

That's not how I wrote CSS 3 years ago

MATUZOVIĆ

FRONT-END / AI1Y / HTML / CSS



Hi, I'm Manuel! 🦊

matuzo.at

manuel@matuzo.at

front-end.social/@matuzo

htmlhell.dev



```
1  div {
2      position: absolute;
3      inset: 0;
4  }
```

inset shorthand property

```
1  input:where([type="text"], [type="email"]) {
2
3
4  }
5
6  a:is(:link, :visited) {
7
8
9  }
10
11
12
13
14
15
```

:where() and :is() pseudo-classes

```
1  @layer base {
2
3  }
4
5  @layer components {
6
7  }
8
9  @layer utility {
10
11 }
12
13
14
15
```

cascade layers

```
1  body {
2    background-color: #123456;
3    background-color: rgb(18 52 86);
4    background-color: hsl(210deg 65.38% 20.39%);
5    background-color: hwb(210deg 7.06% 66.27%);
6    background-color: lch(21 24.7 264.72);
7    background-color: lab(21 -2.27 -24.59);
8    background-color: oklch(0.32 0.07 251.18);
9    background-color: oklab(0.32 -0.02 -0.07);
10 }
11
12
13
14
15
```

different colour notions, from HEX to oklab()



#100DaysOfMoreOrLess
ModernCSS

100 Days Of More Or Less Modern CSS

posted on September 26., 2022

It's time to get me up to speed with modern CSS. There's so much new in CSS that I know too little about. To change that I've started #100DaysOfMoreOrLessModernCSS. Why more or less modern CSS? Because some topics will be about cutting-edge features, while other stuff has been around for quite a while already, but I just have little to no experience with it.

[Day 100: it's over, or is it!?](#)

[Day 99: native nesting](#)

[Day 98: oklab\(\) and oklch\(\)](#)

[Day 97: animating grids](#)

[Day 96: the margin-trim property](#)

[Day 95: the color-mix\(\) function](#)

[Day 94: the accent-color property](#)

[Day 93: the lch\(\) color function](#)

[Day 92: relative color syntax](#)

[Day 91: a previous sibling selector with :has\(\)](#)

[Day 90: scoped styles in container queries](#)

[Day 89: higher-order custom properties](#)

[Day 88: CSS Motion Path](#)

[Day 87: mask properties](#)

[Day 86: the initial-letter property](#)

[Day 85: typed custom properties in container style queries](#)

[Day 84: the @property at-rule](#)

[Day 83: computed values in container style queries](#)

[Day 82: value processing](#)

[Day 81: the order of individual transform properties](#)

[Day 80: container style queries](#)

[Day 79: font-tech\(\) and font-format\(\)](#)

[Day 78: container query units](#)

[Day 77: block-size, inline-size, vi, and vb](#)

[Day 76: overwriting colors in font palettes](#)

[Day 75: font palettes](#)

[Day 74: using !important in cascade layers](#)

[Day 73: size container features](#)

[Day 72: the masonry-auto-flow property](#)

[Day 71: the masonry keyword](#)

[Day 70: the defined pseudo-class](#)



What's new in CSS and UI: I/O 2023 Edition

Published on Thursday, May 11, 2023



By [Una Kravets](#), [Bramus](#) and [Adam Argyle](#)

The past few months have ushered in a golden era for web UI. New platform capabilities have landed with tight cross-browser adoption that support more web capabilities and customization features than ever.

Here are 20 of the most exciting and impactful features that landed recently or are coming soon:

Table of contents

The New Responsive

[Container Queries](#)

[Style Queries](#)

[:has\(\)](#)

[nth-of syntax](#)

[text-wrap: balance](#)

[initial-letter](#)

[Dynamic viewport units](#)

[Wide-gamut color spaces](#)

[color-mix\(\)](#)

CSS Foundations

[Nesting](#)

[Cascade layers](#)

[Scoped CSS](#)

[Trigonometric functions](#)

[Individual transform properties](#)

Day 75: font palettes

posted on January 6., 2023

It's time to get me up to speed with modern CSS. There's so much new in CSS that I know too little about. To change that I've started [#100DaysOfMoreOrLessModernCSS](#). Why more or less modern CSS? Because some topics will be about cutting-edge features, while other stuff has been around for quite a while already, but I just have little to no experience with it.

Apparently, multicolored typefaces on the web are a thing. You can use and modify them in CSS.

```
@font-face {
  font-family: 'Rocher';
  src: url('/fonts/RocherColorGX.woff2');
}

h1 {
  font-family: "Rocher";
}
```

demo



Pretty cool, right? What's even cooler is that color fonts come with a default color palette and optionally with a set of alternative palettes that you can access via CSS.



The font "Rocher" comes with 11 palettes

Day 76: overwriting colors in font palettes

posted on January 9., 2023

It's time to get me up to speed with modern CSS. There's so much new in CSS that I know too little about. To change that I've started [#100DaysOfMoreOrLessModernCSS](#). Why more or less modern CSS? Because some topics will be about cutting-edge features, while other stuff has been around for quite a while already, but I just have little to no experience with it.

You can use the `override-colors` property to override colors in a font palette.

Color fonts come with one or more predefined color palettes. You can select them by using the `font-palette` property. You can also define your own color palettes or change specific colors in a palette using the `override-colors` property.

```
@font-face {
  font-family: 'Rocher';
  src: url('/fonts/RocherColorGX.woff2');
}

h1 {
  font-family: "Rocher";
}
```

That's the default font palette of the "Rocher" font.

demo - default



Using the `font-palette-value` rule you can create a new font palette. You reference the typeface you want to create the palette for using the `font-family` property. Color palettes of the "Rocher" font consist of 4 colors. We can override colors by defining the index (starting with 0) of the color and a valid color value.

Day 82: value processing

posted on January 17., 2023

It's time to get me up to speed with modern CSS. There's so much new in CSS that I know too little about. To change that I've started [#100DaysOfMoreOrLessModernCSS](#). Why more or less modern CSS? Because some topics will be about cutting-edge features, while other stuff has been around for quite a while already, but I just have little to no experience with it.

This post differs from most of the other posts because it's not about modern CSS, but about CSS fundamentals. When I was writing about [custom properties](#) and especially about [container style queries](#), I realized that I had to understand some of the basics of the language first before I could comprehend how certain properties and rules worked.

The final value of a property in CSS is the result of a multi-step calculation. In this process, the actual value of a property can come from different sources, take on different forms, and undergo adjustments.

Declared Values

A property may have multiple declared values. Each property declaration applied to an element contributes a declared value.

```
h1 {
  color: aqua;
}

#heading {
  color: rebeccapurple;
}

.heading {
  color: fuchsia;
}
```

The `color` property has 3 declared values. The cascade takes these values and chooses a single

Day 63: explicit defaulting with inherit, initial, unset, and revert

posted on December 21., 2022

It's time to get me up to speed with modern CSS. There's so much new in CSS that I know too little about. To change that I've started [#100DaysOfMoreOrLessModernCSS](#). Why more or less modern CSS? Because some topics will be about cutting-edge features, while other stuff has been around for quite a while already, but I just have little to no experience with it.

There are several CSS-wide property values you can use to specify a particular defaulting behavior for a property explicitly.

Okay, okay, I know, these keywords aren't new at all, except for `revert` maybe which is newish, but if I want to write about `revert-layer`, which is brand new, I need a basic understanding of all keywords. Also, I had the feeling that most of you, like me, don't know what all of these keywords do, and I was right. At least, if you want to trust this [poll on Mastodon](#).

Our baseline

We'll work with the following example. We have a `<div>` with a `1px solid green` border, a `red` text color, and a `10px` margin. Nested in the div is a `<h2>` with a `blue` text color.

```
<div>
  <h2>standard</h2>
</div>
```

```
div {
  color: red;
  border: 1px solid green;
  margin: 10px;
}

h2 {
  color: blue;
```

Day 74: using !important in cascade layers

posted on January 5., 2023

It's time to get me up to speed with modern CSS. There's so much new in CSS that I know too little about. To change that I've started [#100DaysOfMoreOrLessModernCSS](#). Why more or less modern CSS? Because some topics will be about cutting-edge features, while other stuff has been around for quite a while already, but I just have little to no experience with it.

In order to understand how `!important` works in cascade layers, you have to understand how `!important` works generally. The conclusion of this post might not be what you expect.

The basics

Let's start nice and easy. We have two declarations with the same specificity. The second one wins because it comes later in the document.

```
h1 {
  color: green;
}

h1 {
  color: red;
}
```

demo 1

I'm red

Adding `!important` to the first declaration increases its specificity, turning the color green.

```
h1 {
  color: green !important;
}
```




ME



BOB

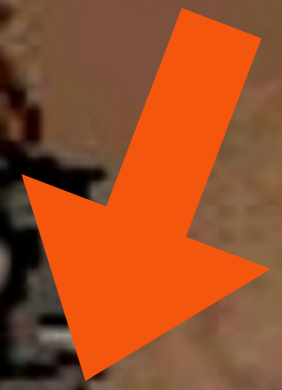


1

2

3

HUGO

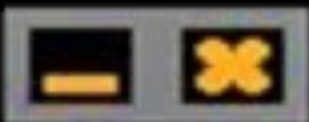


30000





DREAMCAST ❤️



COUNTER-STRIKE

www.counter-strike.net

Console
Resume Game

Return to game.

Training
Config
Save/load game

Learn how to play Counter-Strike, be prepared for the real battles.

Change Counter-Strike's video, audio, and control settings.

Save your place in the training mission, or load a past save.

Play CS
Custom Game

Get online and play Counter-Strike with others from around the world.

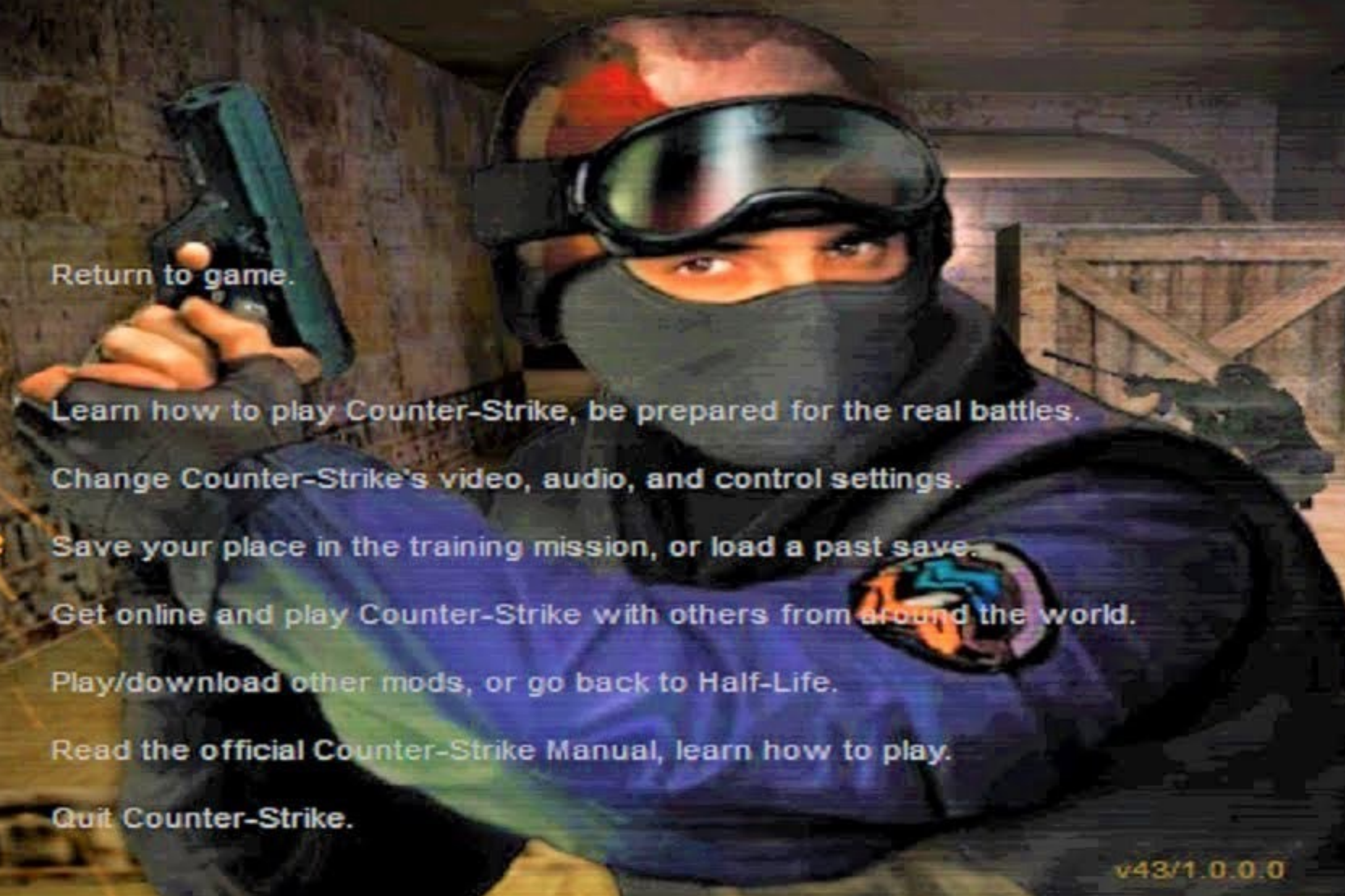
Play/download other mods, or go back to Half-Life.

How to Play

Read the official Counter-Strike Manual, learn how to play.

Quit

Quit Counter-Strike.





DARK HEADSHOTMAN

ME

THE ANARCHIST

NATURAL BORN KILLER

[—>MTM<—]

Mange Ta Merde

MANGETAMERDE

[News](#)[Members](#)[Wars](#)[Join Us](#)[FUN](#)[Downloads](#)[Screenshot](#)[Gastebuch](#)[Forum](#)[Chat](#)[Umfragen](#)[Rulez](#)[Links](#)[Login](#)[Home](#)

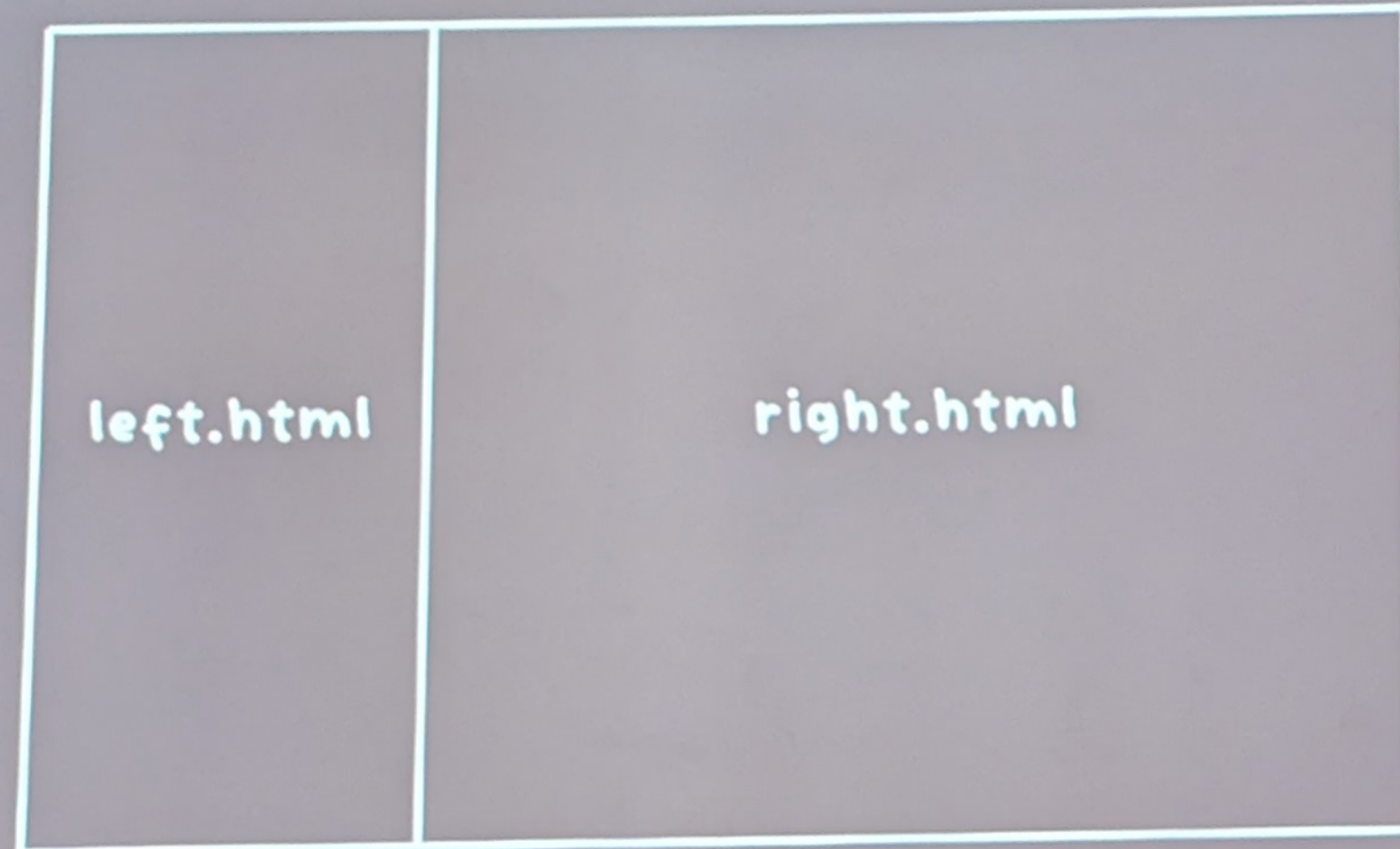
Der StrikeNet.com - Client

Advanced Combat Terminal

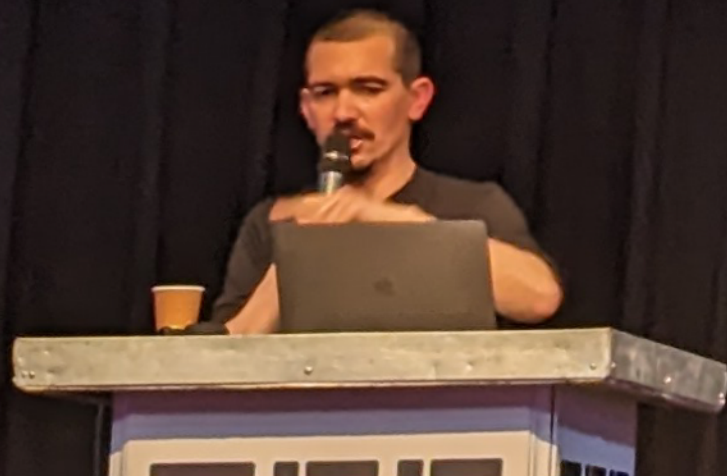
GameServer-Suche - Skinsystem

KLICKE HIER UM ZUM ULTIMATIVEM N

[-->MTM<--] (Mange Ta Merde) wurde von DHM (Dark Headshotman), ODW (O?l Dirty Wutti), NBK (Natural Born Killer), MW (Methodwutti) gegr?ndet und TA (The Anarchist), TCM (Texas Chainsaw Masacer) sind dazu gekommen. Wir haben auch noch zwei Testmembers: EPD (El Pollo Diablo), Devil.



```
<frameset cols="30%, *">  
  <frame src="left.html"></frame>  
  <frame src="right.html"></frame>  
</frameset>
```




```
1 <body bgcolor="#000000" text="#ffffff" link="#000000"
2     alink="000000" vlink="#000000">
3     <center>
4         <h1>
5             <font color="#ff0000">M</font>
6             <font color="#ff0000">A</font>
7             <font color="#ff0000">N</font>
8             <font color="#ff0000">G</font>
9             <font color="#ff0000">E</font>
10            <font color="#ff0000">T</font>
11            <font color="#ff0000">A</font>
12        </h1>
13    </center>
14 </body>
15
```

presentational attributes and elements

```
1 <center>
2   <br /><br />
3   <table width="500" border="2">
4     <tbody>
5       <tr>
6         <td align="center">
7           <marquee>
8             KLINKE HIER UM...
9           </marquee>
10        </td>
11      </tr>
12    </tbody>
13  </table>
14 </center>
15
```

center element, table layouts, and marquee 🔥

```
1  #header #left .blue {
2    color: red;
3  }
4
5
6  ul li li ol li {
7    font-weight: bold;
8  }
9
10 li {
11   font-weight: normal !important;
12 }
13
14
15
```

complex selectors and properties with high specificity



Scalable and Modular Architecture for CSS

A flexible guide to developing sites small and large.



“SMACSS is becoming one of the most useful contributions to front-end discussions in years” *

I’ve been analyzing my process (and the process of those around me) and figuring out how best to structure code for projects on a larger scale. What I’ve found is a process that works equally well for sites small and large.

Learn how to structure your CSS to allow for flexibility and maintainability as your project and your team grows.

What is it?

SMACSS (pronounced “smacks”) is more style guide than rigid framework. There is no library within here for you to download or install.

What’s in SMACSS?

Preface

1 About the Author



The Baker Street
INQUIRER



THE
WEBLOGUE

BACK
ISSUES

ABOUT
OUR PAPER

“Give me problems, give me *work*.”

In the year 1878 I took my degree of Doctor of Medicine of the University of London, and proceeded to Netley to go through the course prescribed for surgeons in the army. Having completed my studies there, I was duly attached to the Fifth Northumberland Fusiliers as Assistant Surgeon. The regiment was stationed in India at the time, and before I could join it, the second Afghan war had broken out. On landing at Bombay, I learned that my corps had advanced through the passes, and was already deep in the enemy's country.

victors & villains



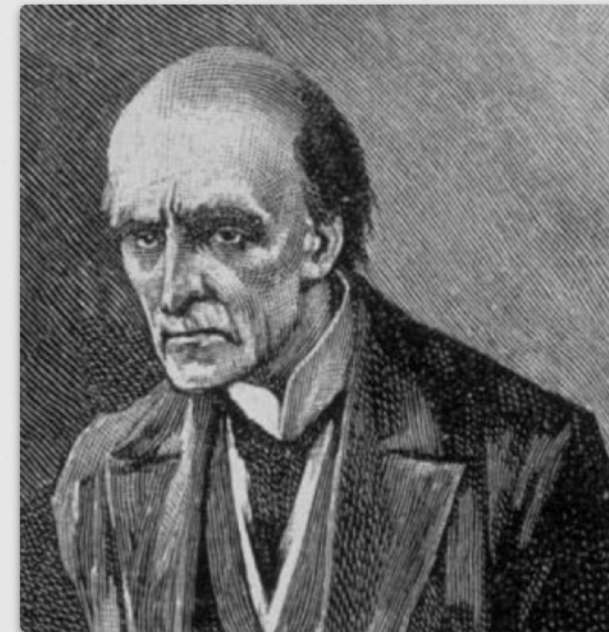
SHERLOCK
HOLMES



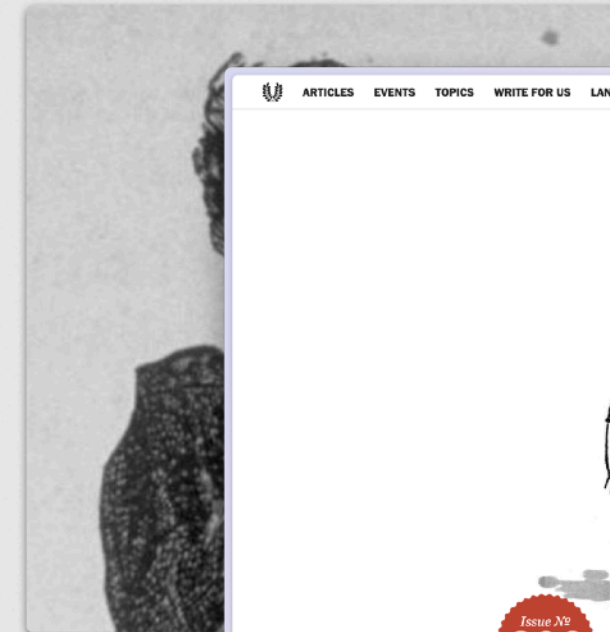
DR. JOHN HEMISH
WATSON



MYCROFT
HOLMES



PROF. JAMES
MORIARTY



Illustrations by Sidne
What r

ARTICLES EVENTS TOPICS WRITE FOR US LANGUAGE (ENGLISH) ▼

A LIST APART

Issue No. **306**

Responsive Web Design

by [Ethan Marcotte](#) · May 25, 2010
Published in CSS, Interaction Design, Layout & Grids, Mobile/Multidevice, Responsive Design

“The control which designers know in the print medium, and often desire in the web medium, is simply a function of the limitation of the printed page. We should embrace the fact that the web doesn't have the same constraints, and design for this flexibility. But first, we must 'accept the ebb and flow of things.'”

—John Allsopp, “A Dao of Web Design”

The English architect Christopher Wren once quipped that his chosen field “aims for Eternity,” and there's something appealing about that formula: Unlike the web, which often feels like aiming for next week, architecture is a discipline very much defined by its permanence.

```
1 #header #left .blue {
2   color: red;
3 }
4
5 ul li li ol li {
6   font-weight: bold;
7 }
8
9 li {
10   font-weight: normal !important;
11 }
12
13
14
15
```

complex selectors and properties with high specificity

```
16 @media (max-width: 1024px) {
17   #header #left .blue {
18     color: green;
19   }
20
21   body li {
22     font-weight: normal !important;
23   }
24 }
25
26
27
28
29
30
```

repetition of complex selectors and high specificity

```
1  .col-start__logo--highlight {
2    color: red;
3  }
4
5  .sub-list_sub-item {
6    font-weight: bold;
7  }
8
9  li {
10   font-weight: normal;
11 }
12
13
14
15
```

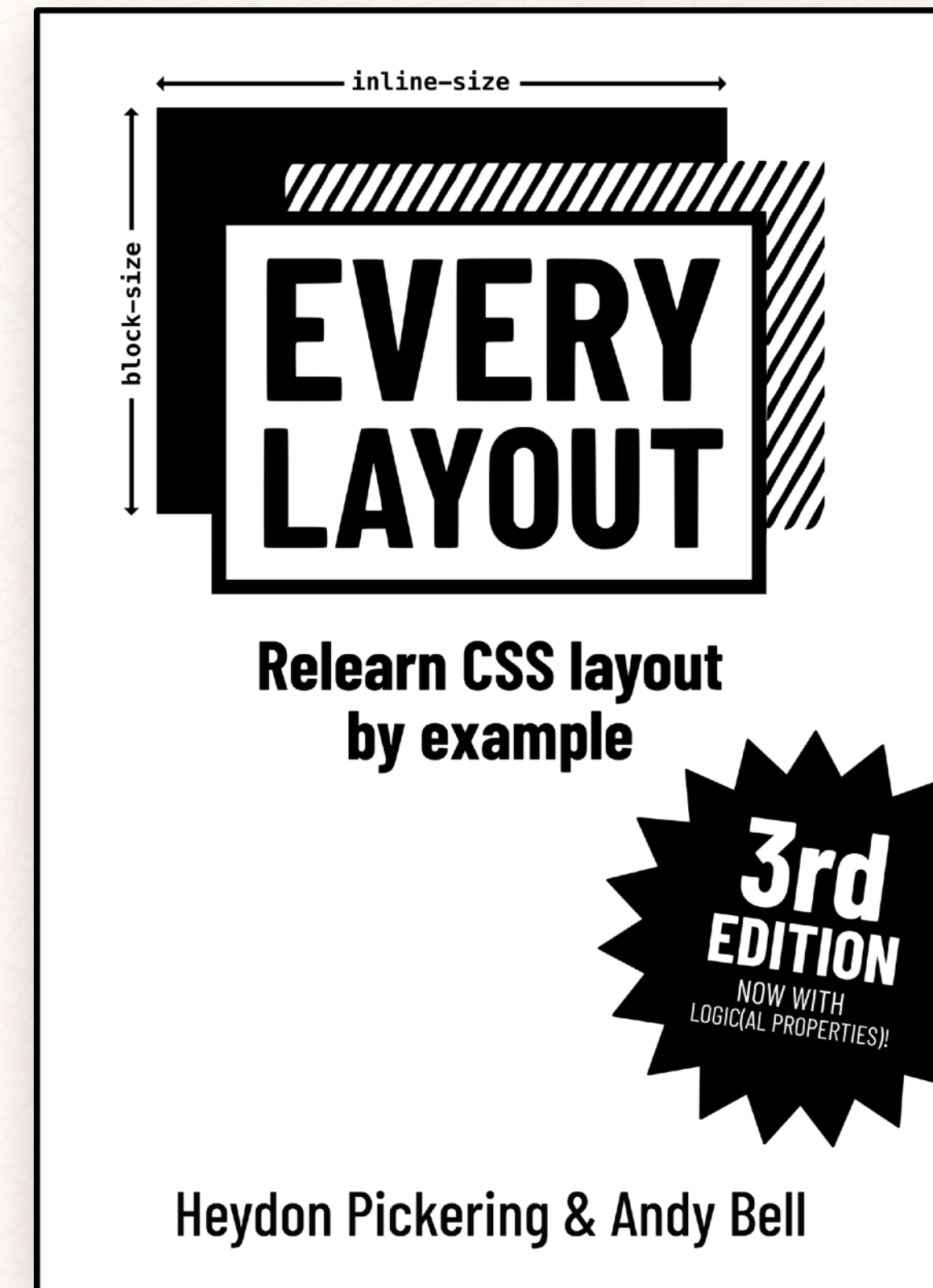
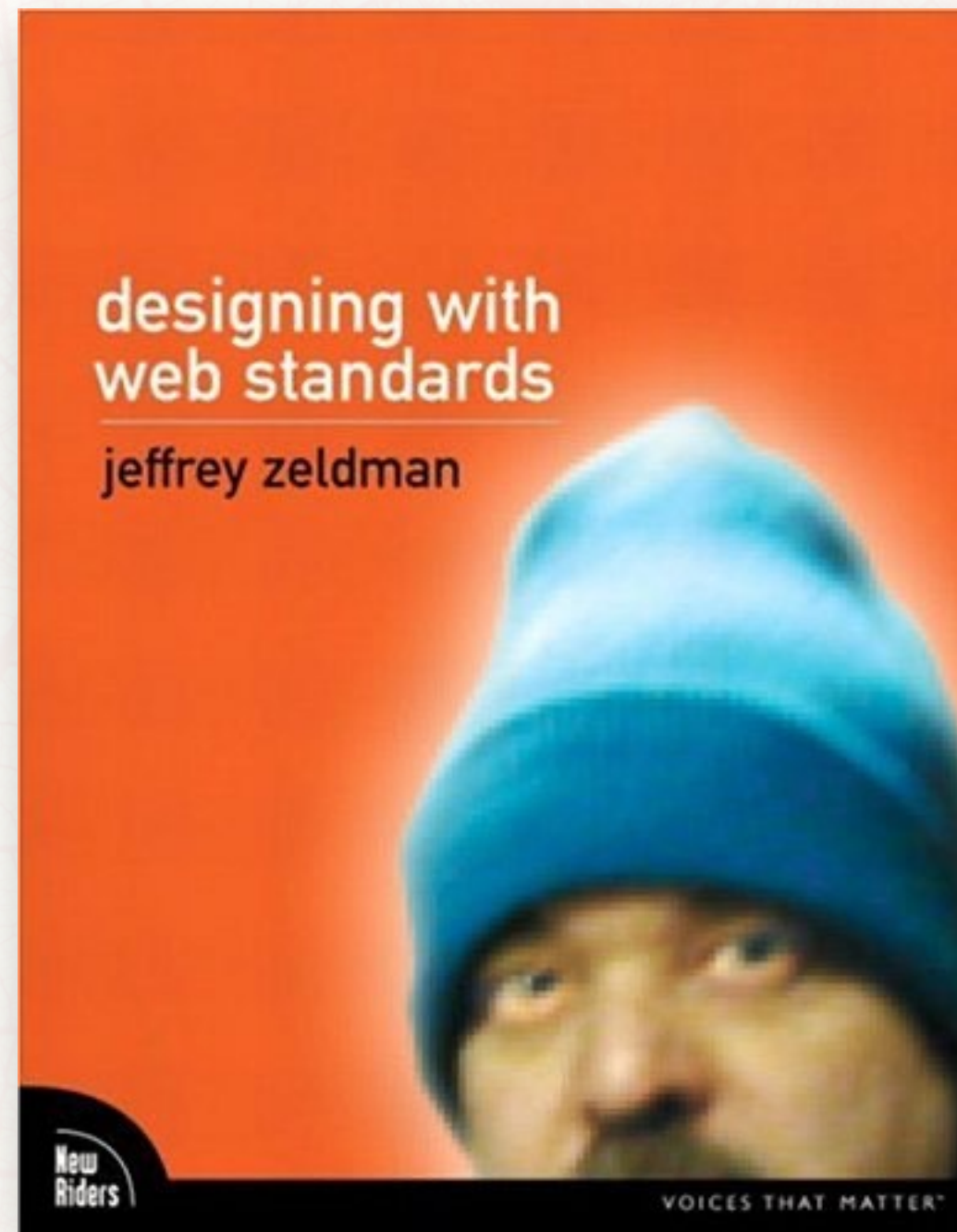
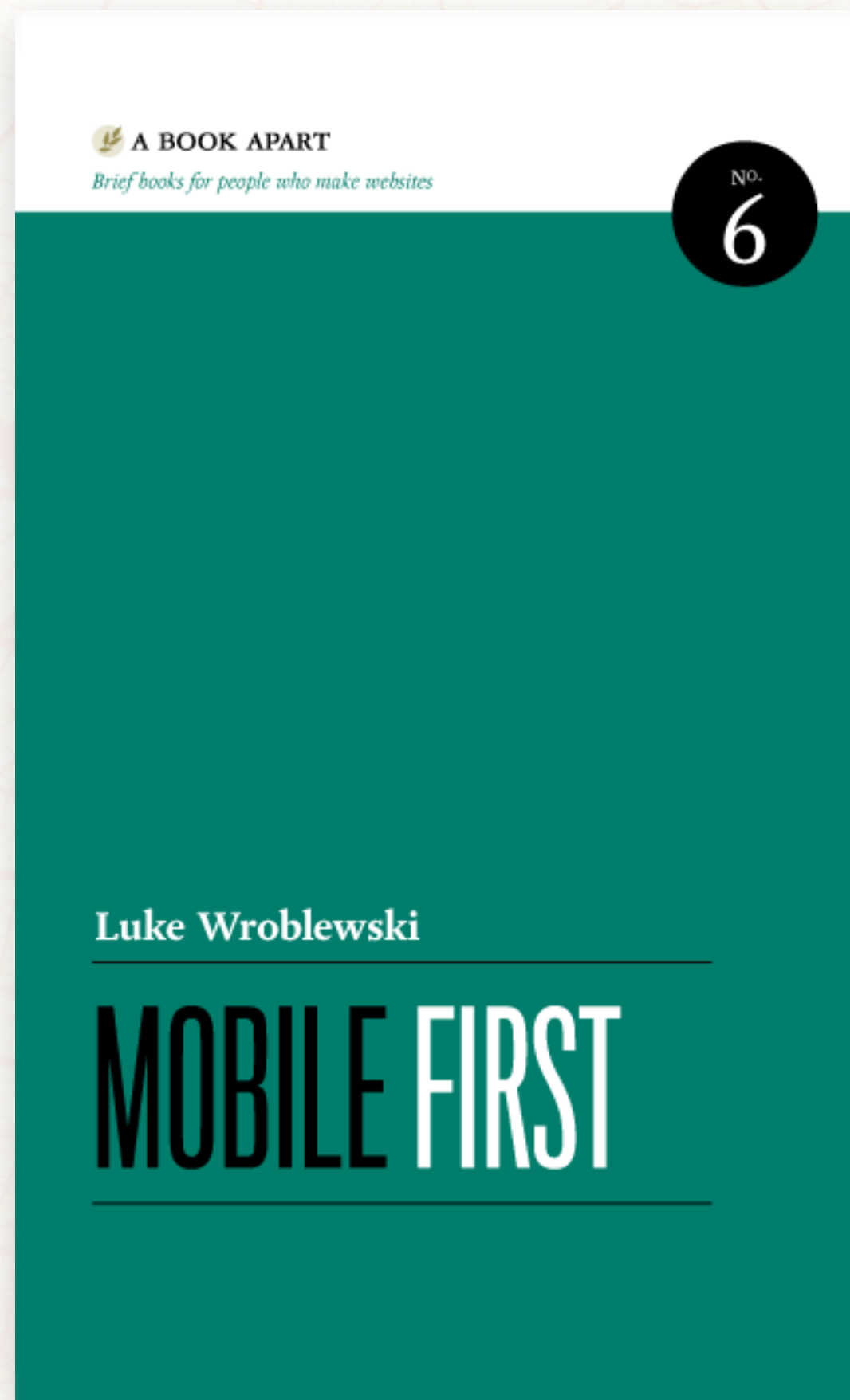
BEM syntax and low specificity selectors

```
16  @media (max-width: 1024px) {
17    .col-start__logo--highlight {
18      color: green;
19    }
20
21    .sub-list_sub-item {
22      font-weight: normal;
23    }
24  }
25
26
27
28
29
30
```

repetition of low specificity selectors

```
1  .col-start {
2    $color: red;
3
4    &__logo {
5      &--highlight {
6        color: $color
7      }
8    }
9
10   ul {
11     @include reset-list;
12   }
13 }
14
15
```

Sass syntax



ARTICLES EVENTS TOPICS WRITE FOR US LANGUAGE (ENGLISH) ▼

A LIST APART

A Dao of Web Design

by [John Allsopp](#) · April 07, 2000
Published in [Accessibility](#), [CSS](#), [Layout & Grids](#), [Typography & Web Fonts](#)

What Zen was to the 70's (most famously with motorcycle maintenance), the *Tao Te Ching* was to the 90's. From Piglet and Pooh to Physics and back, many have sought sense in applying the *Tao Te Ching* to something (the *Tao of Physics*), or something to the *Tao Te Ching* (the *Tao of Pooh*). It can be a cheap trick, but lately it has struck me that there is more than a little to be understood about web design by looking through the prism of the Tao.

Daoism is a philosophy, like Buddhism, a way of living, of being in the world, which stems from a text of great antiquity, the *Tao Te Ching*, whose 81 "chapters" enigmatically sweep across human experience, but with a strong common theme, that of harmony.

For the last couple of years, for better or worse, my life has revolved more than a little around style sheets. I write software, tutorials, and guides for them; I've answered too many questions to count about them on newsgroups and via email; I've fought for their adoption with [The Web Standards Project](#). And slowly I've come to understand web design entirely differently because of them, and to see a strong association between design and the Tao.

What I sense is a real tension between the web *as we know it*, and the web *as it would be*. It's the tension between an existing medium, the printed page, and its child, the web. And it's time to really understand the relationship between the parent and the child, and to let the child go its own way in the world.

Same old new medium?

Well established hierarchies are not easily uprooted; Closely held beliefs are not easily released; So ritual enthralls generation after generation.

Tao Te Ching: 38 Ritual

no comments yet

Share this: [Facebook](#) [Twitter](#)

Become a patron

Translations: [Russian](#) [French](#)

Recently by [John Allsopp](#)
Semantics in HTML 5
The BBC's dropping of hCalendar because of accessibility and usability concerns demonstrates that we have pushed the semantic capability of HTML far beyond what it can handle. The need to clearly and unambiguously add rich

元完全な技

Zen Garden

The Beauty of CSS Design.

The Road to Enlightenment
Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.

We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaaS, and the major browser creators.

The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?
There is a continuing need to show the power of CSS. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The HTML remains the same, the only thing that has changed is the external CSS file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. Designers and coders alike have contributed to the beauty of the web; we can always push it further.

Participation
Strong visual design has always been our focus. You are modifying this page, so strong CSS skills are necessary too, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the [CSS Resource Guide](#) for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the

select a design:
Mid Century Modern by Andrew Lohman
Garments by Dan Mall
Steel by Steffen Knoeller
Apothecary by Trent Walton
Screen Filler by Elliot Jay Stocks
Fountain Kiss by Jeremy Carlson
A Robot Named Jimmy by meltmedia
Verde Moderna by Dave Shea

archives:
Next Designs >
View All Designs

resources:
View This Design's CSS
CSS Resources
FAQ
Submit a Design
Translations

Wired News

Search: News Search

Home Business Culture Technology Politics Wired Mag Animation

Text Size: A A A A

Get Wired News in [Spanish](#), [Portuguese](#), or [Japanese](#)

QUOTE MARKS
"My joke was that I'm recession proof."
— A psychic on her profession's rising popularity with the financially downtrodden.

WIRED MAGAZINE
Issue 10.10
Read *Wired* magazine's [October issue](#), online now.
Special Offer!
[Subscribe to Wired](#)

SUBSETS
[Business](#) MORE >
E-Biz
Good buys or goodbye?
[Unwired News](#)
And wireless shall set you free

[Culture](#) MORE >
[Joystick](#)
Games games games games
[DAT's Entertainment](#)
Not the same old song and dance
[Cult of Mac](#)

TOP STORIES
updated 11:25 a.m. Oct. 11, 2002 PDT

Tracking Lil' Suzie
[A Kiddie GPS for the Masses?](#)
A startup is developing a Global Positioning System that parents can use to keep track of junior. Its main selling point is the inexpensive price tag, around \$100. By [Elisa Batista](#)

Nasdaq Down, Psychic Readings Up
Looking for a recession-proof business? Consider becoming a tarot card reader. In troubled economic times, people turn more than ever to what lies beyond the five senses. [Ned Randolph](#) reports from San Francisco.

Weather Forecasts to Believe In
Satellites that take a sideways look at the Earth's atmosphere and take more accurate measurements could help meteorologists better predict storm behavior -- and lend TV weather people some authority. By [Mark Baard](#).

Wired News: A Site for Your Eyes
Wired News has a different appearance, but the new design isn't just about look and feel. The site now complies with standards recommended by the World

Wired News on the Go Sync up, head out, read Wired News on your handheld at your leisure.

CLICK OR MISS
Turn the Page
Wired News' new look is a [radical departure](#) from previous versions. The change is as much about substance as style.

HOT OFF THE WIRE
Stories and photos from AP and Reuters feeds
[Breaking News](#)
[Image Galleries](#)

<advertisement>
The GigaLaw Guide to Internet Law A unique, comprehensive, user-friendly guide to the legal issues relevant to conducting...
Fast Company: Save over 80% on the how-to magazine for people who make business work. Plus get a FREE gift!

Hier lies
clearfix:after { }

R.I.P.


2004 - 2013

ITCSS Home About Services Speaking Case Studies Contact

26 November, 2018

ITCSS x Skillshare

Back in February 2018, Scott Sullivan, Partnerships Team Lead at **Skillshare**, sent me an email asking if I'd be interested in collaborating on an official ITCSS video course in conjunction with them.



Write Better CSS

REACH
SPECIFICITY
ITCSS
EXPLICITNESS

Sign up and learn ITCSS today!

Projects



Hi there, I'm Harry. I am an **award-winning** Consultant Web Performance Engineer, designer, developer, writer, and speaker from the UK. I **write, Tweet, speak,** and **share code** about measuring and improving site-speed. You **should hire me.**

Follow @csswizardry

Suffering? Fix It

CUBE CSS

What does CUBE CSS stand for?
How does CUBE CSS compare to other methodologies?
Does CUBE CSS scale?
A high-level overview

Principles

CSS

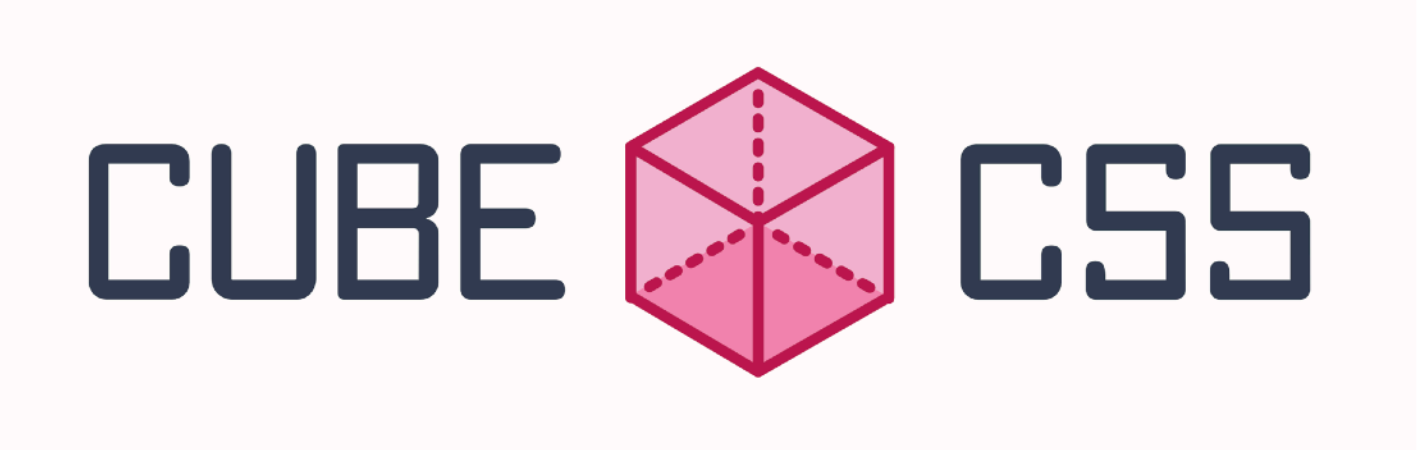
Composition

Utility

Block

Exception

Grouping



CUBE CSS is a CSS methodology that's orientated towards simplicity, pragmatism and consistency. It's designed to work *with* the medium that you're working in—often the browser—rather than *against* it.

This project is very much a work in progress so this documentation will continue to evolve.

What does CUBE CSS stand for?

CUBE stands for **Composition Utility Block Exception**. CSS stands for **Cascading Style Sheets**. The CSS part comes straight from the language: **CSS**.

The name of this methodology gives the game away straight away: it's an **extension of CSS** rather

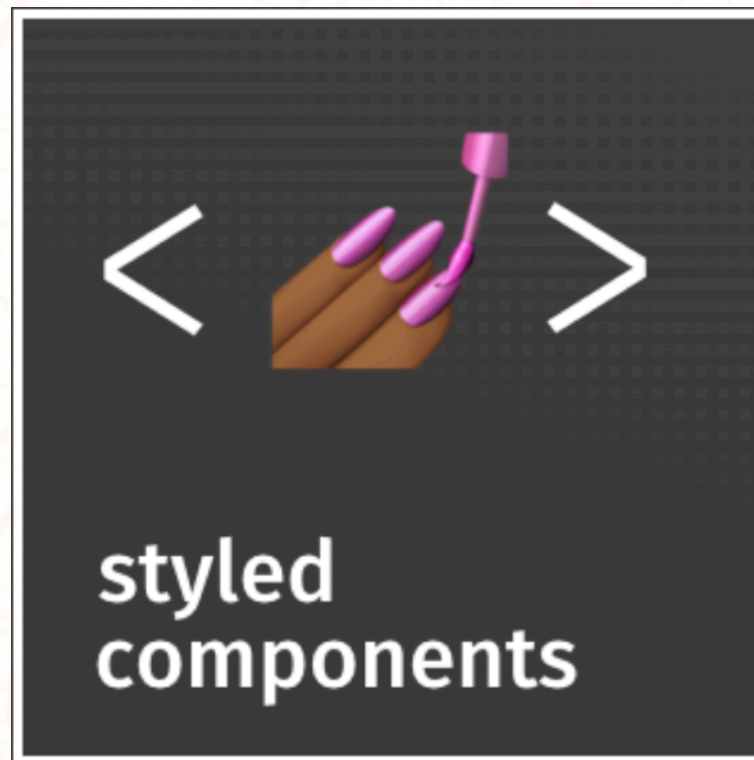


ATOMS MOLECULES ORGANISMS TEMPLATES PAGES

Let's explore each stage in more detail.

atoms

Atoms are the basic building blocks of matter. Applied to web interfaces, atoms are our HTML tags, such as a form label, an input or a button.



 tailwindcss



Custom Properties

```
1  :root {
2    --color-a: hsl(210 65% 20%);
3    --color-b: hsl(166 65% 20%);
4  }
5
6  body {
7    color: var(--color-a);
8    border-color: var(--color-b);
9  }
10
11
12
13
14
15
```

simple usage of custom properties


```
1  :root {
2    --color-a-h: 210;
3    --color-a-s: 65%;
4    --color-a-l: 20%;
5
6    --color-a: hsl(var(--color-a-h)
7                  var(--color-a-s)
8                  var(--color-a-l));
9    --color-b: hsl(166 var(--color-a-s) var(--color-a-l));
10 }
11
12 body {
13   color: var(--color-a);
14   border-color: var(--color-b);
15 }
```

reusing colour parameters



```
1  :root {
2    --color-a: #123456;
3    --color-b: hsl(from var(--color-a) 166 s l);
4  }
5
6  body {
7    color: var(--color-a);
8    border-color: var(--color-b);
9  }
10
11
12
13
14
15
```

relative colour syntax

Day 92: relative color syntax

posted on January 31., 2023

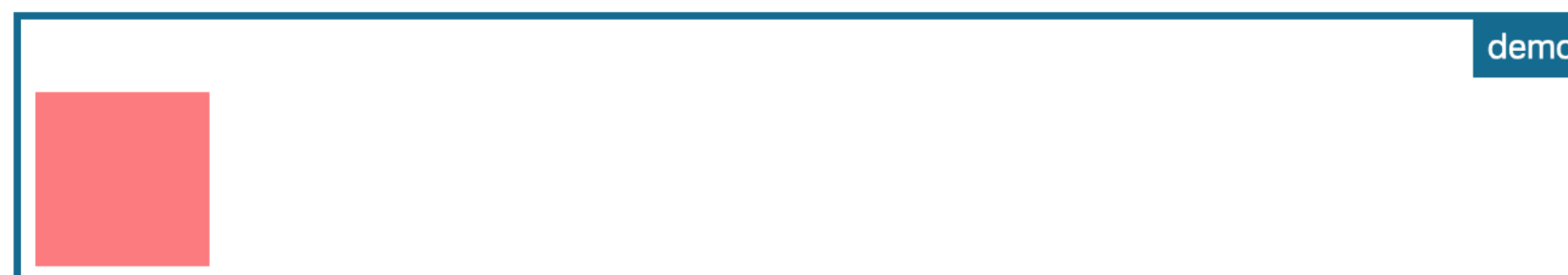
It's time to get me up to speed with modern CSS. There's so much new in CSS that I know too little about. To change that I've started [#100DaysOfMoreOrLessModernCSS](#). Why more or less modern CSS? Because some topics will be about cutting-edge features, while other stuff has been around for quite a while already, but I just have little to no experience with it.

With the relative color syntax we can modify existing colors using color functions. If an origin color is specified, each color channel can either be directly specified, or taken from the origin color and modified.

For example, we can take a HEX color and add opacity using the `rgb()` color function and the `from` keyword.

```
div {
  --color: #FF0000;
}

div {
  background-color: rgb(from var(--color) r g b / 50%);
}
```



Or we can take the color and replace a specific channel.

```
div {
  background-color: rgb(from var(--color) r g 150);
}
```

Using CSS Variables Correctly

Custom Properties have the potential to change how we write and think about CSS.

Posted by Mike on 18th June 2017 (about a 14 minute read).

CSS Variables (also known as Custom Properties) are now supported in all modern browsers and people are starting to use them in production. This is great but they're different from variables in preprocessors and I've already seen some examples of people using them without considering the advantage they offer.

They have the potential to change how we write and think about CSS. I thought I'd do a few quick demos that show some good and bad ways to use CSS variables, and how their differences from preprocessors might change how we structure CSS.

How do they differ?

Firstly how do they differ? The main difference is CSS variables can change. This

```
1  ul {
2    display: flex;
3    flex-direction: column;
4  }
5
6  @media (min-width: 768px) {
7    ul {
8      flex-direction: row;
9    }
10 }
11
12
13
14
15
```

duplicate declaration of the flex-direction property

```
1  ul {
2    --flex-direction: column;
3
4    display: flex;
5    flex-direction: var(--flex-direction);
6  }
7
8  @media (min-width: 768px) {
9    ul {
10     --flex-direction: row;
11   }
12 }
13
14
15
```

single declaration of the flex-direction property

```
1  ul {
2    display: flex;
3    flex-direction: var(--flex-direction, column);
4  }
5
6  @media (min-width: 768px) {
7    ul {
8      --flex-direction: row;
9    }
10 }
11
12
13
14
15
```

single declaration of the flex-direction property.



Parkpickerl im grünen Prater

Seit 4. Juli gelten im Prater die [flächendeckende Kurzparkzone](#) und das Parkpickerl für Bezirks-Bewohner*innen.

```
1 <wm-card>
2   <h3 slot="heading">
3     Parkpickerl im grünen Prater
4   </h3>
5
6   <div slot="content">
7     <p>
8       Seit 4. Juli gelten im Prater die
9       flächendeckende Kurzparkzone und
10      das Parkpickerl für
11      Bezirks-Bewohner*innen.
12    </p>
13  </div>
14
15  
18 </wm-card>
```

markup of a card component

Slots

Name	Beschreibung
<code>heading</code>	Überschrift
<code>media</code>	Bild oder Video
<code>eyecatcher</code>	Störer
<code>content</code>	Inhalt der Card
<code>postcontent</code>	Zusätzlicher Inhalt am Ende der Card, z.B. Tags

CSS API

Property	Beschreibung
<code>--card-background</code>	Hintergrundfarbe der Card
<code>--card-border</code>	Rahmen der Card
<code>--card-eyecatcher-spacing</code>	Abstand des Störers zum Rand
<code>--card-eyecatcher-padding</code>	Innenabstand im Störer
<code>--card-eyecatcher-fontsize</code>	Schriftgröße im Störer
<code>--card-media-background</code>	Hintergrundfarbe
<code>--card-media-color</code>	Textfarbe
<code>--card-spacing</code>	Innenabstand in der Card
<code>--card-spacing-large</code>	Innenabstand in der Card auf großen Bildschirmen
<code>--card-shadow</code>	Schatten um die Card

```
1  wm-card {
2    --_color: darkorchid;
3
4    --card-background: hsl(from var(--_color) h s 99%);
5    --card-border: 1px solid var(--_color);
6    --card-shadow: none;
7    --card-spacing-large: 1rem;
8  }
9
10
11
12
13
14
15
```

tweaking “public” properties of the card component



Parkpickerl im grünen Prater

Seit 4. Juli gelten im Prater die [flächendeckende Kurzparkzone](#) und das Parkpickerl für Bezirks-Bewohner*innen.



Parkpickerl im grünen Prater

Seit 4. Juli gelten im Prater die [flächendeckende Kurzparkzone](#) und das Parkpickerl für Bezirks-Bewohner*innen.

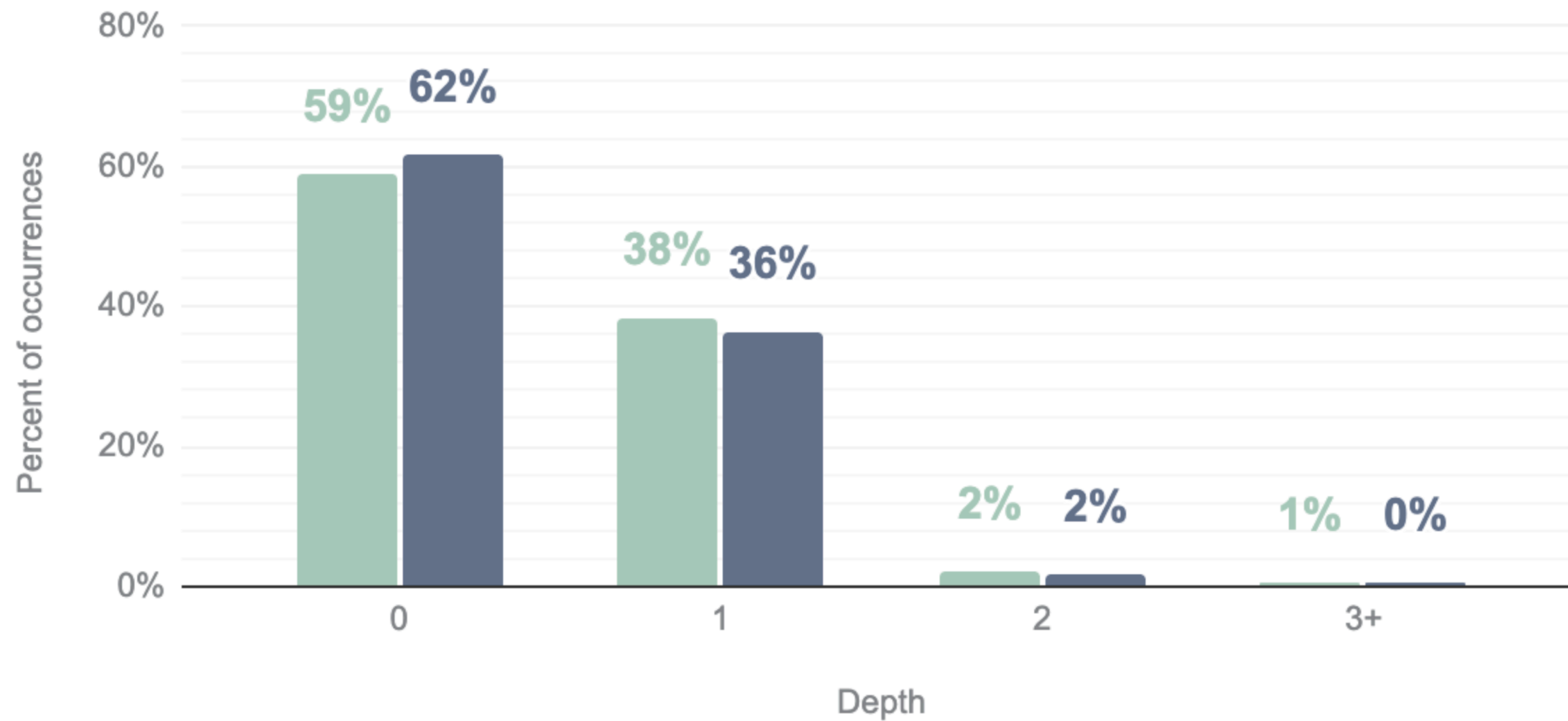
```
1 <wm-card style="--card-shadow: none">
2   ...
3 </wm-card>
4
5
6
7
8
9
10
11
12
13
14
15
```

card web component variation and style attribute

Custom property depth

Web Almanac 2022: CSS

desktop mobile



```
1 <wm-card style="
2     --_color: darkorchid;
3     --card-bg: hsl(from var(--_color) h s 99%);
4     --card-border: 1px solid var(--_color);
5     --card-shadow: none;
6     --card-spacing-large: 1rem;">
7 </wm-card>
8
9
10
11
12
13
14
15
```

multiple custom properties defined via the style attribute

```
1 <wm-card class="cssday"
2 ...
3 </wm-card>
4
5
6
7
8
9
10
11
12
13
14
15
```

```
16 .cssday {
17   --_color: darkorchid;
18   --card-bg: hsl(from var(--_color) h s 99%);
19   --card-border: 1px solid var(--_color);
20   --card-shadow: none;
21   --card-spacing-large: 1rem;
22 }
23
24
25
26
27
28
29
30
```

```
1 <wm-card style="--card-style: cssday">
2   ...
3 </wm-card>
4
5 <style>
6 @container style(--card-style: cssday) {
7   .card {
8     --_color: darkorchid;
9     --card-background: hsl(from var(--_color) h s 95%);
10    --card-border: 4px solid var(--_color);
11    --card-shadow: none;
12    --card-spacing-large: 1rem;
13  }
14 }
15 </style>
```

using a container style query to apply multiple properties at once



```
1 <div class="blockquote">
2   <blockquote>
3     Underline your fucking links you sociopaths.
4   </blockquote>
5
6   <p><em>Heydon Pickering</em></p>
7 </div>
8
9
10
11
12
13
14
15
```

simple blockquote component

“Underline your fucking links you sociopaths.”

Heydon Pickering

```
1 <div class="blockquote" style="--quote-type: highlight">
2   <blockquote>
3     Underline your fucking links you sociopaths.
4   </blockquote>
5
6   <p><em>Heydon Pickering</em></p>
7 </div>
8
9
10
11
12
13
14
15
```

variation of the blockquote component

```
1  @container style(--quote-type: highlight) {
2    blockquote {
3      --font-size: 4rem;
4    }
5
6    blockquote::after {
7      display: none;
8    }
9
10   ...
11  }
12
13
14
15
```

style query for the blockquote component



Underline your fucking links you sociopaths.

Heydon Pickering



BLOG

SPEAKING

PROJECTS

ABOUT ME

We happy?

Look, just because I don't be givin' no man a foot massage don't make it right for Marsellus to throw Antwone into a glass motherfuckin' house, fuckin' up the way he talks. Motherfucker do that shit to me, he better paralyze my ass, 'cause I'll kill the motherfucker, know what I'm sayin'?

“Underline your fucking links you sociopaths.”

Heydon Pickering

Well, the way they make shows is, they make one show. That show's called a pilot. Then they show that show to the people who make shows, and on the strength of that one show they decide if they're going to make more shows. Some pilots get picked and become television programs. Some don't, become nothing. She starred in one of the ones that became nothing.

Now that we know who you are, I know who I am. I'm not a mistake! It all makes sense! In a comic, you know how you can tell who the arch-villain's going to be? He's the exact opposite of the hero. And most times they're friends, like you and me! I should've known way back when... You know why, David? Because of the kids. They called me Mr Glass.

“You looked like a crab having a sad wank.”

Heydon Pickering

You think water moves fast? You should see ice. It moves like it has a mind. Like it knows it killed the world once and got a taste for murder. After the avalanche, it took us a week to climb out. Now, I don't know exactly when we turned on each other, but I know that seven of

```
1 <article style="--quote-type: highlight">
2   ...
3   <div class="blockquote">
4     <blockquote>
5       Underline your fucking links you sociopaths.
6     </blockquote>
7
8     <p><em>Heydon Pickering</em></p>
9   </div>
10  ...
11 </article>
12
13
14
15
```

children inheriting style config from parent article

MATUZO VIĆ

FRONT-END / AIYY / HTML / CSS



BLOG

SPEAKING

PROJECTS

ABOUT ME

We happy?

Look, just because I don't be givin' no man a foot massage don't make it right for Marsellus to throw Antwone into a glass motherfuckin' house, fuckin' up the way he talks. Motherfucker do that shit to me, he better paralyze my ass, 'cause I'll kill the motherfucker, know what I'm sayin'?

“Underline your fucking links you sociopaths.

Heydon Pickering

Well, the way they make shows is, they make one show. That show's called a pilot. Then they show that show to the people who make shows, and on the strength of that one show they decide if they're going to make more shows. Some pilots get picked and become television programs. Some don't, become nothing. She starred in one of the ones that became nothing.

Now that we know who you are, I know who I am. I'm not a mistake! It all makes sense! In a comic, you know how you can tell who the arch-villain's going to be? He's the exact opposite of the hero. And most times they're friends, like you and me! I should've known way back when... You know why, David? Because of the kids. They called me Mr Glass.

“You looked like a crab having a sad wank.

Heydon Pickering

You think water moves fast? You should see ice. It moves like it has a mind. Like it knows it killed the world once and got a taste for murder. After the avalanche, it took us a week to

MATUZO VIĆ

FRONT-END / AIYY / HTML / CSS



BLOG

SPEAKING

PROJECTS

ABOUT ME

We happy?

Look, just because I don't be givin' no man a foot massage don't make it right for Marsellus to throw Antwone into a glass motherfuckin' house, fuckin' up the way he talks. Motherfucker do that shit to me, he better paralyze my ass, 'cause I'll kill the motherfucker, know what I'm sayin'?

Well, the way they make shows is, they make one show. That show's called a pilot. Then they show that show to the people who make shows, and on the strength of that one show they decide if they're going to make more shows. Some pilots get picked and become television programs. Some don't, become nothing. She starred in one of the ones that became nothing.

Now that we know who you are, I know who I am. I'm not a mistake! It all makes sense! In a comic, you know how you can tell who the arch-villain's going to be? He's the exact opposite of the hero. And most times they're friends, like you and me! I should've known way back when... You know why, David? Because of the kids. They called me Mr Glass.

“Underline your fucking links you sociopaths.”

Heydon Pickering

“You looked like a crab having a sad wank.”

Heydon Pickering

You think water moves fast? You should see ice. It moves like it has a mind. Like it knows it killed the world once and got a taste for murder. After the avalanche, it took us a week to climb out. Now, I don't

know exactly when we turned on each other, but I know that seven of us survived the slide... and only five made it out. Now we took an oath, that I'm breaking now. We said we'd say it was the snow that killed the other two, but it wasn't. Nature is lethal but it doesn't hold a candle to

```
1  :root {
2    --theme: light;
3  }
4
5  .site-header {
6    --nav-style: main;
7  }
8
9  .intro {
10   --paragraph-style: initial-letter;
11 }
12
13 .selected-posts {
14   --card-style: large;
15 }
```

applying global, per page, and per section styles in CSS

MATUZOVIĆ



FRONT-END / A11Y / HTML / CSS



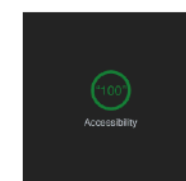
[Blog](#) [Speaking](#) [Projects](#)
[About Me](#)

Hello, I'm Manuel Matuzović, a frontend developer, consultant, accessibility auditor, writer, and speaker from Vienna, currently living in Graz. I'm a fan of web standards and a believer in progressive enhancement and inclusive design. I'm specialized in web accessibility and large-scale CSS.

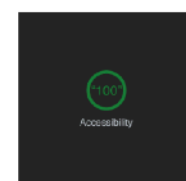
Upcoming events

- 1. Rheinwerk Spotlight: Barrierefreie Webseiten - Online
June 5., 2023
- 2. CSS Day - Amsterdam
June 8., 2023

Selected blog posts



The most inaccessible site
posted on May 31., 2019

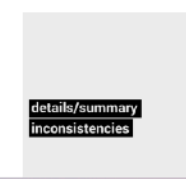


The most inaccessible site
posted on May 31., 2019

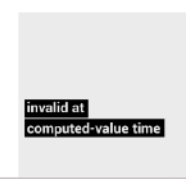


My current HTML boilerplate
posted on April 9., 2021

Recent blog posts



[details/summary inconsistencies](#)
posted on April 21., 2023




[Invalid at computed-value time](#)
posted on April 18., 2023

Code File Edit Selection View Go Run Terminal Window Help

http://localhost:5501

MATUZOVIĆ

FRONT-END / A11Y / HTML / CSS



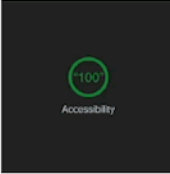
[Blog](#) [Speaking](#) [Projects](#)
[About Me](#)

Hello, I'm Manuel Matuzović, a frontend developer, consultant, accessibility auditor, writer, and speaker from Vienna, currently living in Graz. I'm a fan of web standards and a believer in progressive enhancement and inclusive design. I'm specialized in web accessibility and large-scale CSS.

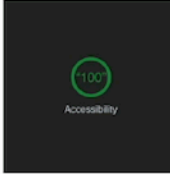
Upcoming events

1. Rheinwerk Spotlight: Barrierefreie Webseiten - Online
June 5., 2023
2. CSS Day - Amsterdam
June 8., 2023


Selected blog posts



The most inaccessible site
posted on May 31., 2019

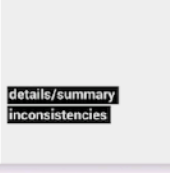


The most inaccessible site
posted on May 31., 2019

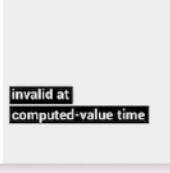


My current HTML boilerplate
posted on April 9., 2021

Recent blog posts



[details/summary inconsistencies](#)
posted on April 21., 2023



[Invalid at computed-value time](#)
posted on April 18., 2023

```
# config.css 6 X
css > 0_api > # config.css > .site-header
1  /* Layers */
2
3  @layer base, helpers, components, utility;
4
5  /* Frame */
6
7  :root {
8  /* --theme: dark; */
9  }
10
11 .site-header {
12 /* --nav-style: main; */
13 }
14
15 /* Homepage */
16 .intro {
17 /* --paragraph-style: initial-letter; */
18 }
19
20 .upcoming-events {
21 /* --list: reset; */
22 /* --card-style: event; */
23 /* --grid: 2col; */
24 }
25
26 .selected-posts {
27 /* --card-style: large; */
28 /* --grid: 1col; */
29 /* --grid-gap: var(--space-xl); */
30 }
31
32 .recent-posts {
33 /* --card-style: small; */
34 /* --grid: 2col; */
35 }
36
```

Ln 12, Col 24 Spaces: 2 UTF-

```
1  .card {
2    --_card-columns: 6rem 1fr;
3    --_card-gap: 0 1.2rem;
4
5    display: grid;
6    gap: var(--_card-gap);
7    grid-template-columns:
8      var(--_card-columns);
9  }
10
11  .card .content {
12    --_card-content: none;
13
14    display: var(--_card-content);
15  }
```

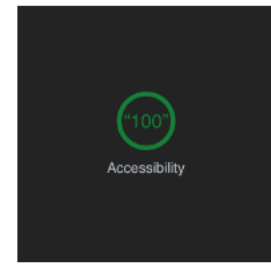
card default styles

```
16  @container (min-width: 64em) {
17    .card {
18      --_card-columns: 10rem 1fr;
19    }
20
21    .content {
22      --_card-content: block;
23    }
24  }
25
26
27
28
29
30
```

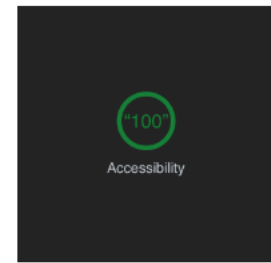
card on viewports larger than 64em

Default

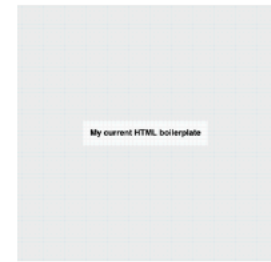
Selected blog posts



The most inaccessible site
posted on May 31., 2019



The most inaccessible site
posted on May 31., 2019



My current HTML boilerplate
posted on April 9., 2021

```
1  .card {
2    --_card-columns: 6rem 1fr;
3    --_card-gap: 0 1.2rem;
4
5    display: grid;
6    gap: var(--_card-gap);
7    grid-template-columns:
8      var(--_card-columns);
9  }
10
11  .card .content {
12    --_card-content: none;
13
14    display: var(--_card-content);
15  }
```

card default styles

```
16  @container style(--card-style: large) {
17    .card {
18      --_card-columns: 10rem 1fr;
19    }
20    .content {
21      --_card-content: block;
22    }
23  }
24
25
26  @container (min-width: 64em) {
27    .card {
28      --card-style: large;
29    }
30  }
```

style query + card on viewports larger than 64em

```
1  .card-large {  
2    --card-style: large;  
3  }  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15
```

large card styling


```
1  const card = document.querySelector('.card')
2
3  // Get custom property value
4  card.style.getPropertyValue('--card-style')
5
6  // Set custom property value
7  card.style.setProperty('--card-style', 'large')
8
9
10
11
12
13
14
15
```

getting and setting custom properties in JavaScript

Cascade Layers

“Layers help us reclaim declarative control of the intent behind our CSS without removing helpful information about the specificity of selectors. We’re managing the cascade instead of removing it.”

Miriam Suzanne

“Selectors are there for a reason, and we should take advantage of their full potential. Instead, we’ve been restricting what selectors are allowed and enforcing more and more limited conventions in an attempt to somehow avoid the cascade

Miriam Suzanne

...but the cascade is unavoidable; it's the core of the language.

There will always be selector conflicts, and there will always be an algorithm to resolve those conflicts. All we've been doing is denying ourselves access to one of the coolest features on the web platform."

Miriam Suzanne


```
1  @layer base {
2    p { margin-block-start: 0; }
3  }
4
5  @layer components {
6    .card__content {
7      margin-block-start: 1rem;
8    }
9  }
10
11 @layer utility {
12   .u-mbs0 {
13     margin-block-start: 0;
14   }
15 }
```

cascade layers

```
1  @layer base {
2    html.js.js.js input[type="text"]#yolo#yolo {
3      border: 2px solid blue;
4    }
5  }
6
7  @layer components {
8    input {
9      border-color: green;
10   }
11  }
12
13  /* Result: border-color: green; */
14
15
```

a layer defined later in the document overrides a layer defined earlier


```
1  @layer components {
2    .card {
3      border: 20px solid var(--border-color, currentColor);
4    }
5
6    .card:has(img[loading="lazy"]) {
7      --border-color: red;
8    }
9
10   .card--large:has(img) {
11     --border-color: blue;
12   }
13 }
14
15 /* Result: border-color: red; */
```

the specificity of the selector in line 5 is higher than in line 9

```
1  @layer components {
2    .card {
3      border: 20px solid var(--border-color, currentColor);
4    }
5
6    .card:has(img[loading="lazy"]) {
7      --border-color: red;
8    }
9
10   .card--large:has(img[loading="lazy"]) {
11     --border-color: blue;
12   }
13 }
14
15 /* Result: border-color: blue; */
```

the specificity of the selector in line 5 is higher than in line 9

```
1  @layer components {
2    .card {
3      border: 20px solid var(--border-color, currentColor);
4    }
5
6    .card:has(img[loading="lazy"]) {
7      --border-color: red;
8    }
9
10   .card--large:has(img) {
11     --border-color: blue !important;
12   }
13 }
14
15 /* Result: border-color: blue; */
```

the specificity of the selector in line 5 is higher than in line 9

```
1  @layer components {
2    .card {
3      border: 20px solid var(--border-color, currentColor);
4    }
5
6    .card:where(:has(img[loading="lazy"])) {
7      --border-color: red;
8    }
9
10   .card--large:has(img) {
11     --border-color: blue;
12   }
13 }
14
15 /* Result: border-color: blue; */
```

the specificity of the selector in line 5 is higher than in line 9

```
1  @layer components {
2    @layer default {
3      .card {
4        border: 20px solid var(--border-color, currentColor);
5      }
6
7      .card:has(img[loading="lazy"]) {
8        --border-color: red;
9      }
10   }
11
12   .card--large:has(img) {
13     --border-color: blue;
14   }
15 }
```

base styles in a nested default layer, variations unlayered

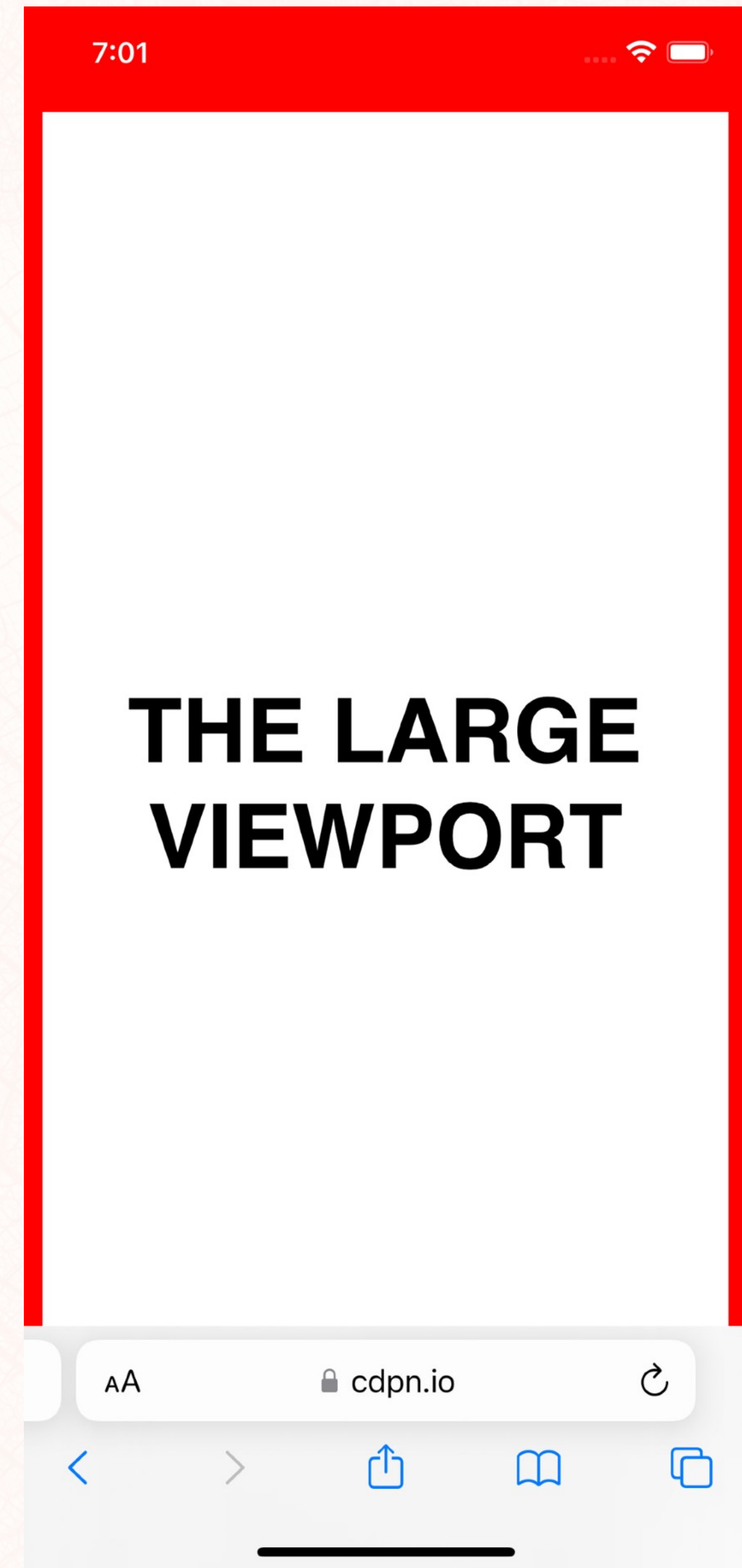
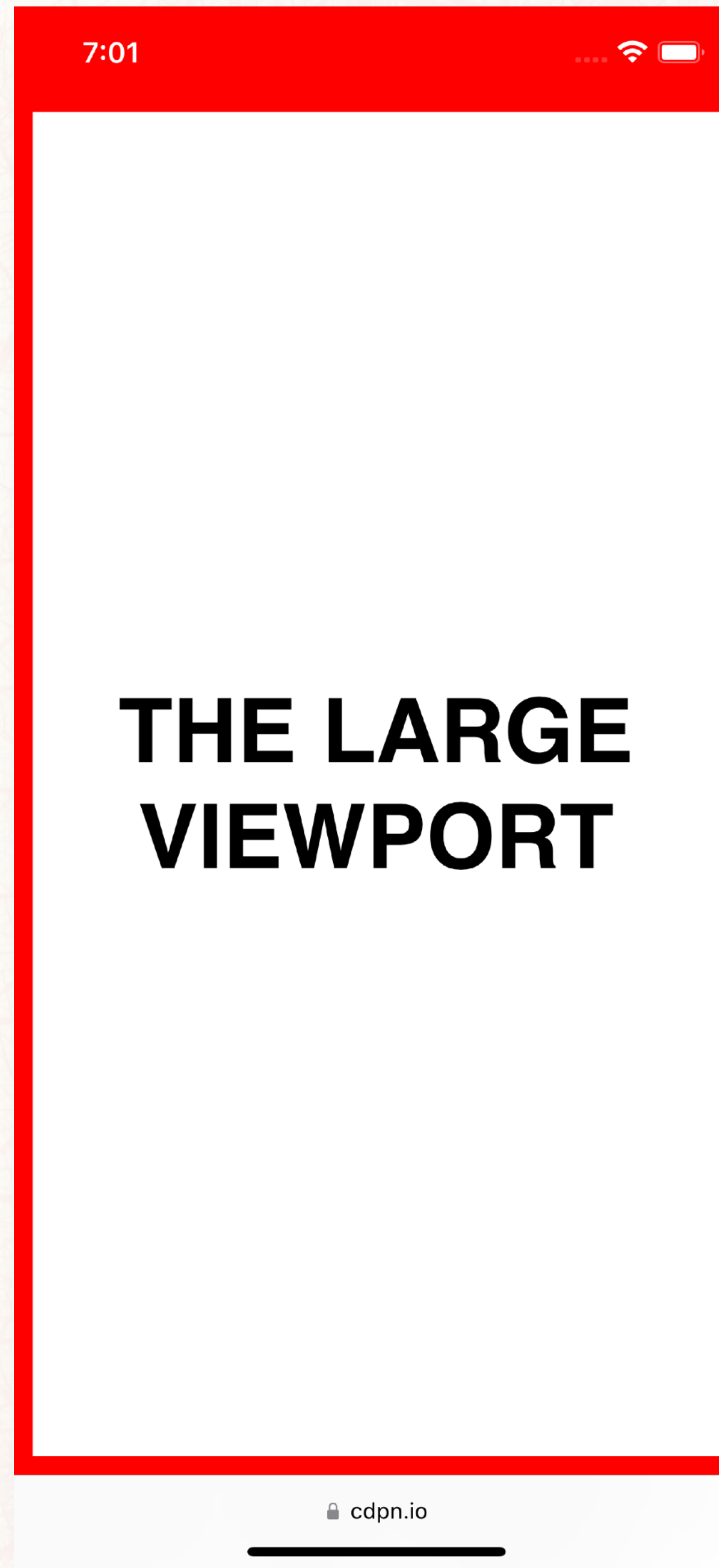
```
1  @layer components {
2    @layer default {
3      .card img {
4        border: 20px solid var(--border-color, currentColor);
5      }
6
7      .card:has(img[loading="lazy"]) { --border-color: red; }
8    }
9
10   @layer large {
11     .card--large:has(img) {
12       --border-color: blue;
13     }
14   }
15 }
```

base styles in a nested default layer, variations in dedicated layers

- I don't need Sass anymore
- I went from BEM (Block Element Modifier) to BAPI (Block & API)
- I spend more time in CSS because I write less classes
- I embrace the cascade
- Custom Properties – Declarations - Config



JavaScript



```
1  let vh = window.innerHeight * 0.01
2  document.documentElement.style.setProperty('--vh', `${vh}px`)
3
4  <html style="--vh:11.7px;" lang="en">
5    <head>
6      <style>
7        body {
8          height: calc(var(--vh, 1vh) * 100);
9        }
10     </style>
11   </head>
12 </html>
13
14
15
```

JS hack for dynamic viewport height in CSS

```
1  body {  
2    height: 100dvh;  
3  }  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15
```

dynamic viewport height in CSS

Day 38: vh, svh, lvh, and dvh

posted on November 16., 2022

It's time to get me up to speed with modern CSS. There's so much new in CSS that I know too little about. To change that I've started [#100DaysOfMoreOrLessModernCSS](#). Why more or less modern CSS? Because some topics will be about cutting-edge features, while other stuff has been around for quite a while already, but I just have little to no experience with it.

Using the viewport unit `vh` in desktop browsers is usually straight-forward, `100vh` matches the height of the viewport. On mobile that's different because the viewport height changes depending on whether or not certain user interface elements are visible, `100vh` doesn't always match the height of the viewport.

On mobile we have the small viewport and the large viewport. According to the spec, the small viewport is *"the viewport sized assuming any UA [User Agent/Browser] interfaces [for example the address bar] that are dynamically expanded and retracted to be expanded"*.

Chrome File Edit View History Bookmarks Profiles Tab Window Help

localhost:8080

Home Blog About me TIL

Manuel Matuzovic

Hi! I'm a frontend developer from Vienna. I'm specialized in **HTML**, **accessibility**, and **CSS layout and architecture**.

If you want to get in touch, write me an [email](#) or find me on [Mastodon](#).

Recent posts

details/summary inconsistencies

April 21., 2023

Scott O'Hara wrote a [fantastic blog post about the details and summary elements](#) last year. He explains that there are a lot of oddities and inconsistencies, and he backs his statements with [detailed testing](#).

Invalid at computed-value time

April 18., 2023

I rewatched [Lea Verous's talk about custom properties](#) recently and learned something I missed the first time I watched it.

It's very likely that...

April 17., 2023

I repeatedly see certain bad practices in HTML that ironically contain clues for implementing them properly in their class names or in the way they're built. In this evergreen post, I collect them.

[More blog posts](#)

```
1  html {  
2    scrollbar-gutter: stable;  
3  }  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15
```

stable scrollbar gutter in CSS

Chrome File Edit View History Bookmarks Profiles Tab Window Help

localhost:8080

Home Blog About me TIL

Manuel Matuzovic

Hi! I'm a frontend developer from Vienna. I'm specialized in **HTML**, **accessibility**, and **CSS layout and architecture**.

If you want to get in touch, write me an [email](#) or find me on [Mastodon](#).

Recent posts

details/summary inconsistencies

April 21., 2023

Scott O'Hara wrote a [fantastic blog post about the details and summary elements](#) last year. He explains that there are a lot of oddities and inconsistencies, and he backs his statements with [detailed testing](#).

Invalid at computed-value time

April 18., 2023

I rewatched [Lea Verous's talk about custom properties](#) recently and learned something I missed the first time I watched it.

It's very likely that...

April 17., 2023

I repeatedly see certain bad practices in HTML that ironically contain clues for implementing them properly in their class names or in the way they're built. In this evergreen post, I collect them.

[More blog posts](#)

Day 20: the scrollbar-gutter property

posted on October 21., 2022

It's time to get me up to speed with modern CSS. There's so much new in CSS that I know too little about. To change that I've started [#100DaysOfMoreOrLessModernCSS](#). Why more or less modern CSS? Because some topics will be about cutting-edge features, while other stuff has been around for quite a while already, but I just have little to no experience with it.

The `scrollbar-gutter` property allows you decide whether content within an element fills the total available space or if it stops at the scrollbar gutter. The scrollbar gutter is the space between the inner border edge and the outer padding edge of an element used by the scrollbar.

By default, if the content in an element is not overflowing, there's no gutter.

Overflowing content

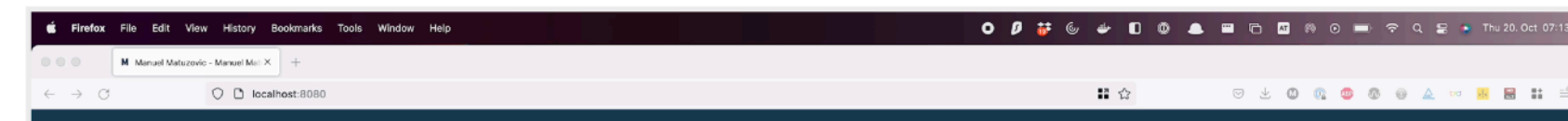
Day 25: scrollbar gutters in body and html

posted on October 28., 2022

It's time to get me up to speed with modern CSS. There's so much new in CSS that I know too little about. To change that I've started [#100DaysOfMoreOrLessModernCSS](#). Why more or less modern CSS? Because some topics will be about cutting-edge features, while other stuff has been around for quite a while already, but I just have little to no experience with it.

When I wrote about the [scrollbar-gutter property](#), my first thought was "omg! I'll put this in my reset stylesheet and use it on the `<body>` by default". I wanted to do that in order to prevent the page from "jumping" when switching from a long to a short page, a page with overflow to one without.

Here's a quick demo to illustrate the issue.





A velvet sofa, airy built-ins, and a touch of boucle



Can't decide how to decorate the main space in your home? Here are many...



When it comes to home renovations, bathrooms are usually the most...



I decided it was time to do something interesting on the wall above the...

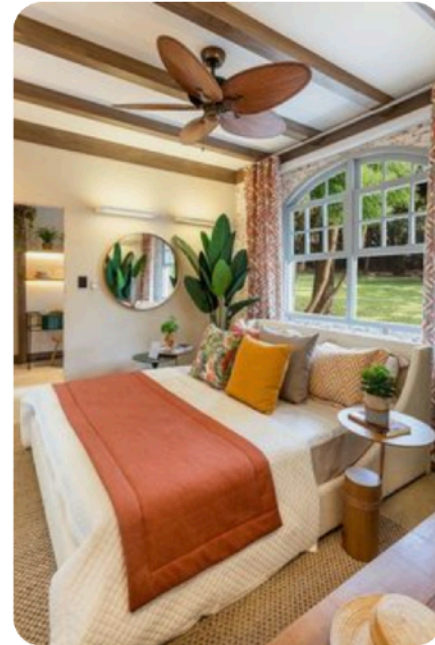


NestInOrder | We are passionate about decluttering and organizing space at...

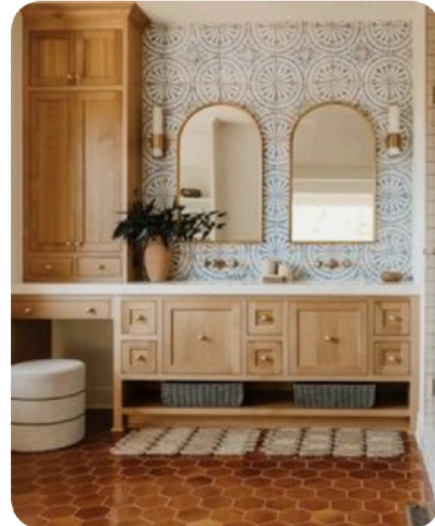
NO Nest In Order



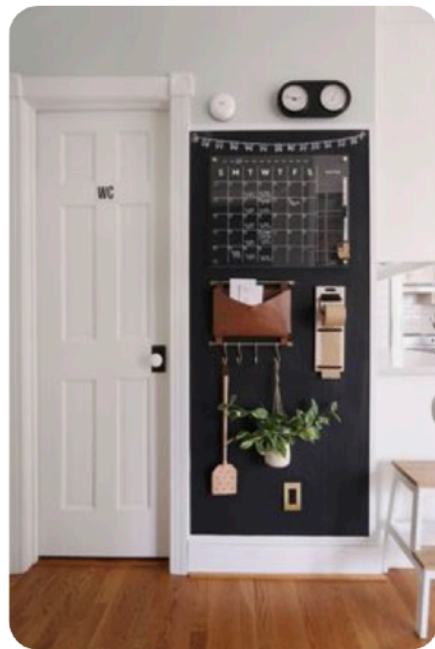
Colour and creativity flourish in George Hewitt's family home, inspiring her...



Having the perfect Master Bedroom it's a chance to really reflect your individu...



Warming your space up can be as easy as one, two, warm tones! From amber...



You will love this acrylic calendar! Acrylic is the most amazing writing...

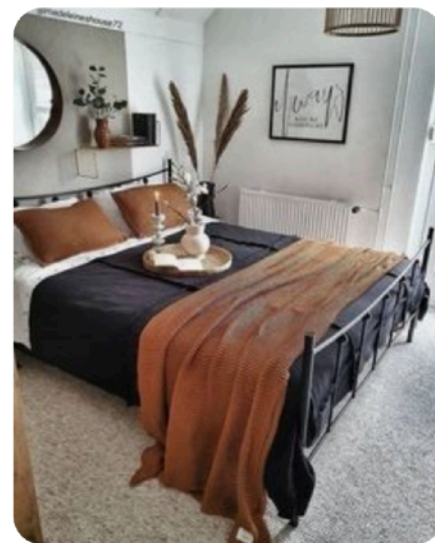


Click for a list of the decor from last week's living room reveal, including ou...



Modern Bathroom|woshroom decor ideas...

a home decor

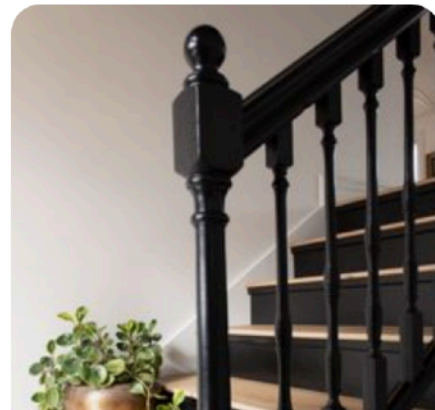


One of the simplest and most stylish ways to incorporate some boho vibes...



Hy.this is modern home decor trends here I will show you Modern living roo...

H home decor interior design



```
1  ol {
2    display: grid;
3    gap: 1rem;
4    grid-template-columns: repeat(auto-fill, minmax(16rem, 1fr));
5    grid-template-rows: masonry;
6    masonry-auto-flow: next;
7  }
8
9
10
11
12
13
14
15
```

masonry layout in native CSS without JavaScript

Day 71: the masonry keyword

posted on January 2., 2023

It's time to get me up to speed with modern CSS. There's so much new in CSS that I know too little about. To change that I've started [#100DaysOfMoreOrLessModernCSS](#). Why more or less modern CSS? Because some topics will be about cutting-edge features, while other stuff has been around for quite a while already, but I just have little to no experience with it.

The `masonry` keyword allows you to create masonry grid layouts.

If you want to create a masonry layout, all you have to do is turn one of the grid axes into a masonry axis using the `masonry` keyword.

```
<ol>
<li>
```

Day 72: the masonry-auto-flow property

posted on January 3., 2023

It's time to get me up to speed with modern CSS. There's so much new in CSS that I know too little about. To change that I've started [#100DaysOfMoreOrLessModernCSS](#). Why more or less modern CSS? Because some topics will be about cutting-edge features, while other stuff has been around for quite a while already, but I just have little to no experience with it.

If you're creating a masonry layout, the packing algorithm puts items into the column with the most space by default. This can cause accessibility issues. The `masonry-auto-flow` property gives us control over the automatic placement of items in a masonry layout.

In the following layout, you can see how the tile for "Day 63" is placed in the middle column. Since day 64 is in the last column of the first row, you'd expect day 63 to be in the first column of the second row. If you compare the heights of the items in the first column, you can see that the middle column is the one with the most available space. That's why the algorithm placed day 63 there.

http://localhost:5501

MATUZO VIĆ

FRONT-END / A11Y / HTML / CSS



Show modal

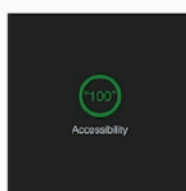
Hello, I'm Manuel Matuzović, a frontend developer, consultant, accessibility auditor, writer, and speaker from Vienna, currently living in Graz. I'm a fan of web standards and a believer in progressive enhancement and inclusive design. I'm specialized in web accessibility and large-scale CSS.

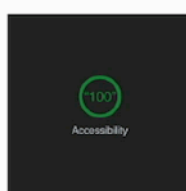
Upcoming events

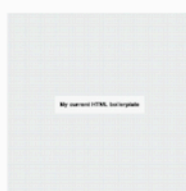
June 5. Rheinwerk Spotlight: Barrierefreie Webseiten - Online

June 8. CSS Day - Amsterdam

Selected blog posts

 **The most inaccessible site**
posted on May 31., 2019

 **The most inaccessible site**
posted on May 31., 2019

 **My current HTML boilerplate**
posted on April 9., 2021

- BLOG
- SPEAKING
- PROJECTS
- ABOUT ME

```
1  html:has(dialog[open]) {  
2    overflow: clip;  
3  }  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15
```

prevent scrolling when there's an open dialog on the page

http://localhost:5501

MATUZO VIĆ



FRONT-END / A11Y / HTML / CSS



- BLOG
- SPEAKING
- PROJECTS
- ABOUT ME

Show modal

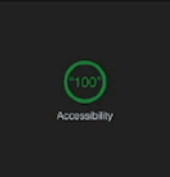
Hello, I'm Manuel Matuzović, a frontend developer, consultant, accessibility auditor, writer, and speaker from Vienna, currently living in Graz. I'm a fan of web standards and a believer in progressive enhancement and inclusive design. I'm specialized in web accessibility and large-scale CSS.

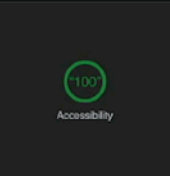
Upcoming events

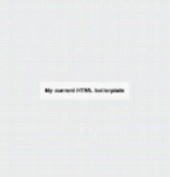
June
5. Rheinwerk Spotlight: Barrierefreie Webseiten - Online

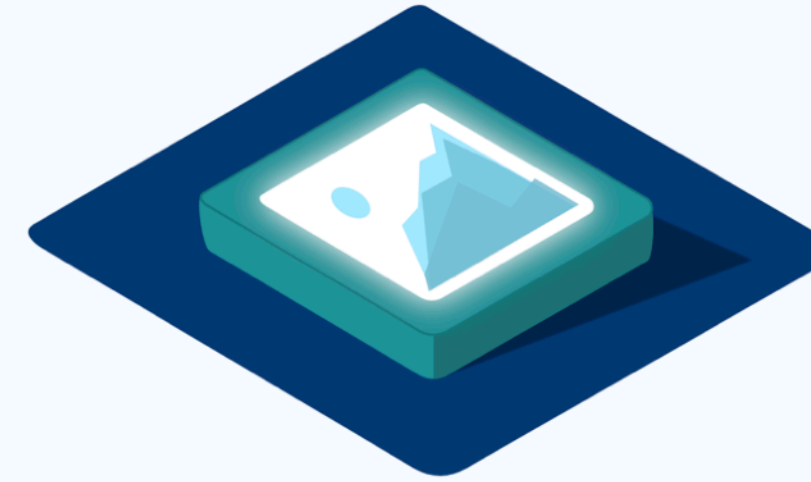
June
8. CSS Day - Amsterdam

Selected blog posts

 **The most inaccessible site**
posted on May 31., 2019

 **The most inaccessible site**
posted on May 31., 2019

 **My current HTML boilerplate**
posted on April 9., 2021



Building a lightbox with the Dialog element

7 min read. *Posted on February 8, 2023.*

From the [Polypane blog](#)

Table of contents

- # The requirements
- # [Demo](#)
- # ▶ [The code](#)
- # [Future improvements](#)

The dialog element lets you overlay content on top of your site in something called the "top layer", which is a special layer that sits above all other content. This way you never have to worry about your dialog being covered by other elements on the page due to z-index issues. While, like the name suggests, it's mostly used for dialogs, it can be used for any kind of overlay content, including lightboxes.

Recently we've been helping out with the [DevtoolsTips](#) website, adding Polypane tips and going through the list of issues to add and improve some of the features.

By using our site you agree to our [privacy policy](#) and [disclaimer](#) ×

While the site has a ton of screenshots, they were shown relatively small and inline.

If you want to be kept up to date with new articles, CSS resources and tools, join our newsletter.

Your e-mail [Subscribe](#)

Or subscribe elsewhere:

[@Polypane](#)

[RSS](#)



BUBBLE
BUBBLE
BUBBLE
BUBBLE
BUBBLE
BUBBLE
BUBBLE
BUBBLE
BUBBLE
BUBBLE



```
1  li:hover {
2    scale: 1.5 1.3;
3  }
4
5  li:has(+ li:hover), li:hover + li {
6    scale: 1.4 1.2;
7  }
8
9  li:has(+ li + li:hover), li:hover + li + li {
10   scale: 1.3 1.1;
11 }
12
13 li:has(+ li + li + li:hover), li:hover + li + li + li {
14   scale: 1.1 1;
15 }
```

previous element selector using :has()

Blog Bio Books Contact

Adrian Roselli

Crypto Machine Learning Token SME

CSS-only Widgets Are Inaccessible

Updated March 31, 2023, originally posted March 25, 2023; [4 Comments](#)

Usually,

Interactive widgets powered with only CSS are relatively common as people are playing with all the ways CSS can respond to, or create, interactions. CodePen contests are a great venue to see these experiments (which hopefully are never moved to a live human-facing project).

The problem with many of these examples is they need to convey their state (such as expanded or not), properties (think of relationships), or values, and (sometimes) dynamic names. While CSS is ideal to show these visually, they need to be passed to accessibility APIs so they can be conveyed to users of assistive technologies.

Not everything that relies on CSS for interaction is inaccessible. There are always going to be [exceptions](#), but those are (and should be) rare.

Examples

INAC
inacCSS-onlyible
IBLE

I originally titled this *InacCSS-onlyible*. I even made this typographically, er, distinct image. Then I realized it was silly and will instead use the neologism in a talk so I can hear the groans IRL.

SMASHING MAGAZINE

Articles Books Workshops Conferences Membership Job Board Newsletter Podcasts More

Search articles...

Accessibility UX CSS JavaScript Performance Design Figma Wallpapers React Vue Round-Ups Web Design Guides Business Career Privacy Jump to all articles

Stephanie Eckles / JUN 14, 2021 / [2 comments](#)

When CSS Isn't Enough: JavaScript Requirements For Accessible Components

16 min read CSS, JavaScript, Frameworks, Accessibility

Share on [Twitter](#), [LinkedIn](#)

QUICK SUMMARY - Spoiler alert: tooltips, modals, tabs, carousels, and dropdown menus are some of the user interface components that require more than CSS. To ensure accessibility of your interface, JavaScript is a necessary addition to accomplish focus management, respond to keyboard events, and toggle ARIA attributes.

As the author of [ModernCSS.dev](#), I'm a big proponent of CSS solutions. And, I love seeing the clever ways people use CSS for really out-of-the-box designs and interactivity! However, I've noticed a trend toward promoting "CSS-only" components using methods like the "[checkbox hack](#)". Unfortunately, hacks like these leave a significant amount of users unable to use your interface.

This articles covers several common components and why CSS isn't sufficient for covering accessibility by detailing the JavaScript requirements. These requirements are based on the [Web Content Accessibility Guidelines \(WCAG\)](#) and

ABOUT THE AUTHOR

[Stephanie Eckles](#) is a front-end focused software engineer. She's the author of [ModernCSS.dev](#) which provides modern solutions to old CSS problems. She is the ... [More about Stephanie](#)

Email Newsletter

Your email

[Weekly tips on front-end & UX.](#)
Trusted by 200,000+ folks.

Thank you ❤️