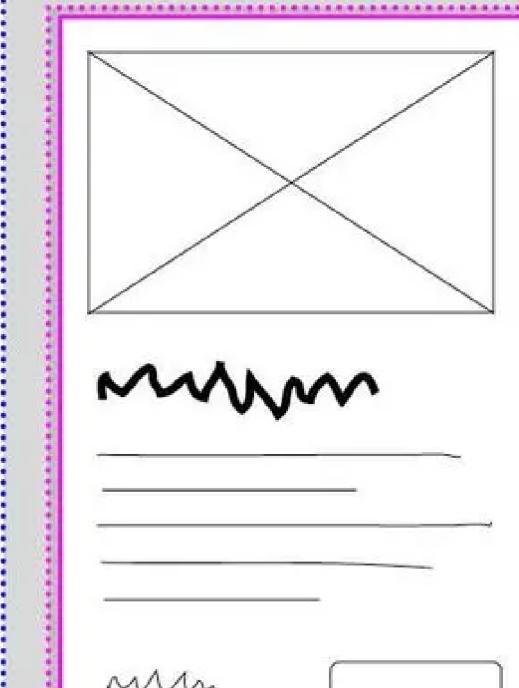
.sidebar



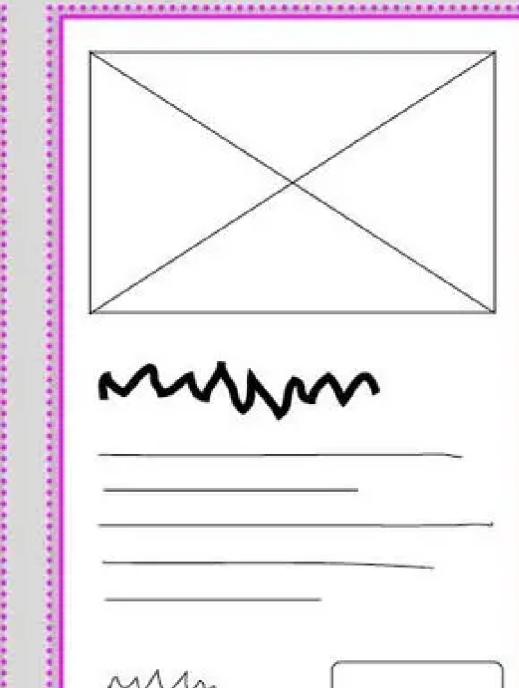
main



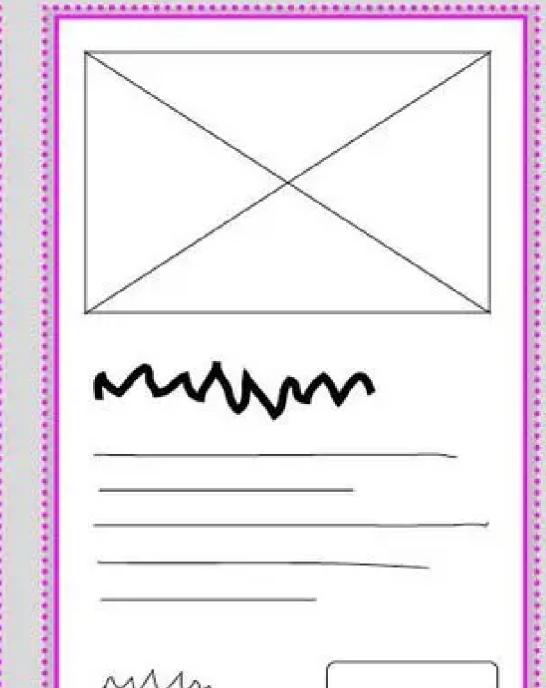
.grid-item



.grid-item



.grid-item



Layout Loops

CSS *Context* vs *Content*

"Normal" Flow

Defines the default layout system on the web – a **document-like flow** of responsive **block** and **inline** elements.

While **inline** elements flow in the direction of text, **and are able to wrap around line-breaks** — **block** elements generate a solid box that stacks on the cross-axis.

Edit this text

2010-2020



Laying Foundations

2020 Proposals

- David Baron: `@container`
limited by containment
- Brian Kardell: `switch()`
limited to paint

Defining Containers

.sidebar

main

.grid-item

.grid-item

.grid-item

No Content Sizing

```
.container {  
  contain: size layout style;  
}
```

2D size containment

Is Too Restrictive

HTML

```
1<div class="rad">
2  CSS is
3  <strong contenteditable="">Awesome</strong>
4 </div>
```

CSS

```
1/* demo */
2.rad {
3  border: medium solid;
4  font-size: 2em;
5  margin: 0 auto;
6  max-width: 2.25em;
7  padding: 0.25em 0.5em;
8 }
9
10/* editing */
11.rad [contenteditable] {
```

JS



CSS
is
Awesome

We need
Inline Size Containment

```
.container {  
  contain: inline-size;  
}
```



DRAGONS

```
.container {  
  contain: block-size;  
}
```

```
.sidebar, main, .grid-item {  
  contain: inline-size layout style;  
}
```

```
.sidebar, main, .grid-item {  
  container-type: inline-size;  
}
```

Warning!

Not a Stable Spec (Yet)

Querying Containers

```
@container (min-width: 40em) {  
  .card { /* ... */ }  
  h2 { /* ... */ }  
}
```

```
<div class="container">
  <div class="container">
    <div class="container">
      We can nest containers!
    </div>
  </div>
</div>
```

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

@container (width > 30em) {
  .container { padding: 2em; }
}
```

```
.sidebar {  
  container-type: inline-size;  
  container-name: sidebar;  
}  
  
@container sidebar (width > 30em) {  
  .container { padding: 2em; }  
}
```

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

@container type(inline-size) (width > 30em) {
  .container { padding: 2em; }
}
```

*Chrome **Prototype***

1. Download/Update [Chrome Canary ↗](#)
2. Go to chrome://flags in the URL bar
3. Search for “CSS Container Queries” & enable it
4. You’ll need to restart after turning it on

```

1 .root {
2   display: grid;
3   gap: 1em;
4   margin: 1em;
5 }
6
7 header {
8   grid-column: 1 / -1;
9 }
10
11 header,
12 aside,
13 main {
14   background: gainsboro;
15   padding: 1em;
16 }
17
18 .card {
19   --media: "small";
20   font-family: "Redacted";
21   display: grid;
22   grid-template-columns: ;

```

Media vs Container

✓ sidebar containers

(media > 30em)

(media > 30em)

```

1 <main>
2   <article>
3     <h1> Container Queries with Block Quotes </h1>
4     <section class="first">
5       <h2 class="title"> I </h2>
6       <div>
7         <blockquote class="quote">
8           <div class="media">
9             

```

CSS (SCSS)

```

1 /* declare which containers to query */
2
3 article,
4 section {
5   container-type: inline-size;
6 }
7
8 /* baseblockquote styles */
9
10 .quote {
11   background-color: var(--quote-background, bla

```

JS

26px Twilight (Dark Background) Console Display URL Assets

0.25x 0.5x 1.0x

Container Queries with Block Quotes

I



They do great work. We'll be working with them again soon.

David A. Herron, OddBird

II



They do great work. We'll be working with them again soon.

David A. Herron, OddBird

III



They do great work. We'll be working with them again soon.

David A. Herron, OddBird

photo: Arturo Oliva Pedroza

```

1 .card {
2   /* a container for every card */
3   container-type: inline-size;
4   flex: 1 1 min(100%, 20em);
5 }

6
7 /* the card can't query itself */
8 /* so we add an element directly inside */
9 article {
10   flex: 1;
11   display: grid;
12   grid-template: "img" 8em "content"
13   gap: 1em;
14   padding: 1em;
15 }
16
17 /* change the grid template when the screen width reaches 30em */
18 @container (min-width: 30em) {
19   article {
20     grid-template: "img content" auto
21   }
22 }

```

Resources



Container Queries & The Future of CSS

March 25 in [Talks](#), [CSS](#)

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Fugiat blanditiis excepturi debitis ipsum illo. Reiciendis animi ducimus dicta iste accusantium corrupti, beatae odit ex earum, molestiae at?



Beyond CSS Variables

March 23 in [Talks](#), [CSS](#)

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Fugiat blanditiis excepturi debitis ipsum illo. Reiciendis animi ducimus dicta iste accusantium corrupti, beatae odit ex earum, molestiae at?



JS Danger, CSS-Tricks Edition

March 12 in [Podcasts](#), [CSS](#)

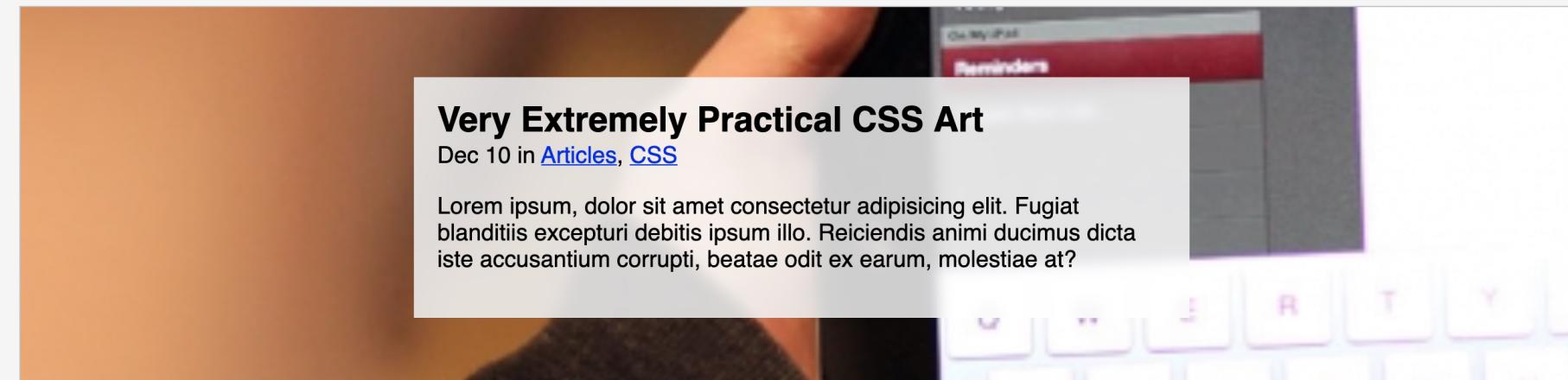
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Fugiat blanditiis excepturi debitis ipsum illo. Reiciendis animi ducimus dicta iste accusantium corrupti, beatae odit ex earum, molestiae at?



User Unfriendly

Feb 11 in [Talks](#), [UX](#)

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Fugiat blanditiis excepturi debitis ipsum illo. Reiciendis animi ducimus dicta iste accusantium corrupti, beatae odit ex earum, molestiae at?



```

1<!-- BOOK TEMPLATE -->
2<template id="book-template">
3  <style>
4    /* Layout Container */
5    :host {
6      display: block;
7      contain: layout style inline;
8      perspective: 1000px;
9      cursor: grab;
10 }
11

```

CSS

```

1:root {
  --spacing: 1rem;
  --highlight-color: #52947c;
}

body {
  font-family: sans-serif;
}

/* Basic Layout Sections */

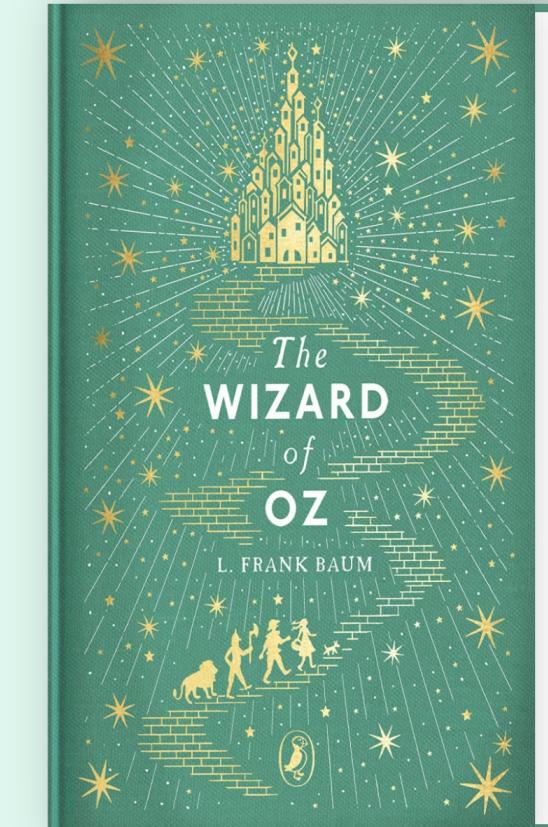
```

JS

26px Twilight (Dark Background) Console Display URL Assets

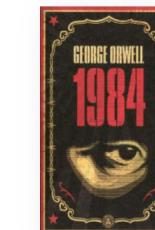
0.25x 0.5x 1.0x

FEATURED

**The Wizard of Oz**

L. Frank Baum

BESTSELLERS

**1984**

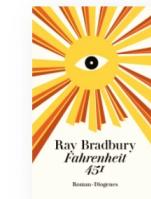
George Orwell

**Moby Dick**

Herman Melville

**Little Women**

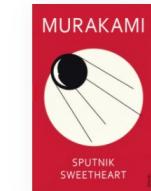
Louisa May Alcott

**Fahrenheit 451**

Ray Bradbury

**Pride and Prejudice**

Jane Austen

**Sputnik Sweetheart**

Haruki Murakami



Don Quixote

```
73 // Cart lines at 30px or wider
74 @container (min-width: 30px) {
75   .cart-lines-group {
76     display: block;
77   }
78 }

79

80 // Add "+" at 50px or wider
81 @container (min-width: 50px) {
82   .plus-group {
83     display: block;
84   }
85 }

86

87 // "Add" instead of "+" at 100px
88 @container (min-width: 100px) {
89   .cart-button {
90     padding: 0 1rem;
91     display: grid;
92     max-width: 120px;
93     grid-template-columns: 1fr 1fr
94   }
}
```

Responsive Button (Container Queries)



Demo Card with Responsive Button

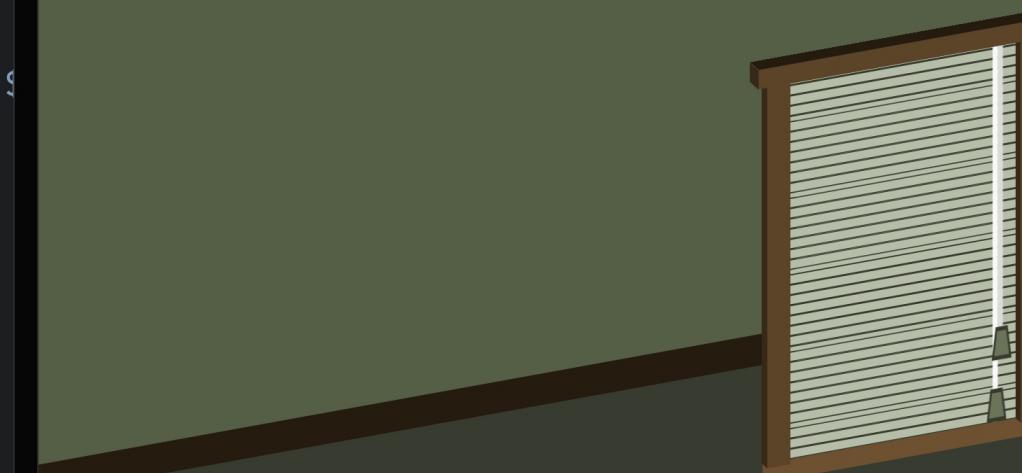
PINK HAND-MADE
PLANTER

\$16.99

Handmade in Brooklyn, NY.
Dimensions: 16" at opening, 10" at base. Round dome shape for single or multiple plantings.

ADD TO CART

```
290 $open = $p / $steps  
291 $peter = $p / $steps  
292  
293 @container (min-height: {$step}px)  
294 .blinds  
295 --open $open  
296  
297 @media --css-container and (min-height: $  
298 .blinds  
299 --open $open  
300  
301 @container (min-height: {$pStep}px)  
302 .peter  
303 --peter $peter  
304  
305 @media --css-container and (min-height: $  
306 .peter  
307 --peter $peter  
308  
309 .window-frame  
310 div  
311 background var(--f-1)
```



HTML

187

```
188 @container glass (min-width: 14em) {  
189   h1 {  
190     --after: "?";  
191   }  
192  
193 section {  
194   --water: rebeccapurple;  
195 }  
196 div {  
197   --cube: 4.5em;  
198 }  
199 }  
200  
201 @container glass (min-width: 24em) {  
202   h1 {  
203     --after: "????";  
204   }  
205  
206 section {  
207   --water: darkviolet;  
208 }
```

JS



This is all because [@5t3ph](#) said [my containers don't hold water](#). Well I sure showed her.

26px Twilight (Dark Background) Console Display URL Assets

0.25x 0.5x 1.0x

More to do...

Container Query units

~~qw | qh | qi | qb | qmin | qmax~~

cqw | cqh | cqi | cqbs | cqmin | cqmax

BREAKING HEADLINES



Try Query Units Today



Why CQ?

Because CH Already Exists

```
1/* containers */
2body,
3main,
4aside {
5  container: inline-size;
6}
7
8/* query units! */
9h1 {
10  font-size: max(2em, 1.25em + 3qi);
11  margin: 2qi 0;
12}
13
14h2 {
15  font-size: max(1.5em, 1.23em + 2qi);
16}
17
18p {
19  font-size: max(1em, 1em + 1qi);
20}
21
22/* layout */
```

Hello Container Units

Aside

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Deleniti voluptate rem velit accusantium quasi, aliquid natus, aperiam officiis repellendus reprehenderit sapiente commodi cum deserunt quae illo blanditiis. Quasi, ullam totam.

Main

This demonstrates basic support for inline query `qi` units, which are likely to be the most used [Container Relative](#) units.

```

19   color: var(--text);
20   background: var(--background);
21 }
22 html,
23 main,
24 article {
25   container: inline-size;
26 }
27
28 /* Query units! */
29 h1,
30 .product-price {
31   font-size: max(1.25rem, 12qi - 1rem);
32   font-weight: 700;
33   line-height: 1;
34   margin: 0;
35   mix-blend-mode: multiply;
36 }
37
38 /* Layout */
39 main {
40   width: min(100% - (var(--margin) * 2), var(

```

Latest Products



Nike Metcon 7 \$130

Women's Training Shoes



Nike Free Metcon 4 \$120

Women's Training Shoes



Nike Space Hippie 04 \$130

Men's Shoes

More to do...

Non-size Queries

```
@container style(--colors: invert) { ... }
```

```
@container state(is-stuck) { ... }
```

Coming Soon

with a *polyfill*

```
@container (width > 30em) { /* CQ support */ }

@supports not (container-type: inline-size) {
  @media (width > 40em) { /* no CQ support */ }
}
```

CQ Proposal Resources

- [Specification ↗](#)
- [Original Explainer ↗](#)
- [CSSWG project & issues ↗](#)
- [More Articles & Demos »](#)

1. Cascade Layers
2. Scoped Styles
3. Container Queries

Building on
Existing **CSS**

Building on
The Cascade

Already make...

Styles Responsive

Already make...

Styles Reusable

Intrinsic Web Design

building *responsive components* in CSS

" Our medium is not done. Our medium is still *going* through radical changes.

– Jen Simmons, *Designing with Grid*



@TerribleMia

Miriam Suzanne