

CSS Overview Condensed

Selectors, Integration, Inheritance, Cascading

R. Scott Granneman & Jans Carton

© 2009 R. Scott Granneman • v3.4 2023-10-10 • Licensed CC BY-SA 4.0



Slides

granneman.com/downloads/web-dev/CSS-Overview-Condensed.pdf

chnsa.ws/css-condensed-slides



Notes

granneman.com/downloads/web-dev/CSS-Overview-Condensed.txt

chnsa.ws/css-condensed-notes



All of my presentations

granneman.com/presentations/all-presentations

chnsa.ws/all-presentations

Scott Granneman

[Writing](#)[Presentations](#)[Teaching](#)[Web Dev](#)[Tech](#)[Personal](#)[Site Map](#)[Search](#)

Scott Granneman

contributed
in St. Louis &
plans, devel
[presentation](#)

Key S

Pages you're

Personal

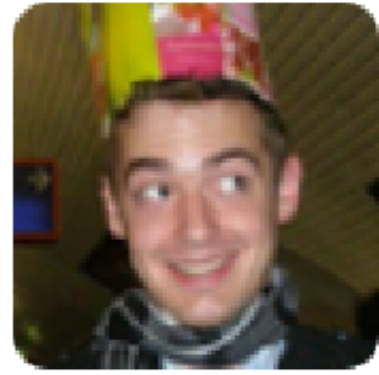
- [Publications](#)
- [Summer Vacation 2007: The Plains](#)

Teaching

- [Web Development Key Links](#)
- Wash U: [Website Design & Development Syllabus](#)
- Webster U (Comp Sci): [Intro to Web Programming Syllabus](#)

in a Web development company in St. Louis, Missouri. Scott has written six books, for SecurityFocus and *Linux Magazine*. He is an Adjunct Professor at Washington University in technology, security, web development, & the Internet. As a Principal of [WebSanity](#), he states using an open source, UNIX-based [content management system](#). He has given speaking to people.

[Interviews](#)[Ladue Chapel](#)[Exploring HTML5 & CSS](#)[Advanced CSS](#)[HTML to CSS to Responsive Web Design](#)[Responsive Web Design for Designers](#)[Web Design](#)[Archived Presentations](#)[Creative Commons License](#)



Gary Bernhardt

@garybernhardt



Follow

Watching me do CSS is like watching a drunk toddler in zero gravity repair a watch with mittens on. Also the watch randomly catches fire.

RETWEETS

59

LIKES

99



12:36 PM - 10 Nov 2015



Some developers really do *not* like CSS

“This was always the central mistake in the design of CSS. It was solving a problem people just didn’t really have. ... CSS was a disaster and has almost certainly wasted incredible amounts of time and money that could have been better spent if a better solution had come along in time.” —stormbrew on Hacker News

Those people are wrong



“Programmers *expect* CSS to operate within the patterns they’ve learned for other languages, and when it doesn’t, the frustration begins.” —Elaina Natario



“Mistaking ‘simple’ for ‘easy’ will only lead to heartache. [S]ome programmers coming to CSS for the first time heard it’s simple, so they assume it’s easy. But then when they try to use it, it doesn’t work. It must be the fault of the language, because they know that they are smart, and this is supposed to be easy. So they blame the language. They say it’s broken. And so they try to ‘fix’ it by making it conform to a more programmatic way of thinking.”
—Jeremy Keith



“What developers IMO fail to get is that CSS is so different because it’s a language to express design and they don’t understand the basics of design. Almost all programming languages are about logic and taught at all CS / programming schools. But design mostly isn’t and is much more complex than people think first...” — maigret, Hacker News



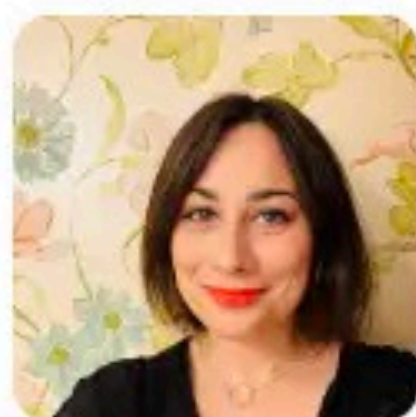
Mia || Miriam @MiriSuza... 12/4/20, 11:51 AM

CSS has been **carefully** &
systematically designed to solve a set of
problems so complex no other language
even tries. There are no comparisons to
make.

Meanwhile, the basics **look** so
declarative & clear that people don't even
think of it as "programming".



Harry Roberts



Sophie

@sophie@social.lol

If you think CSS is some kind of "problem" that needs solving with frameworks, perhaps first consider that maybe the problem is that you don't know how to use it

Nov 12, 2023, 07:39 · 🌐 · ↻ 145 · ★ 80



“CSS is the only language that gets
blamed when the author is bad.”
—Alex Riviere

History

CSS 1	December 1996
CSS 2	May 1998
CSS 2.1	July 2007
CSS 3	June 1999–Now
CSS 4	2012–Now

CSS versions & their release dates

CSS 1 & 2 each were one big specification document

CSS 3 & 4 are not each one large single spec

Instead, they are divided into many separate documents called *modules*

Currently 50+ modules!

Different modules have different statuses

For the status of each, see [www.w3.org/Style/CSS/
current-work](http://www.w3.org/Style/CSS/current-work)

W3C indicates the maturity of specifications by a status code. The CSS working group uses the following, from *least* to *most stable*:

Abbreviation	Full name
FPWD	First Public Working Draft
WD	Working Draft
CR	Candidate Recommendation
PR	Proposed Recommendation
REC	Recommendation
SPSD	Superseded Recommendation

The following code indicates a document that is not intended to become a standard:

Abbreviation	Full name
NOTE	Working Group Note

The names are defined in [section 6](#) of the W3C process document. A REC is what is normally referred to as a 'standard.' W3C encourages everyday use starting from CR.

W3C CSS Working Group: Colors & Status Codes

Can I use grid

? ⚙ Settings

2 results found

#

CSS Grid Layout - CR

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for layout into columns and rows using a set of predictable sizing behaviors

Global

69.53% + 5.62% = 75.15%

unprefixed:

69.53%

Current aligned

Usage relative

Date relative

Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			1 49			10.2			
	2 15	55	60	10.1	47	10.3		4.4	
2 11	16	56	61	11	48	11	all	56	61
		57	62	TP	49				
		58	63		50				
		59	64						

Notes

Known issues (2)

Resources (12)

Feedback

Why CSS?

97.2% of all websites use CSS

Why?

As of Oct. 2023

HTML	Structure & meaning
CSS	Presentation & layout
JavaScript	Behavior & interaction

Separation of Concerns: A design principle that separates a computer program into distinct sections, or *concerns*, each focusing on a specific resource or set of information

Note: HTML, CSS, & JavaScript can sometimes overlap their functions!

HTML

1

<h1>At
the Mountains of Madness</h1>

2

<p>By H. P.
Lovecraft</p>

3

<h2><font face="'Iowan Old Style', Georgia, serif"
size="+2">I</h2>

4

<p>I am forced into
speech because men of science have refused to follow my advice
without knowing why. It is altogether against my will that I tell
my reasons for opposing this contemplated invasion of the
antarctic—with
melting of the
my warning may

5

<p>Doubt of the real
facts, as I must reveal them, is inevitable; yet if I suppressed
what will seem extravagant and incredible there would be nothing
left.</p>

6

<p>

7

Note:
People who use Comic Sans should get a visit from Cthulhu.

8

</p>

9

CSS

JS

At the Mountains of Madness

By H. P. Lovecraft

I

I am forced into speech because men of science have refused to follow my advice without knowing why. It is altogether against my will that I tell my reasons for opposing this contemplated invasion of the antarctic—with its vast fossil hunt and its wholesale boring and melting of the ancient ice-cap—and I am the more reluctant because my warning may be in vain.

Doubt of the real facts, as I must reveal them, is inevitable; yet if I suppressed what will seem extravagant and incredible there would be nothing left.

Note: People who use Comic Sans should get a visit from Cthulhu.

Separate content (HTML) from presentation (CSS)
from behavior (JavaScript)

This makes each one easier to understand if they are
kept separate instead of putting everything in one large,
confusing HTML (or JavaScript) file

Site-wide consistency: control how all content looks using only 1 (or a few) CSS file(s)

Apply different styles to same content in different *media*:

- » desktop web browser
- » mobile web browser
- » auditory
- » print
- » & more!

Adherence to standards



It's fun!



It's fun!

Integrating CSS

4 ways to connect your HTML with your CSS

1. Linking to external styles ★
2. Embedded styles
3. Inline styles
4. `@import`

Linking

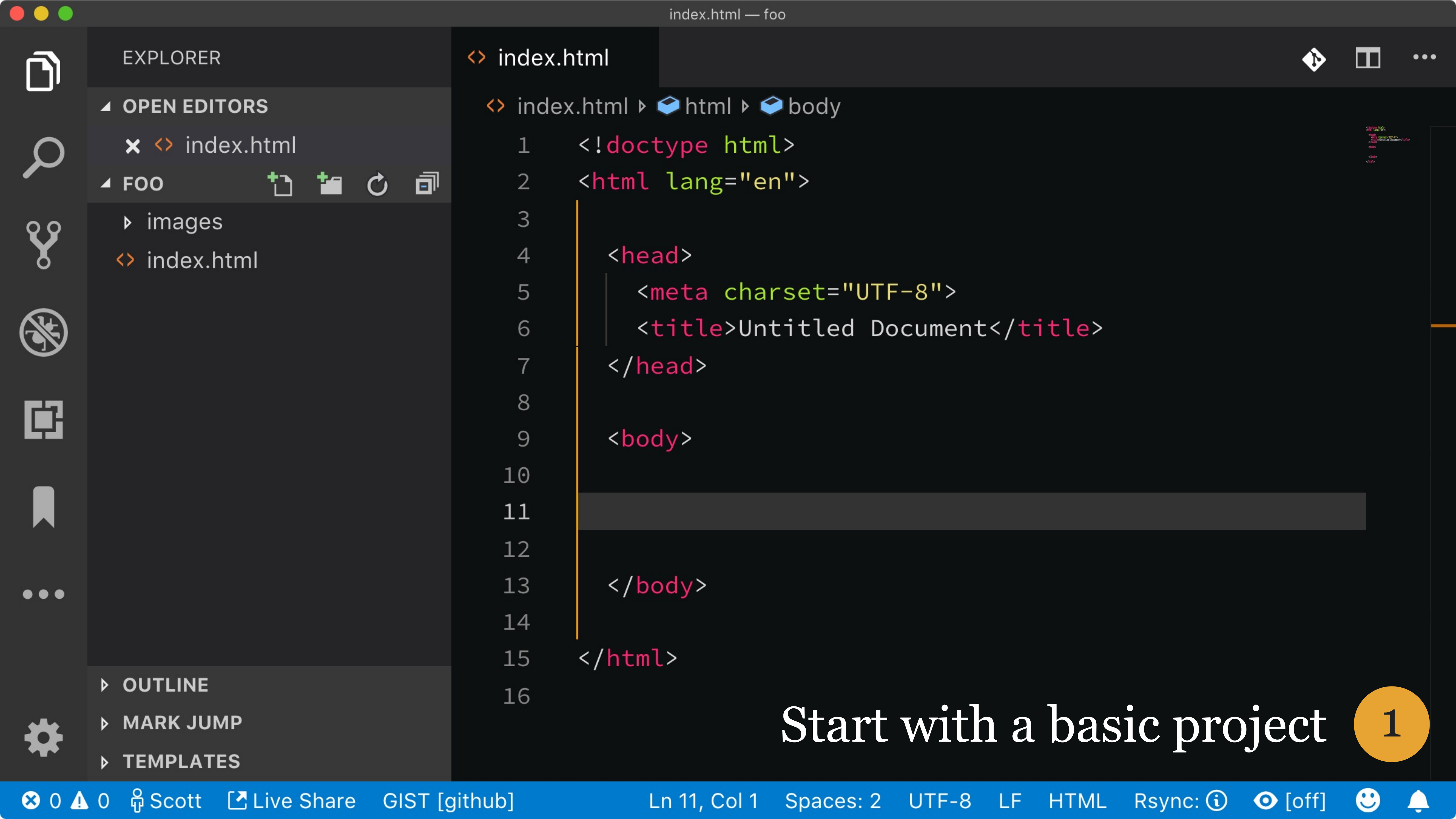
Linked styles use the `<link>` tag to connect the HTML to a CSS file

HTML 4.01 ← Obsolete, so do not use!

```
<head>  
  <link rel="stylesheet" type="text/css"  
    href="/css/main.css">  
</head>
```

HTML 5

```
<head>  
  <link rel="stylesheet" href="/css/main.css">  
</head>
```



<> index.html

<> index.html › html › body

```
1  <!doctype html>
2  <html lang="en">
3
4      <head>
5          <meta charset="UTF-8">
6          <title>Untitled Document</title>
7      </head>
8
9      <body>
10
11
12
13      </body>
14
15  </html>
16
```

Start with a basic project

1



EXPLORER

▲ OPEN EDITORS

✕ <> index.html

▲ FOO



▲ CSS

▸ images

<> index.html



▸ OUTLINE

▸ MARK JUMP

▸ TEMPLATES

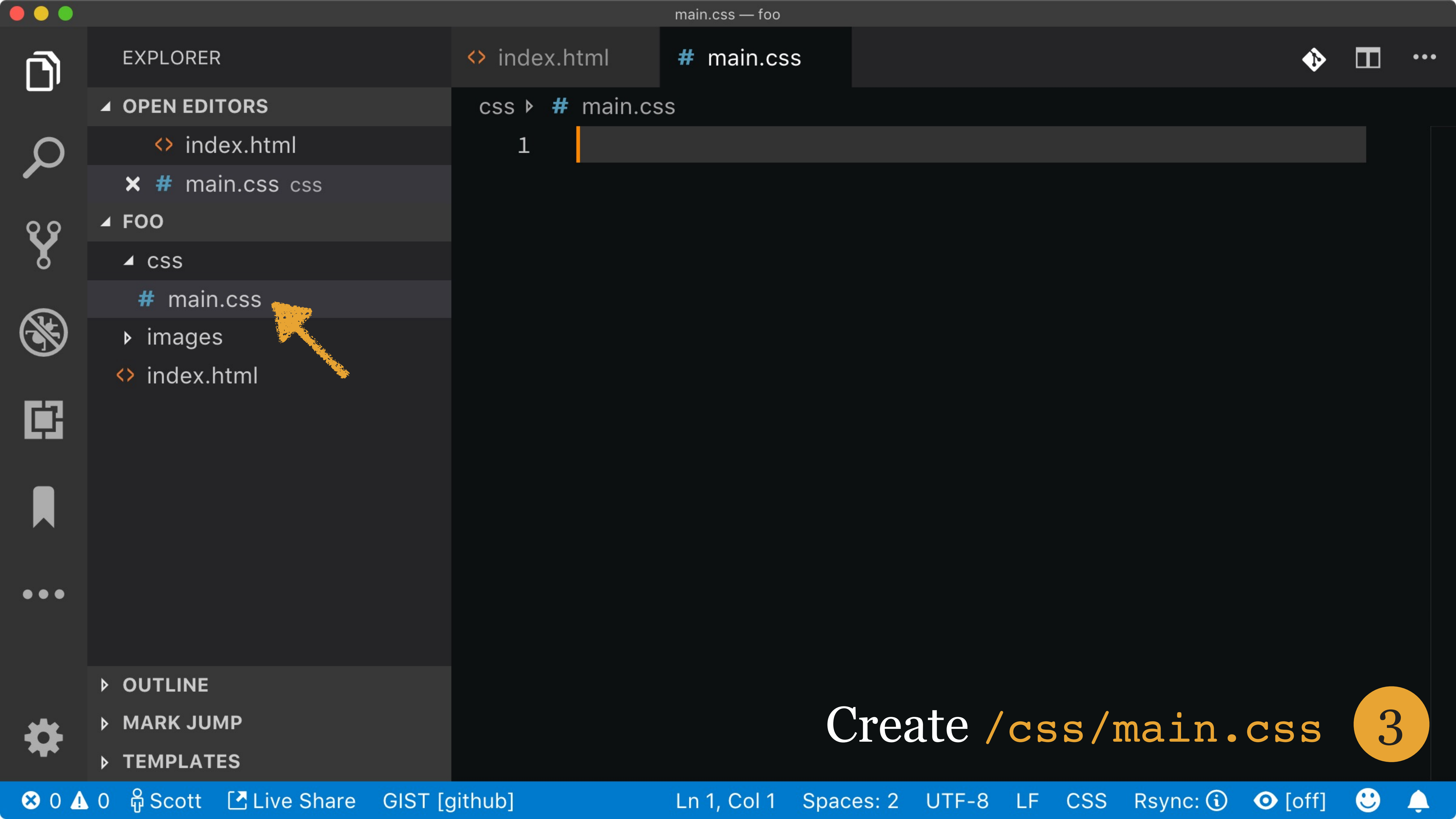
<> index.html

<> index.html ▸ html ▸ body

```
1  <!doctype html>
2  <html lang="en">
3
4      <head>
5          <meta charset="UTF-8">
6          <title>Untitled Document</title>
7      </head>
8
9      <body>
10
11
12
13      </body>
14
15  </html>
16
```

Create a **css** folder

2



Create `/css/main.css`

3



EXPLORER

▲ OPEN EDITORS 1 UNSAVED

● <> index.html

main.css css

▲ FOO

▲ css

main.css

▸ images

<> index.html



▸ OUTLINE

▸ MARK JUMP

▸ TEMPLATES

<> index.html ● # main.css

<> index.html ▸ html ▸ head

```
1  <!doctype html>
2  <html lang="en">
3
4      <head>
5          <meta charset="UTF-8">
6          <title>Untitled Document</title>
7          <link rel="stylesheet" href="/css/main.css">
8      </head>
9
10     <body>
11
12
13
14     </body>
15
16 </html>
17
```

Link to `/css/main.css`

4



What should you name your CSS file?

It doesn't matter

`main.css` ← What we, & many others, use

`typography.css`

`client.css`

`search.css`

`navigation.css`

Where should you place your CSS file?

In your website's root directory, always create these directories:

`css` (or `styles`)

`images` (or `media`)

`js` (or `scripts`)

`fonts` (maybe)

A CSS file is made up of style rules & comments

```
/* Comment */
```

```
blockquote, p, td {  
    font-family: Verdana, sans-serif;  
    font-size: 1em;  
}
```

Style rule 1

```
#footer {  
    font-size: .9em;  
}
```

Style rule 2

```
.emphasis {  
    font-weight: bold;  
}
```

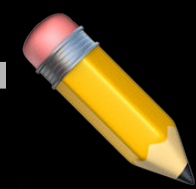
Style rule 3

You can link to more than one style sheet, but you should try to keep those links to a minimum

Every link is another server connection & another download, which slows down page load time

If you have more than one webpage, you really ought to use an external style sheet

You can now change the appearance & layout of an entire site by changing only one document!



SIDE NOTE

You might see advice telling you to add this as the 1st line of your style sheet so the browser knows that `main.css` is encoded as UTF-8:

```
@charset "utf-8";
```

This is no longer correct & can be ignored

Embedded

Embedded styles insert CSS inside `<style> ... </style>`

Most often in the `<head>`, but can be found anywhere in the `<body>`

HTML

```
1 <head>
2   <meta charset="UTF-8">
3   <title>Embedded Styles
    Example</title>
4   <style>
5     html {
6       font-size: 24px;
7       font-family: "Iowan Old
    Style", Georgia, serif;
8     }
9     h1, h2 {
10      font-family: sans-serif;
11    }
12  </style>
13 </head>
14
15 <body>
16   <h1>Cthulhu</h1>
17   <p>Ph'nglui mglw'nafh Cthulhu
    R'lyeh wgah'nagl fhtagn!</p>
18   <h2>Nyarlathotep</h2>
19   <p>Ph'nglui mglw'nafh Cthulhu
    R'lyeh wgah'nagl fhtagn!</p>
20 </body>
```

CSS (SCSS)

JS

Cthulhu

Ph'nglui mglw'nafh Cthulhu R'lyeh wgah'nagl fhtagn!

Nyarlathotep

Ph'nglui mglw'nafh Cthulhu R'lyeh wgah'nagl fhtagn!

Embedded styles are great for one page ...

... but they rapidly become difficult to manage on multiple pages

Avoid using embedded styles except in very specific circumstances

Inline

Inline styles use the `style` global attribute

Do not do this!

⚙ HTML



```
1 <p style="color: red; font-weight: bold;  
  font-size: 180%;">  
2   Do not disturb Cthulhu!  
3 </p>  
4  
5 <p>  
6   Enjoy your trip to R'yleh.  
7 </p>  
8  
9 <p>  
10  We hope you emerge relatively sane.  
11 </p>
```

⚙ CSS



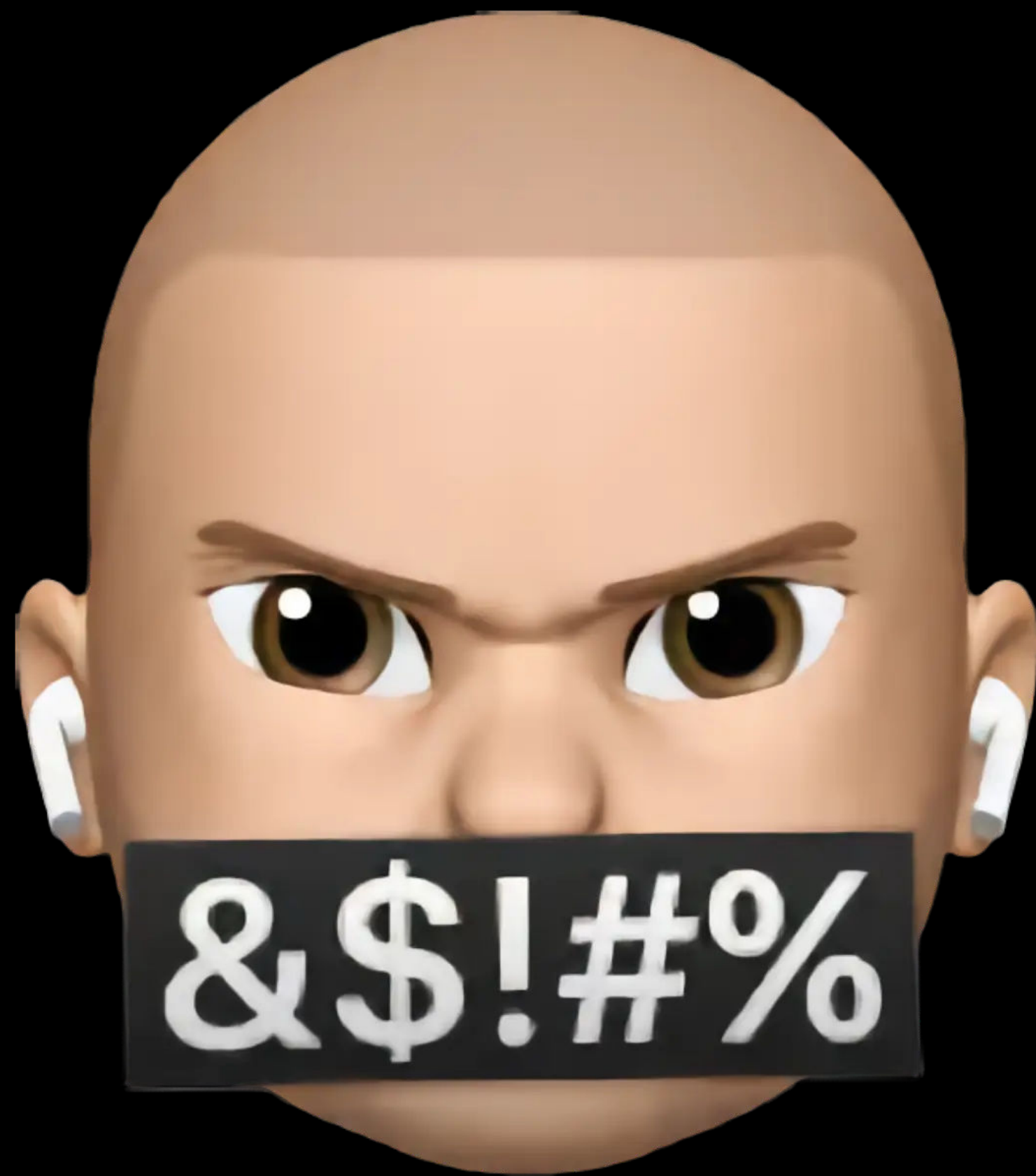
⚙ JS



Do not disturb Cthulhu!

Enjoy your trip to R'yleh.

We hope you emerge relatively sane.



Again, do not do this!

Quick & easy to create, but
difficult & time-consuming
to manage

Must repeat over and over

Can't change the style for
different media, e.g., print
& responsive

@import

`@import` allows you to include external stylesheets in your CSS; in other words, it allows you to link to another CSS file from within a CSS file (yes, this is a little weird)

`@import` must always come 1st, ahead of any other CSS

import.css

import.css ▸ h2

```
1  html {  
2    font-size: 24px;  
3    font-family: "Iowan Old Style", Georgia, serif;  
4  }  
5  
6  h2,  
7  h3,  
8  h4 {  
9    border-bottom: 1px solid black;  
10   font-family: Verdana, Geneva, Tahoma, sans-serif;  
11 }  
12
```

`import.css` is the style sheet that will be imported

⚙ HTML ▾

1 <h2>The Shadow out of Time
<small>H.P.
Lovecraft</small></h2>

2 <p>

3 After twenty-two years of
nightmare and terror, saved only
by a desperate conviction of the
mythical source of certain
impressions, I am unwilling to
vouch for the truth of that which
I think I found in Western
Australia on the night of July
17–18, 1935. There is reason to
hope that my experience was
wholly or partly an
hallucination—for which, indeed,
abundant causes existed. And yet,
its realism was so hideous that I
sometimes find hope impossible.
If the thing did happen, then man

⚙ CSS ▾

1 @import url("https://s3-us-
west-2.amazonaws.com/s.cdn.io
/122116/import.css");

2

⚙ JS ▾

The Shadow out of Time H.P. Lovecraft

After twenty-two years of nightmare and terror, saved only by a desperate conviction of the mythical source of certain impressions, I am unwilling to vouch for the truth of that which I think I found in Western Australia on the night of July 17–18, 1935. There is reason to hope that my experience was wholly or partly an hallucination—for which, indeed, abundant causes existed. And yet, its realism was so hideous that I sometimes find hope impossible. If the thing did happen, then man must be prepared to accept notions of the cosmos, and of his own place in the seething vortex of time, whose merest mention is paralysing. He must, too, be placed on guard against a specific lurking peril which, though it will never engulf the whole race, may impose monstrous and unguessable horrors upon certain venturesome members of it. It is for this latter reason that I urge, with all the force of my being, a final abandonment of all attempts at unearthing those fragments of unknown, primordial masonry which my expedition set out to investigate.

import.css included via @import

HTML

1

<h2>The Shadow out of Time

<small>H.P.

Lovecraft</small></h2>

2

<p>

3

After twenty-two years of

nightmare and terror, saved only

by a desperate conviction of the

CSS

1

@import url("https://s3-us-

west-2.amazonaws.com/s.cdn.io

/122116/import.css");

2

3

h2, h3, h4 {

4

font-family: -apple-system,

BlinkMacSystemFont, "Segoe UI",

Roboto, Oxygen, Ubuntu,

Cantarell, "Fira Sans", "Droid

Sans", "Helvetica Neue", Arial,

sans-serif;

5

}

6

7

small {

8

display: block;

9

margin-top: .25em;

10

}

JS

The Shadow out of Time

H.P. Lovecraft

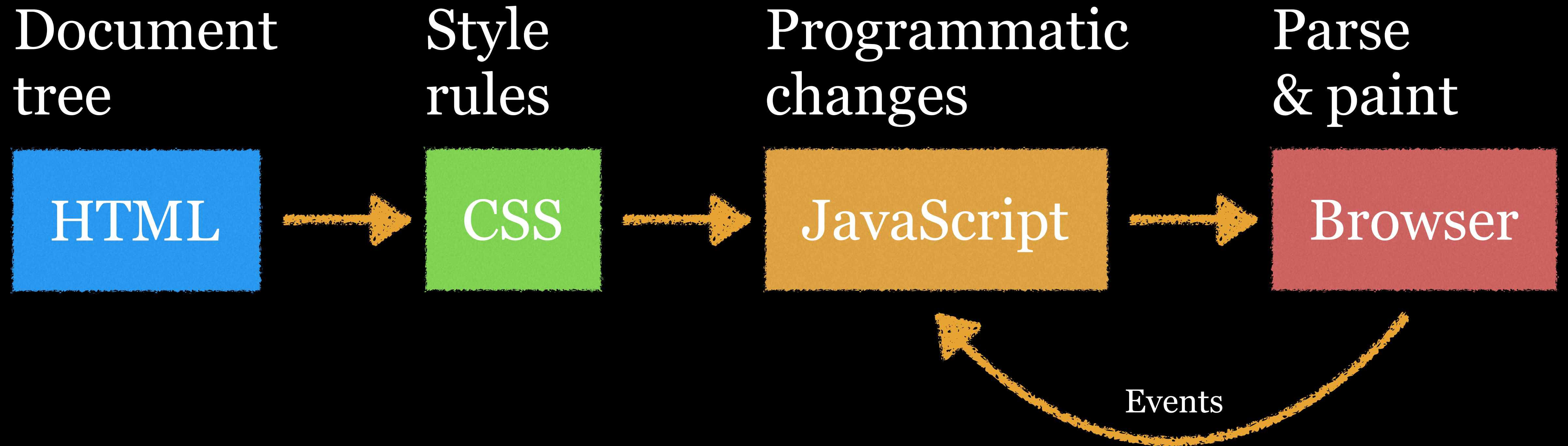
After twenty-two years of nightmare and terror, saved only by a desperate conviction of the mythical source of certain impressions, I am unwilling to vouch for the truth of that which I think I found in Western Australia on the night of July 17–18, 1935. There is reason to hope that my experience was wholly or partly an hallucination—for which, indeed, abundant causes existed. And yet, its realism was so hideous that I sometimes find hope impossible. If the thing did happen, then man must be prepared to accept notions of the cosmos, and of his own place in the seething vortex of time, whose merest mention is paralysing. He must, too, be placed on guard against a specific lurking peril which, though it will never engulf the whole race, may impose monstrous and unguessable horrors upon certain venturesome members of it. It is for this latter reason that I urge, with all the force of my being, a final abandonment of all attempts at unearthing those fragments of unknown, primordial masonry which my expedition set out to investigate.

Put `@import` 1st so it can be overridden

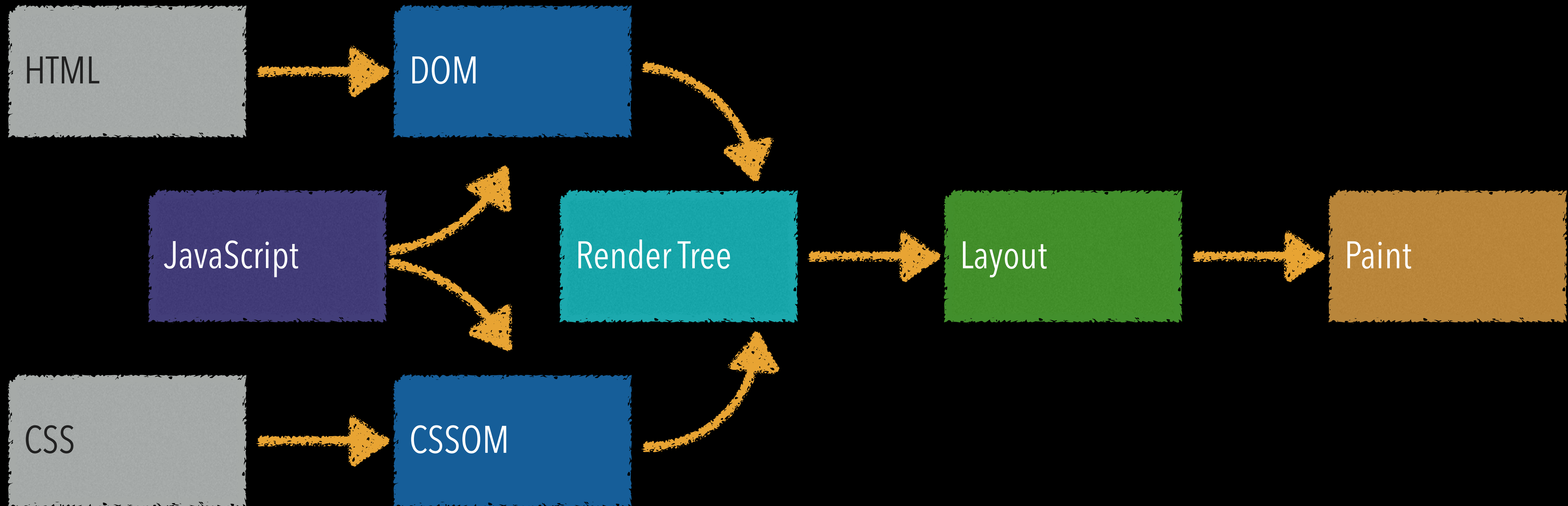
Don't use `@import` unless you absolutely need to

- » Slows down your page loads (the browser downloads your CSS, *then* downloads the imported CSS)
- » Adds complexity

The Browser Processing Pipeline



The browser processing pipeline (simplified)



The browser processing pipeline (more detailed)

To understand CSS, you have to understand

- » the DOM (*Document Object Model*)
- » the CSSOM (*CSS Object Model*)
- » the browser processing pipeline

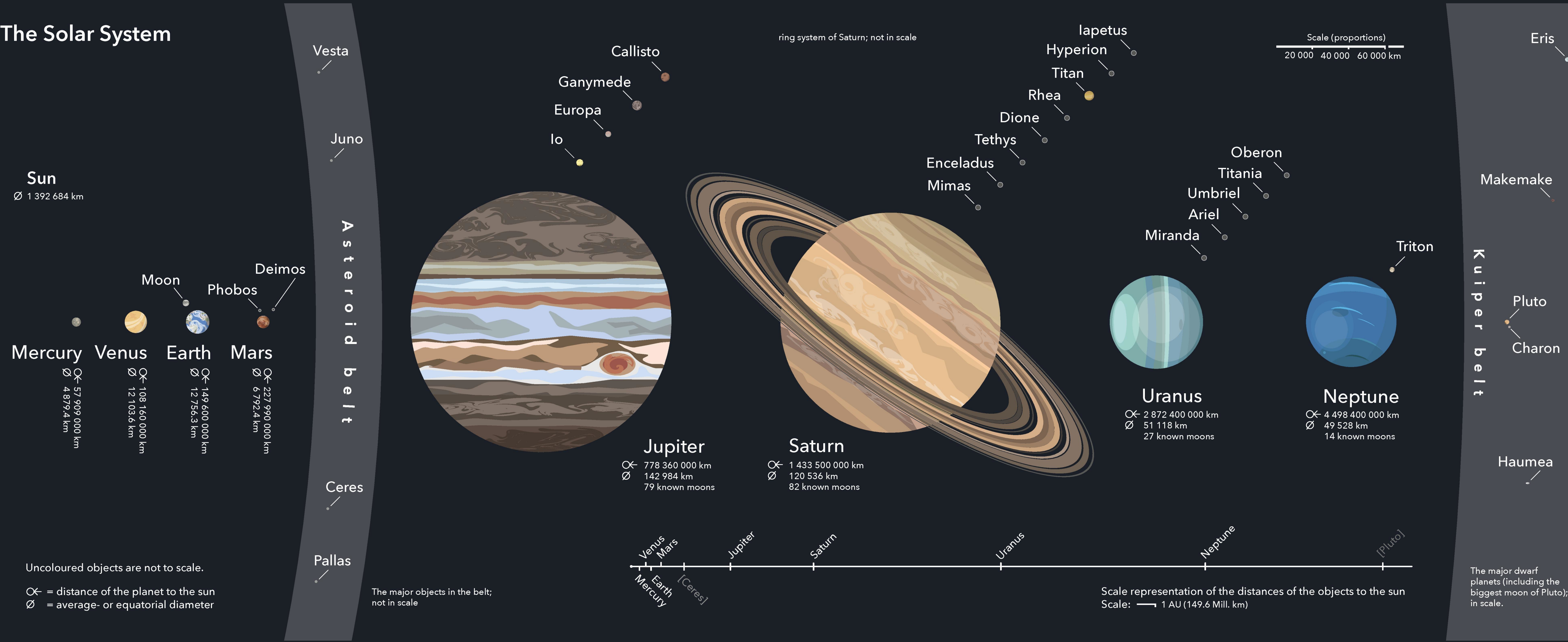
The DOM

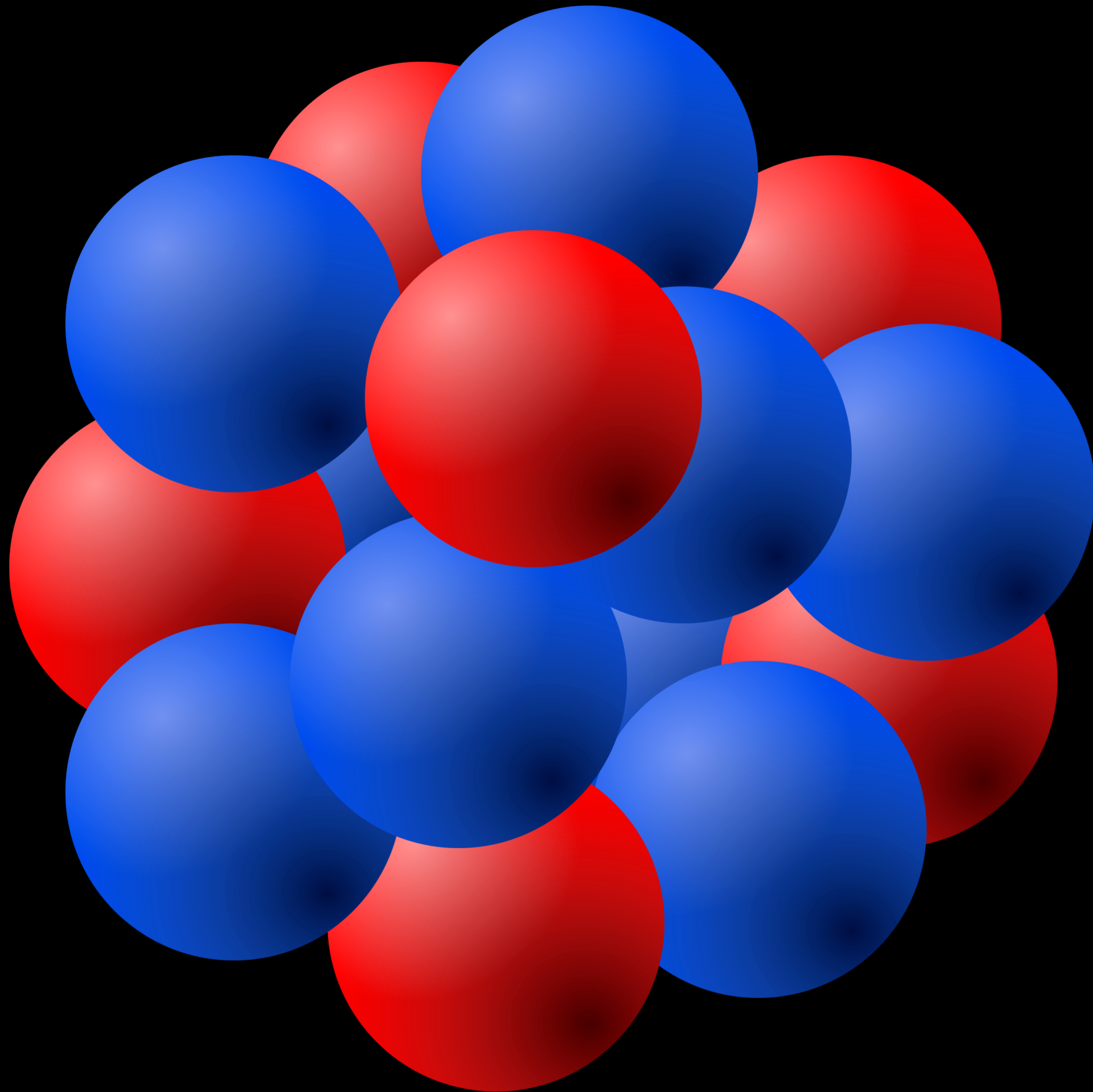
The DOM is a conceptual model

“A *conceptual model* is a representation of a system, made of the composition of concepts which are used to help people know, understand, or simulate a subject the model represents.” —Wikipedia

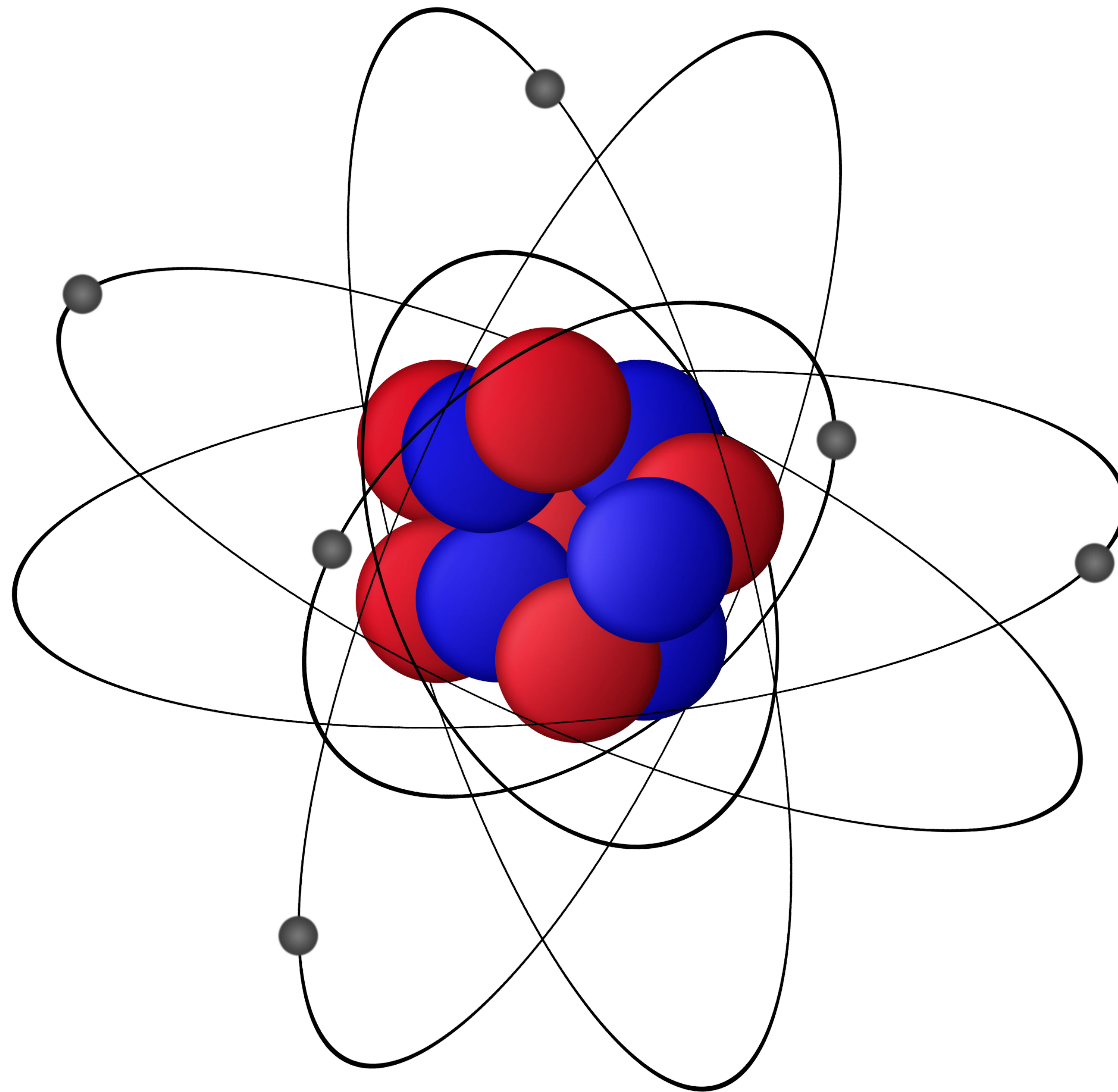
We often illustrate conceptual models of both tangible & intangible things so we can better understand them

The Solar System

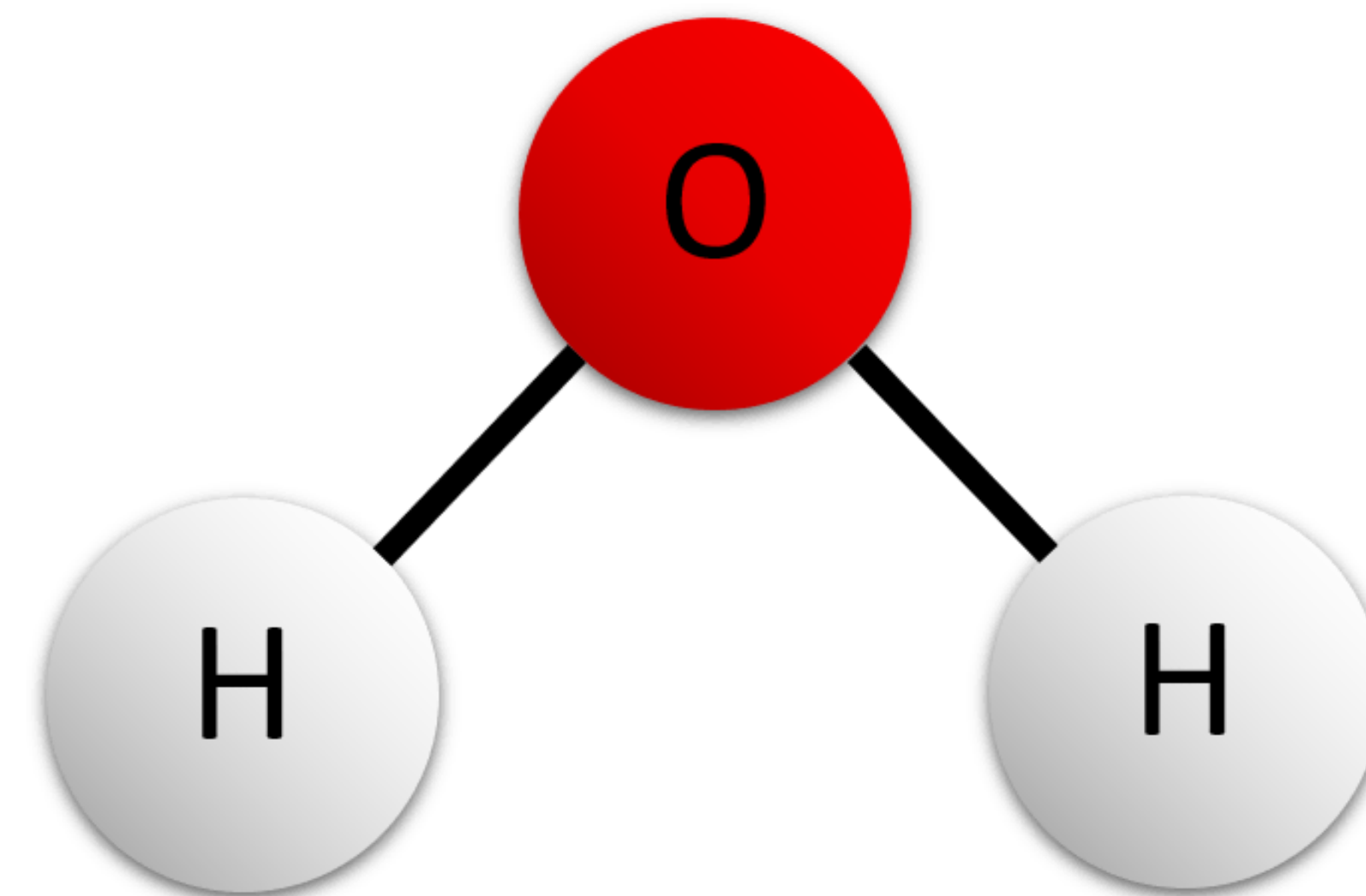
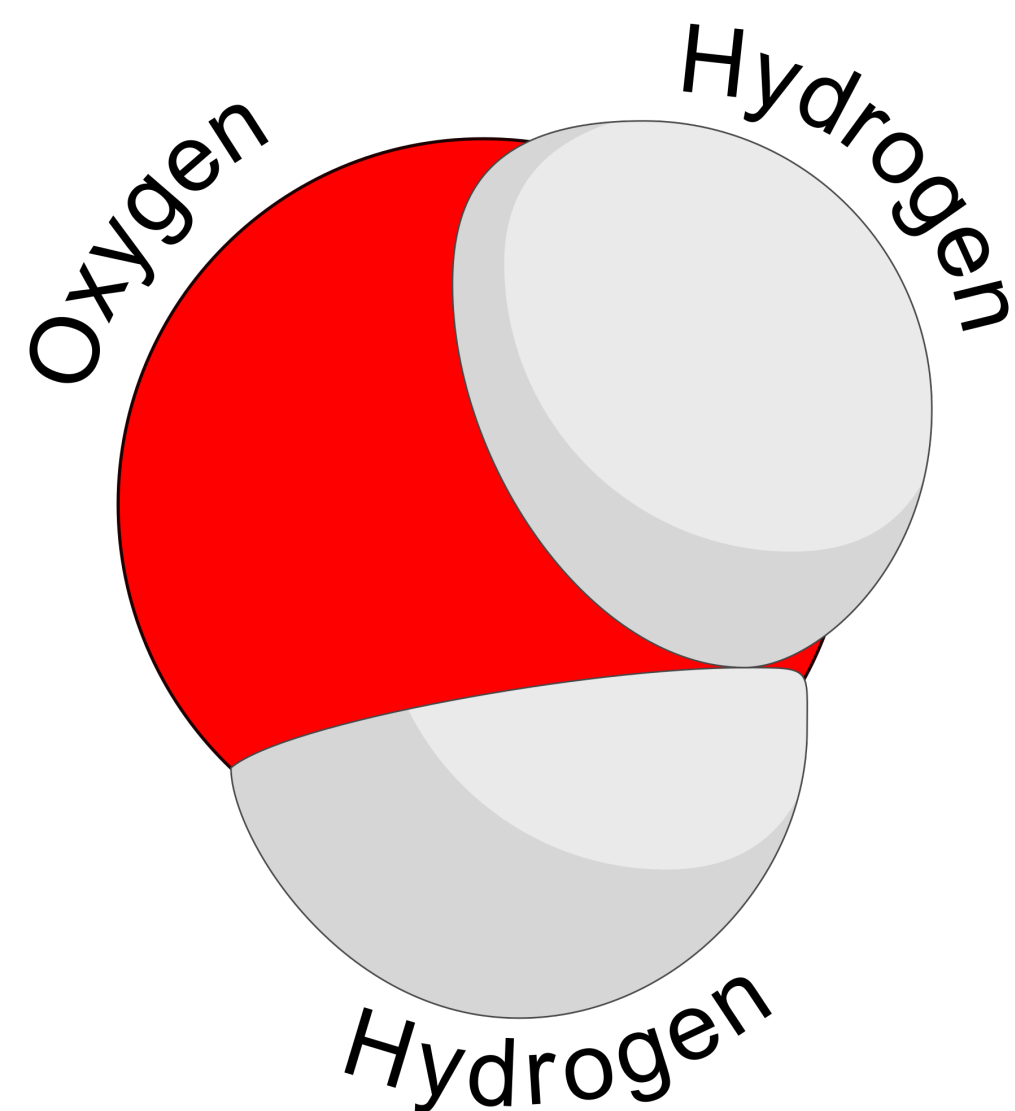




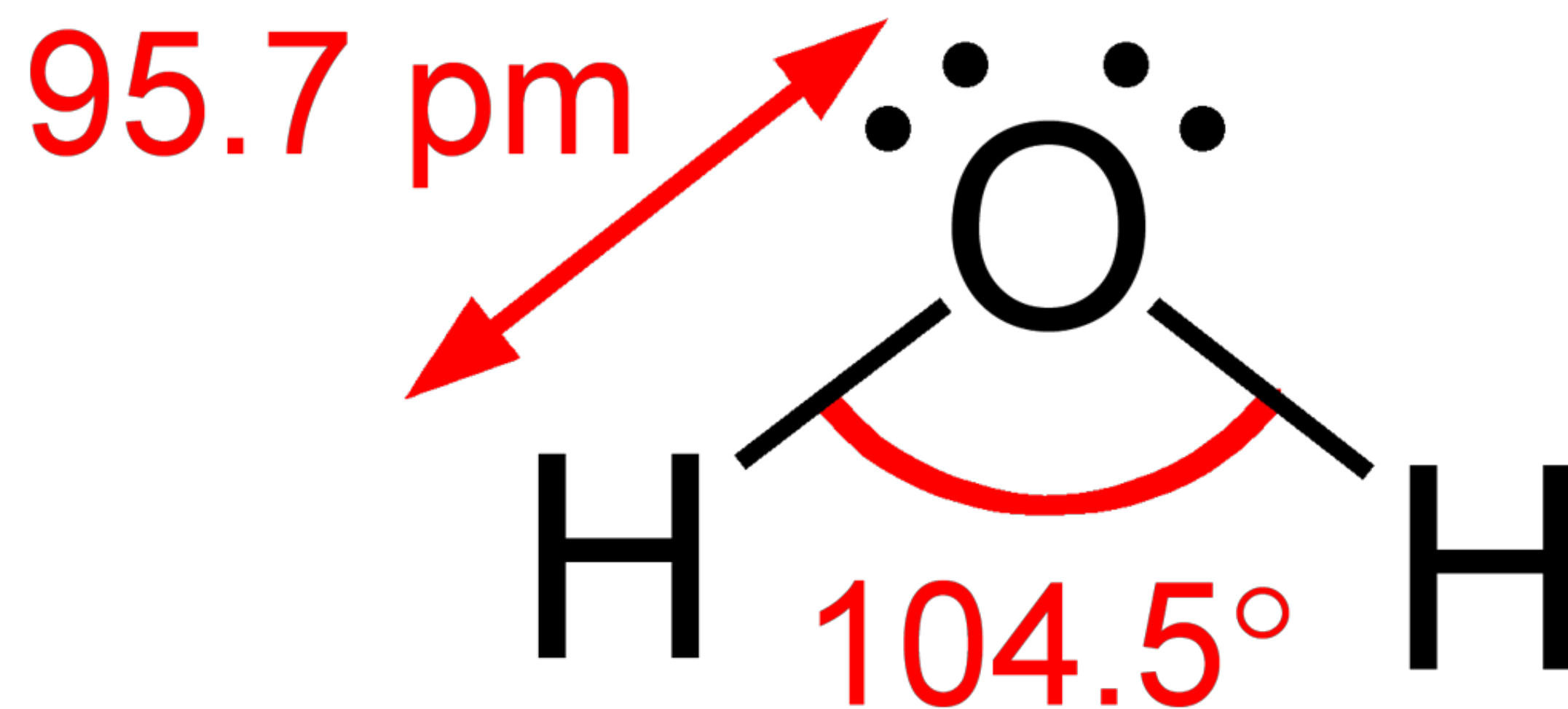
Atomic nucleus
composed of
neutrons (blue)
& protons (red)



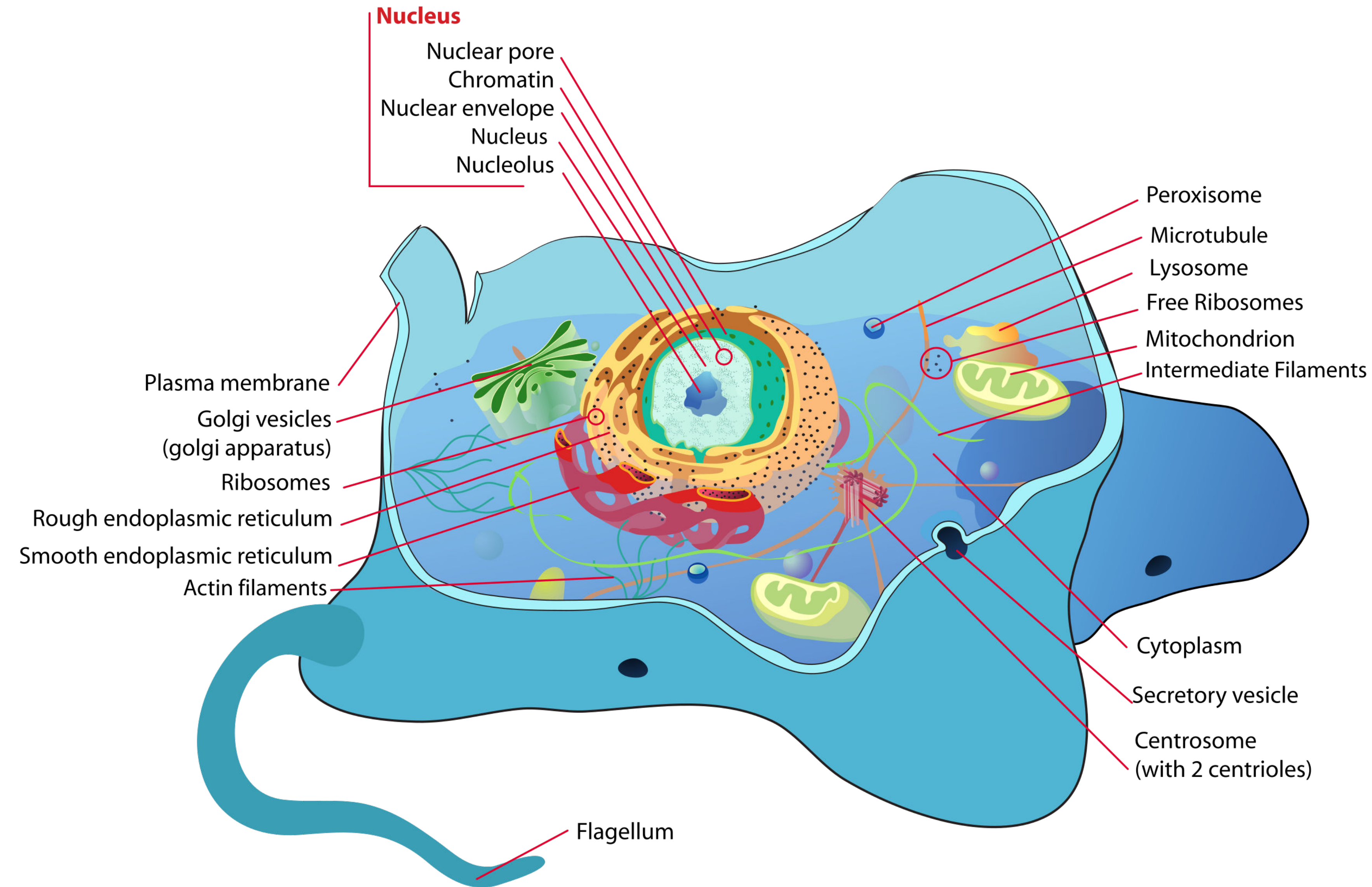
Carbon atom,
with electron
cloud around
nucleus



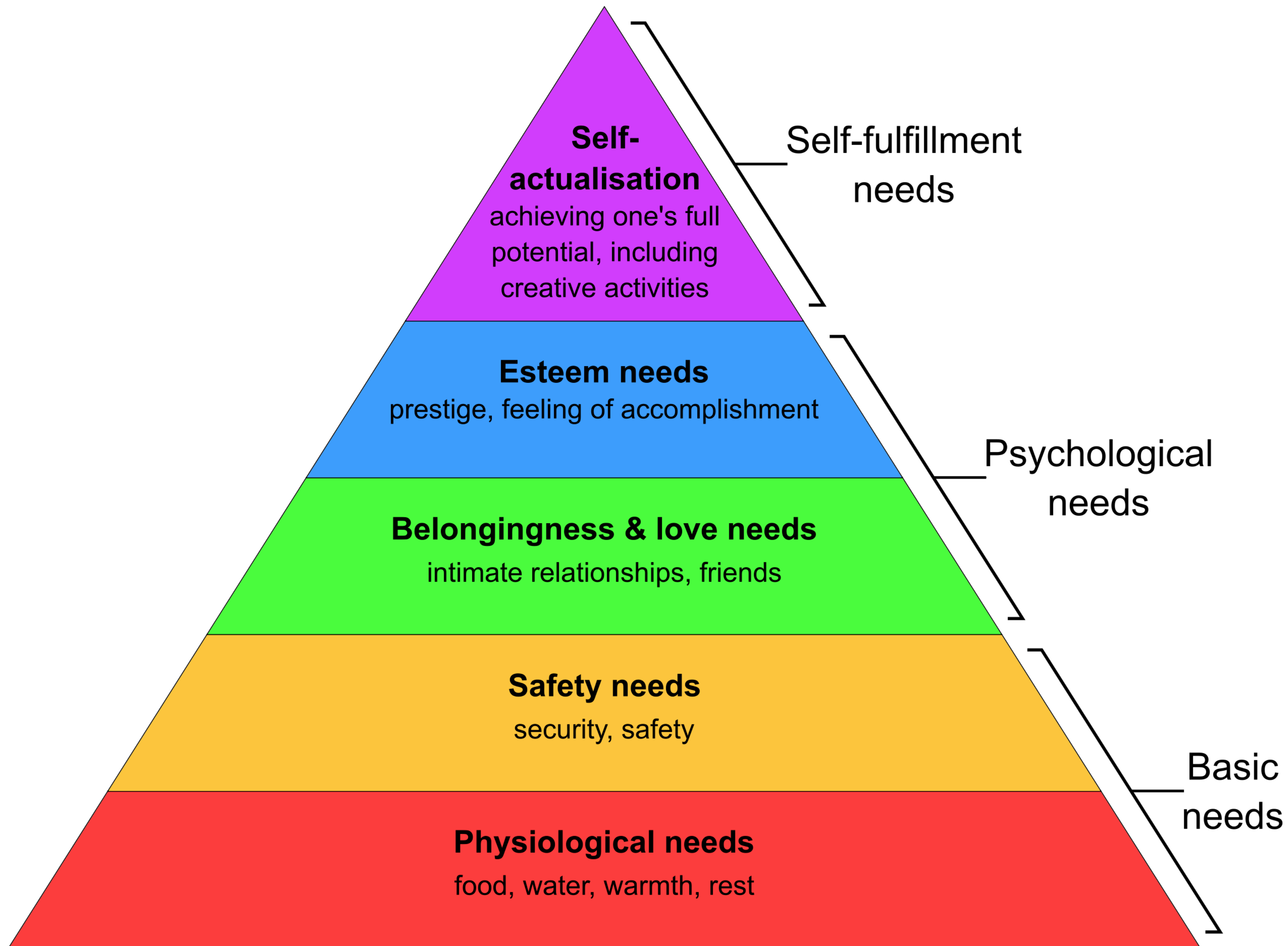
Different models for a water molecule (an electrically neutral group of 2 or more atoms held together by chemical bonds)

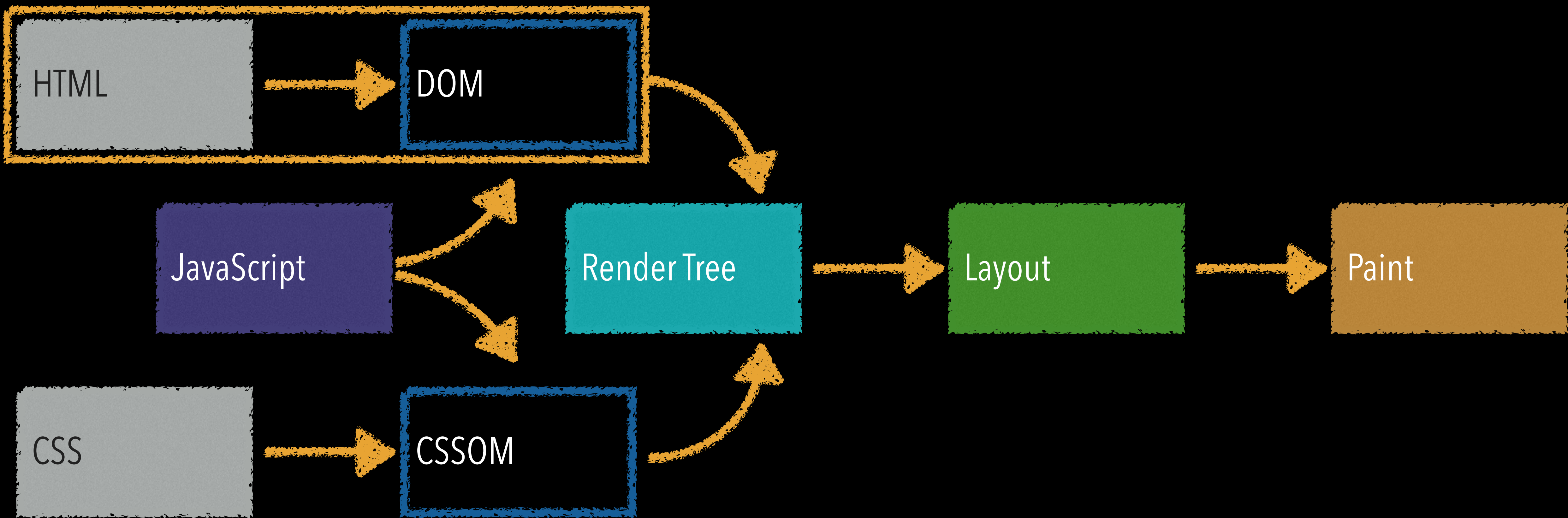


Animal cell,
made up of
molecules,
which contain
millions or even
trillions of
atoms



Maslow's Hierarchy of Needs





Before a webpage appears in a viewport, the rendering engine downloads the HTML & parses it to figure out how to display the webpage on screen

During this process, the rendering engine creates the *DOM tree*

“A Web page is a document. This document can be either displayed in the browser window or as the HTML source. But it is the same document in both cases.

The Document Object Model (DOM) represents that same document so [the document structure, style, and content] can be manipulated. *The DOM is an object-oriented representation of the web page, which can be modified with ... JavaScript.* —MDN web docs

So what's the DOM?

A JavaScript developer will say, “It's a JavaScript interface so that I can manipulate the page”

A CSS developer will say, “It's a set of boxes to style”

An HTML developer will say, “It's hierarchy & meaning”

They are all correct!

When is the DOM different than your HTML?

If you have mistakes in your HTML, the rendering engine “fixes” them when it generates the DOM

Article

Talk

Read

Edit

View history



The Thing (1982 film)

From Wikipedia, the free encyclopedia



This article **needs additional citations for verification**. Please help [improve this article](#) by [adding citations to reliable sources](#). Unsourced material may be challenged and removed.
(May 2016) *([Learn how and when to remove this template message](#))*

The Thing (also known as ***John Carpenter's The Thing***) is a 1982 American science-fiction horror film directed by [John Carpenter](#), written by [Bill Lancaster](#), and starring [Kurt Russell](#). The film's title refers to its primary antagonist: a [parasitic extraterrestrial lifeform](#) that [assimilates](#) other organisms and in turn [imitates](#) them. The Thing infiltrates an Antarctic research station, taking the appearance of the researchers that it absorbs, and [paranoia](#) develops within the group.

The film is based on [John W. Campbell, Jr.](#)'s novella *[Who Goes There?](#)*, which was more loosely adapted by [Howard Hawks](#) and [Christian Nyby](#) as the 1951 film *[The Thing from Another World](#)*.^[3] Carpenter considers *The Thing* to be the first part of his *[Apocalypse Trilogy](#)*,^[4] followed by *[Prince of Darkness](#)* and *[In the Mouth of Madness](#)*. Although the films are narratively unrelated, each features a potentially apocalyptic scenario; should "The Thing" ever reach civilization, it would be only a matter of time before it consumes all life on Earth. Carpenter acknowledged that the work of [H.P. Lovecraft](#) also inspired the film.^{[5][6]}

On June 25, 1982, *The Thing* opened in 840 theaters, and it was the #8 movie of the

(Excellent horror flick!)

The Thing



Source code for that webpage

```
1 <!DOCTYPE html>
2 <html class="client-nojs" lang="en" dir="ltr">
3 <head>
4 <meta charset="UTF-8"/>
5 <title>The Thing (1982 film) - Wikipedia</title>
6 <script>document.documentElement.className = document.documentElement.className.replace( /(^\s)client-nojs(\s|$)/, "$1client-
js$2" );</script>
7 <script>(window.RLQ=window.RLQ||[]).push(function()
{mw.config.set({"wgCanonicalNamespace":"","wgCanonicalSpecialPageName":false,"wgNamespaceNumber":0,"wgPageName":"The_Thing_(19
82_film)","wgTitle":"The Thing (1982
film)","wgCurRevisionId":806123497,"wgRevisionId":806123497,"wgArticleId":30809,"wgIsArticle":true,"wgIsRedirect":false,"wgAct
ion":"view","wgUserName":null,"wgUserGroups":["*"],"wgCategories":["CS1 maint: Unfit url","Webarchive template wayback
links","Pages with citations lacking titles","Pages with citations having bare URLs","Articles needing additional references
from May 2016","All articles needing additional references","Articles needing additional references from February
2017","Articles needing additional references from April 2017","All articles with unsourced statements","Articles with
unsourced statements from August 2016","Articles with hAudio microformats","Wikipedia articles with VIAF
identifiers","Wikipedia articles with GND identifiers","Wikipedia articles with BNF identifiers","1982 films","English-
language films","1982 horror films","1980s science fiction films","1980s science fiction horror films","Films based on science
fiction short stories","American films","American science fiction horror films","Norwegian-language films","Films directed by
John Carpenter","Action horror films","Films based on science fiction novels","Films set in Antarctica","Films set in
1982","Films shot in Alaska","Films shot in British Columbia","Monster movies","Films using stop-motion animation","Universal
Pictures films","Films about shapeshifting","Fictional parasites and parasitoids","Film scores by Ennio Morricone","Splatter
films","Body horror films","Films about extraterrestrial life","Films adapted into novels","Films adapted into comics","Films
adapted into video
games"],"wgBreakFrames":false,"wgPageContentLanguage":"en","wgPageContentModel":"wikitext","wgSeparatorTransformTable":
["",""],"wgDigitTransformTable":["",""],"wgDefaultDateFormat":"dmy","wgMonthNames":
["","January","February","March","April","May","June","July","August","September","October","November","December"],"wgMonthNam
esShort":
["","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"],"wgRelevantPageName":"The_Thing_(1982_film)","wgR
elevantArticleId":30809,"wgRequestId":"WekWTQpAICgAAC6@iyoAAABT","wgIsProbablyEditable":true,"wgRelevantPageIsProbablyEditable
":true,"wgRestrictionEdit":[],"wgRestrictionMove":[],"wgFlaggedRevsParams":{"tags":
{}},"wgStableRevisionId":null,"wgWikiEditorEnabledModules":
{"toolbar":true,"preview":false,"publish":false},"wgBetaFeaturesFeatures":
```

```
<!DOCTYPE html>
```

```
▼ <html class="client-js ve-not-available" lang="en" dir="ltr">
```

```
▶ <head>...</head>
```

```
▶ <body class="mediawiki ltr sitedir-ltr mw-hide-empty-elt ns-0 ns-subject page-The_Thing_1982_film rootpage-The_Thing_1982_film vector-experimental-print-styles vector-nav-directionality skin-vector action-view">...</body>
```

```
</html> = $0
```

DOM for that same page, as rendered by browser Inspector

Source code

No <tbody>

```
41 <div id="bodyContent" class="mw-body-content">
42 <div id="siteSub" class="noprint">From Wikipedia, the free encyclopedia</div>
43 <div id="contentSub"></div>
44 <div id="jump-to-nav" class="mw-jump">
45 Jump to: <a href="#mw-head">navigation</a>, <a href="#p-search">search</a>
46 </div>
47 <div id="mw-content-text" lang="en" dir="ltr" class="mw-content-ltr"><div class="mw-parser-output"><table
class="plainlinks metadata ambox ambox-content ambox-Refimprove" role="presentation">
48 <tr>
49 <td class="mbox-image">
50 <div style="width:52px"><a href="/wiki/File:Question_book-new.svg" class="image"></a></div>
51 </td>
52 <td class="mbox-text">
53 <div class="mbox-text-span">This article <b>needs additional citations for <a href="/wiki/Wikipedia:Verifiability"
title="Wikipedia:Verifiability">verification</a></b>. <span class="hide-when-compact">Please help <a class="external text"
href="//en.wikipedia.org/w/index.php?title=The_Thing_(1982_film)&action=edit">improve this article</a> by <a
href="/wiki/Help:Introduction_to_referencing_with_Wiki_Markup/1" title="Help:Introduction to referencing with Wiki
Markup/1">adding citations to reliable sources</a>. Unsourced material may be challenged and removed.</span> <small><i>(May
2016)</i></small> <small class="hide-when-compact"><i>(May 2016)</i></small> <small class="hide-when-compact"><i>(May 2016)</i></small> <a href="/wiki/Help:Maintenance_template_removal"
title="Help:Maintenance template removal">Learn how and when to remove this template message</a></i></small></div>
54 </td>
55 </tr>
56 </table>
57 <table class="infobox vevent" style="width:22em;font-size:90%;">
58 <tr>
59 <th colspan="2" class="summary" style="text-align:center;font-size:125%;font-weight:bold;font-size:110%;font-
style:italic;">The Thing</th>
60 </tr>
61 <tr>
62 <td colspan="2" style="text-align:center;"><a href="/wiki/File:The_Thing_(1982)_theatrical_poster.jpg" class="image"><img alt=""></a></td>
63 </tr>
```



```

▼ <body class="mediawiki ltr sitedir-ltr mw-hide-empty-elt ns-0 ns-subject page-The_Thing_1982_film rootpage-The_Thing_1982_film
vector-experimental-print-styles vector-nav-directionality skin-vector action-view">
  <div id="mw-page-base" class="noprint"></div>
  <div id="mw-head-base" class="noprint"></div>
  ▼ <div id="content" class="mw-body" role="main">
    <a id="top"></a>
    ▶ <div class="mw-body-content" id="siteNotice">...</div>
    <div class="mw-indicators mw-body-content"> </div>
    ▶ <h1 id="firstHeading" class="firstHeading" lang="en">...</h1>
    ▼ <div id="bodyContent" class="mw-body-content">
      <div id="siteSub" class="noprint">From Wikipedia, the free encyclopedia</div>
      <div id="contentSub"></div>
      ▶ <div id="jump-to-nav" class="mw-jump">...</div>
      ▼ <div id="mw-content-text" lang="en" dir="ltr" class="mw-content-ltr">
        ▼ <div class="mw-parser-output">
          ▼ <table class="plainlinks metadata ambox ambox-content ambox-Refimprove" role="presentation">
            ▼ <tbody>
              ▼ <tr>
                ▼ <td class="mbox-image">
                  ▼ <div style="width:52px">
                    ▼ <a href="/wiki/File:Question_book-new.svg" class="image">
                      
                    </a>
                  </div>
                </td>
                ▶ <td class="mbox-text">...</td>
              </tr>
            </tbody>
          </table>

```

<tbody> added

Rendered DOM

```
<table>
  <tr>
    <td>Foo</td>
    <td>Bar</td>
  </tr>
  <tr>
    <td>Baz</td>
    <td>Qux</td>
  </tr>
</table>
```

Source code

```
<table>
  <tbody>
    <tr>
      <td>Foo</td>
      <td>Bar</td>
    </tr>
    <tr>
      <td>Baz</td>
      <td>Qux</td>
    </tr>
  </tbody>
</table>
```

Rendered DOM

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Herbert West Landscaping</title>
  </head>
  <body>
    
    <h1>We'll fix your lawn... <i>for good</i>!</
h1>
    <p>We apply it all: fertilizers, mulch, &
blood</p>
  </body>
</html>
```

DOM tree as HTML elements, each one a node, with `<html>` as the document *root*

```

<html>
  <head>
    <meta charset="UTF-8">
    <title>Herbert West
Landscaping</title>
  </head>
  <body>
    
    <h1>We'll fix your
lawn... <i>for good</i>!</h1>
    <p>We apply it all:
fertilizers, mulch, &
blood</p>
  </body>
</html>

```

HTML

```

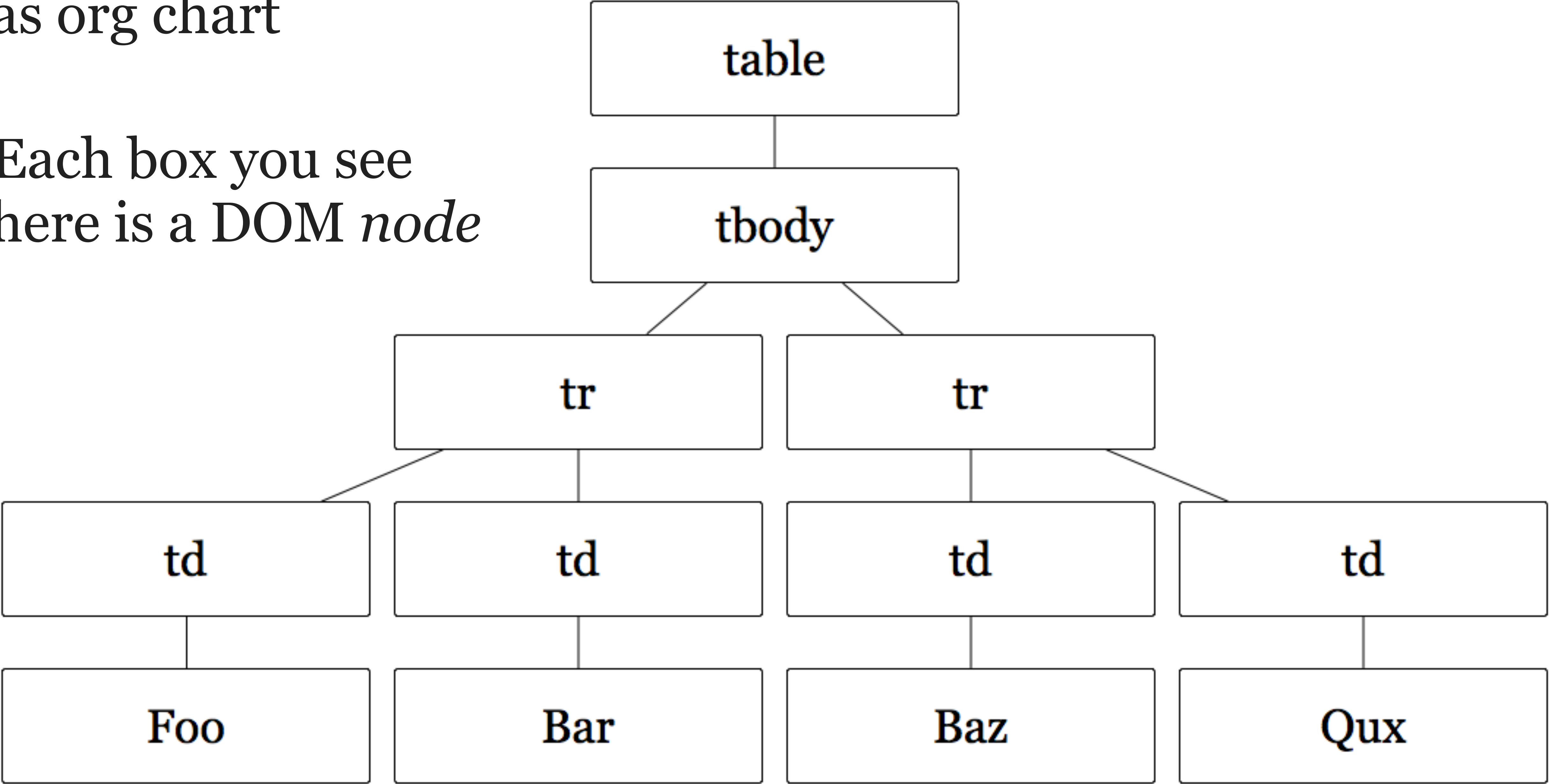
- Document
  - html
    - head
      - meta
      - title
        - "Herbert West
Landscaping"
    - body
      - img
      - h1
        - "We'll fix..."
          - i
            - "for good"
      - p
        - "We apply..."

```

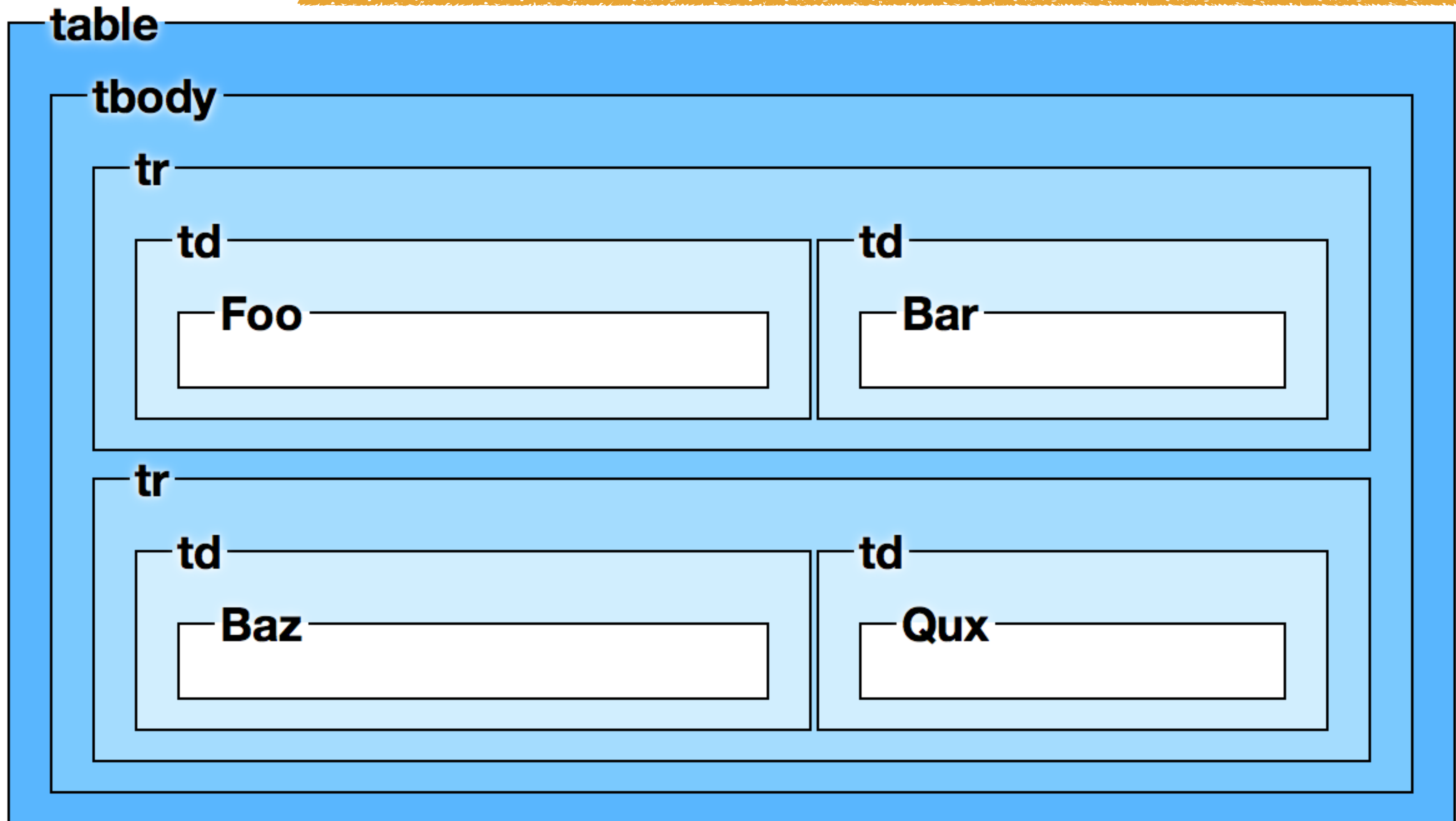
DOM tree

DOM tree
as org chart

Each box you see
here is a DOM *node*

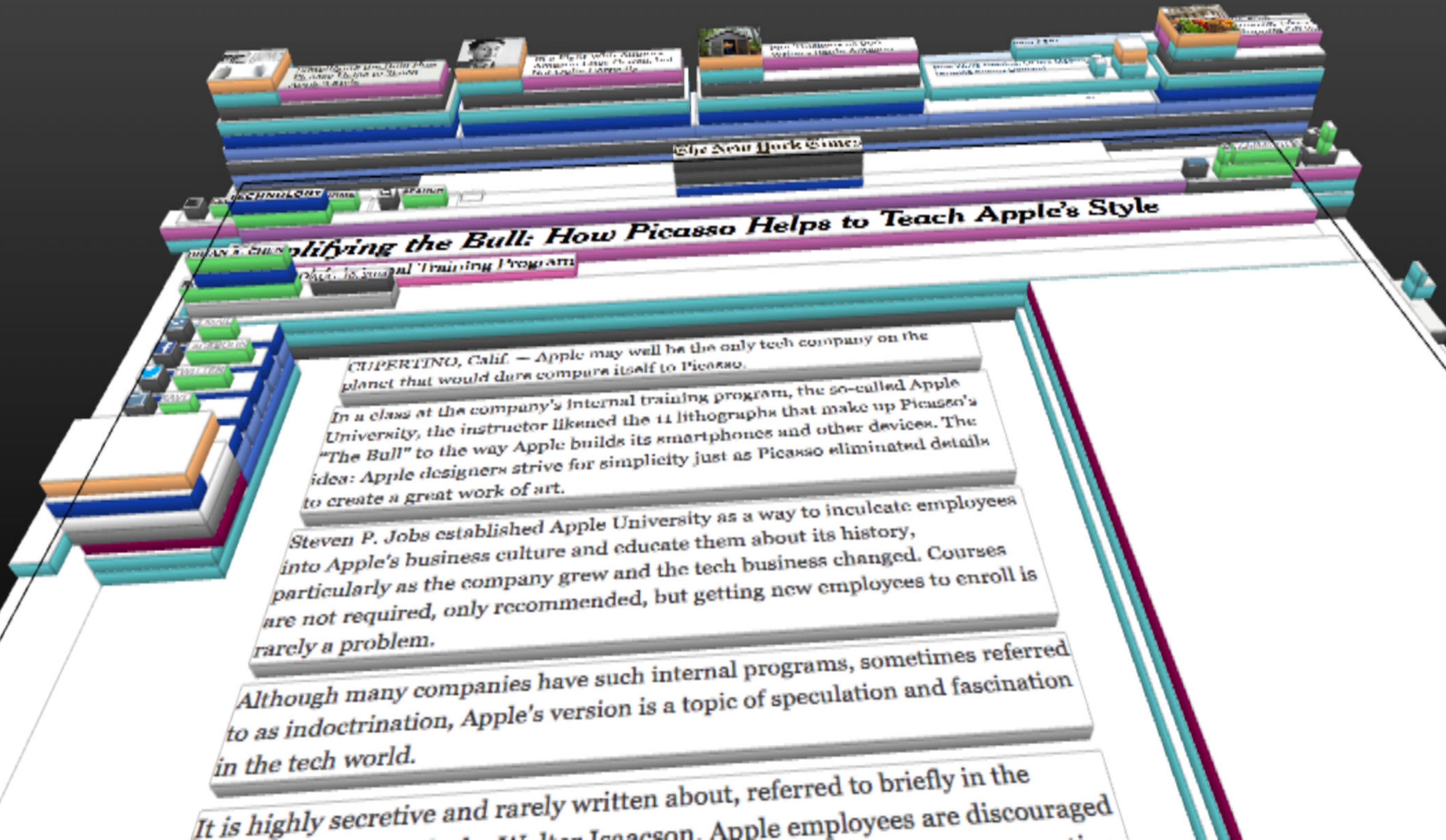


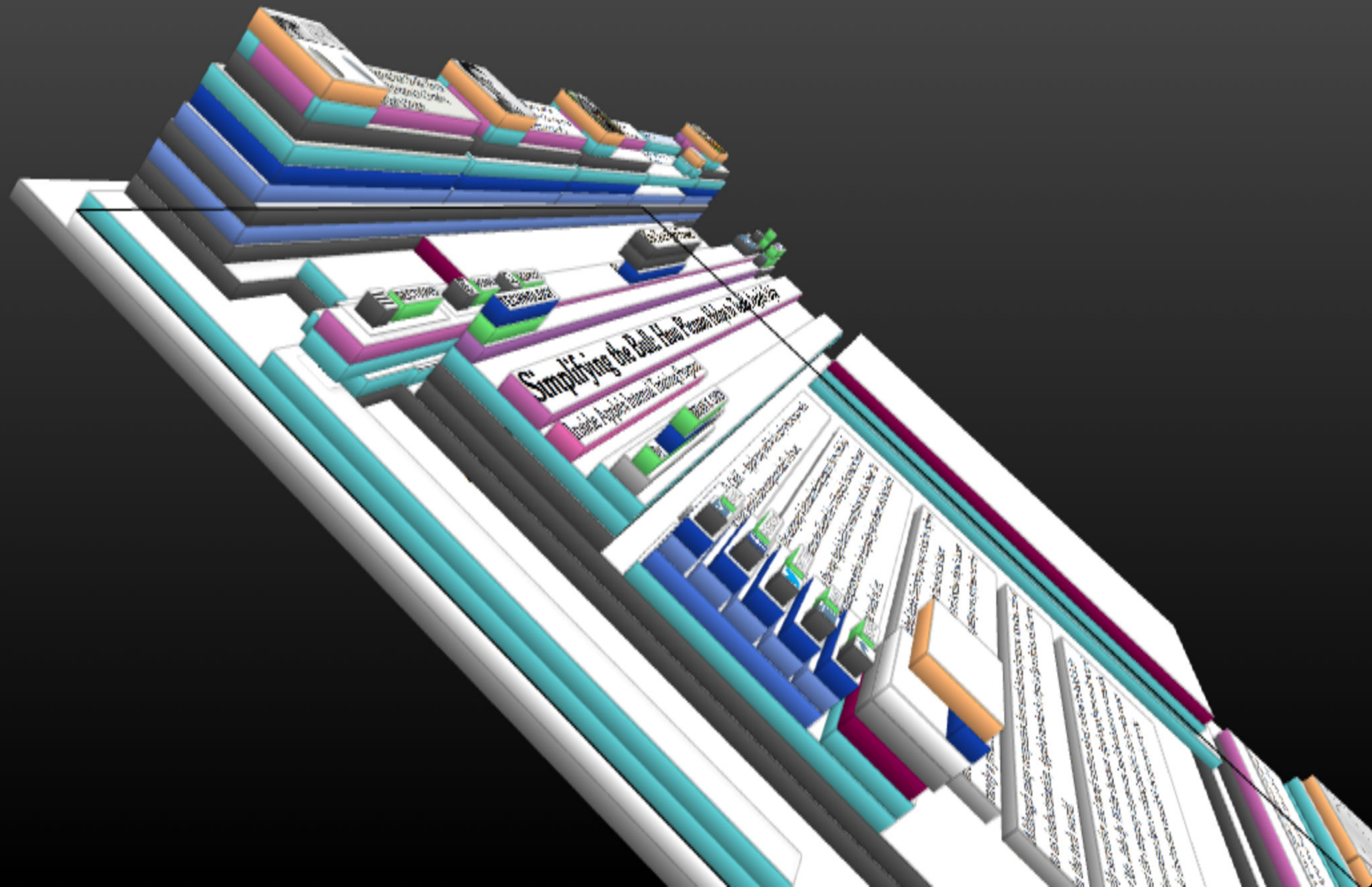
DOM tree as nested boxes, each one a node

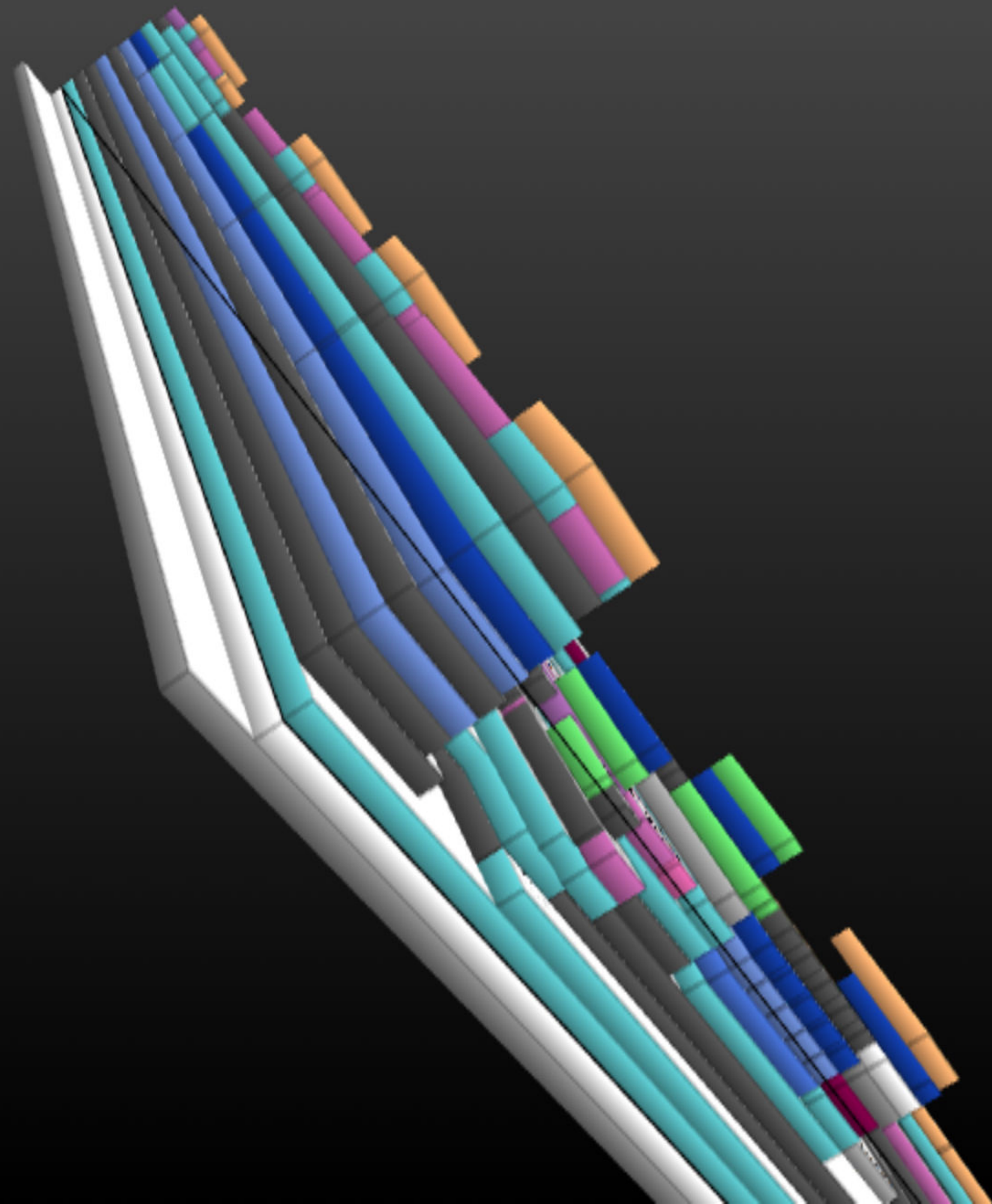


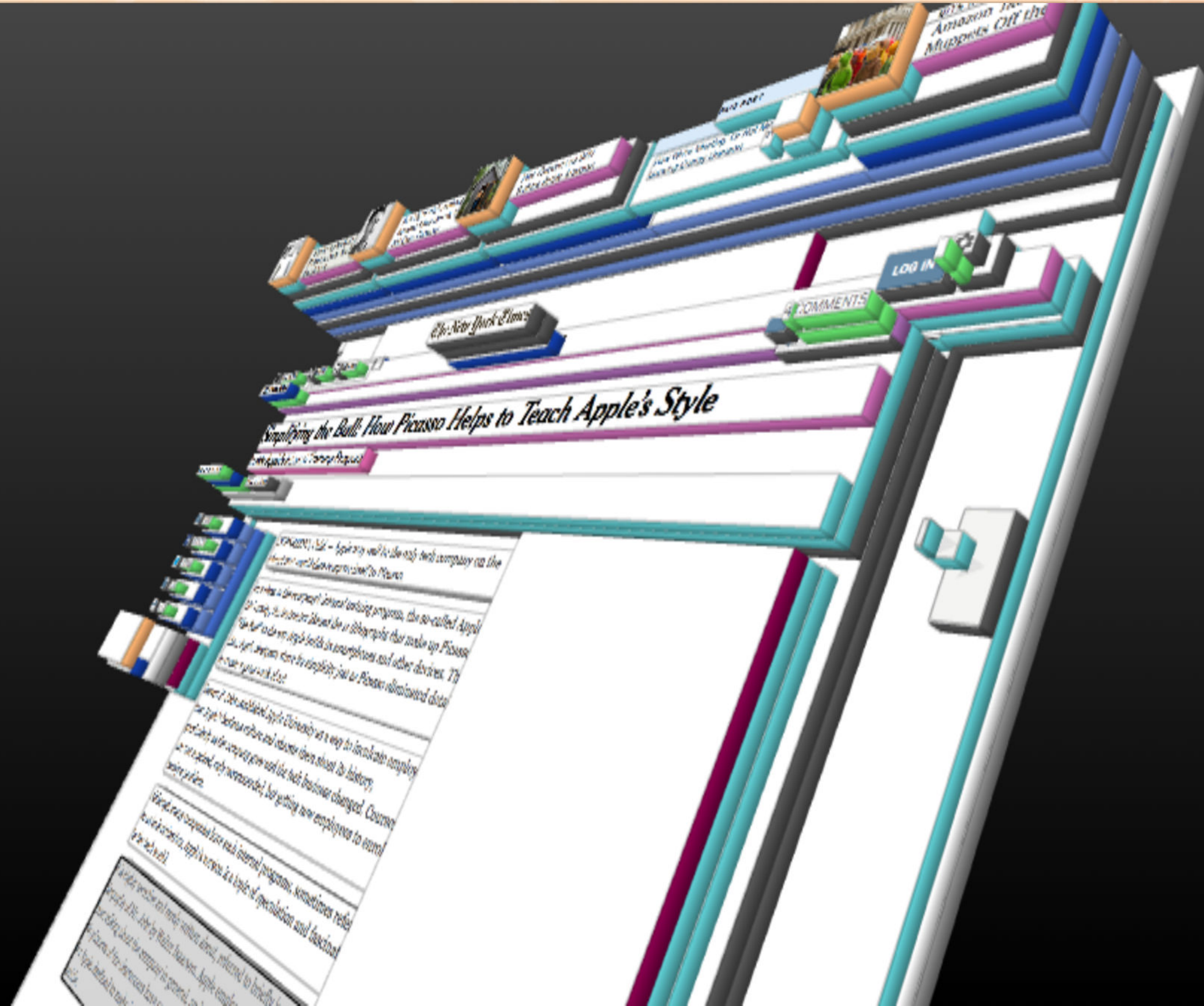
Firefox up to version 47 had a cool feature called *3D View* that let you “rotate and re-orient the 3D presentation of the DOM hierarchy of your page to see it from different angles”

Each box you see is a DOM node

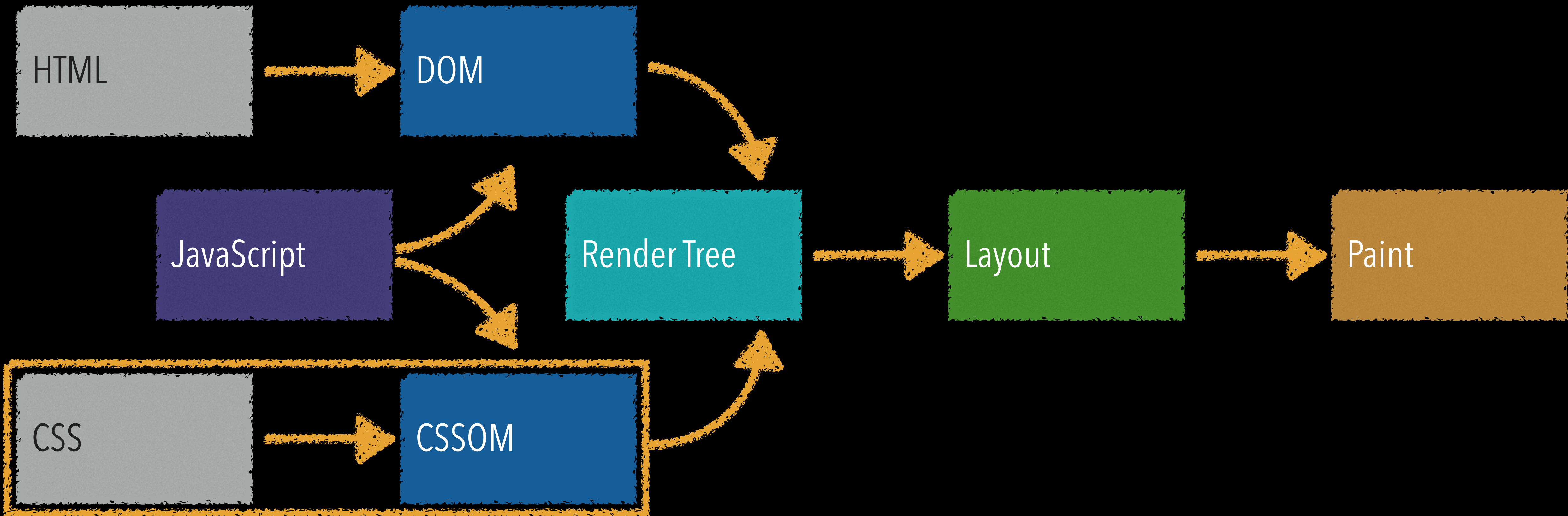








The CSSOM

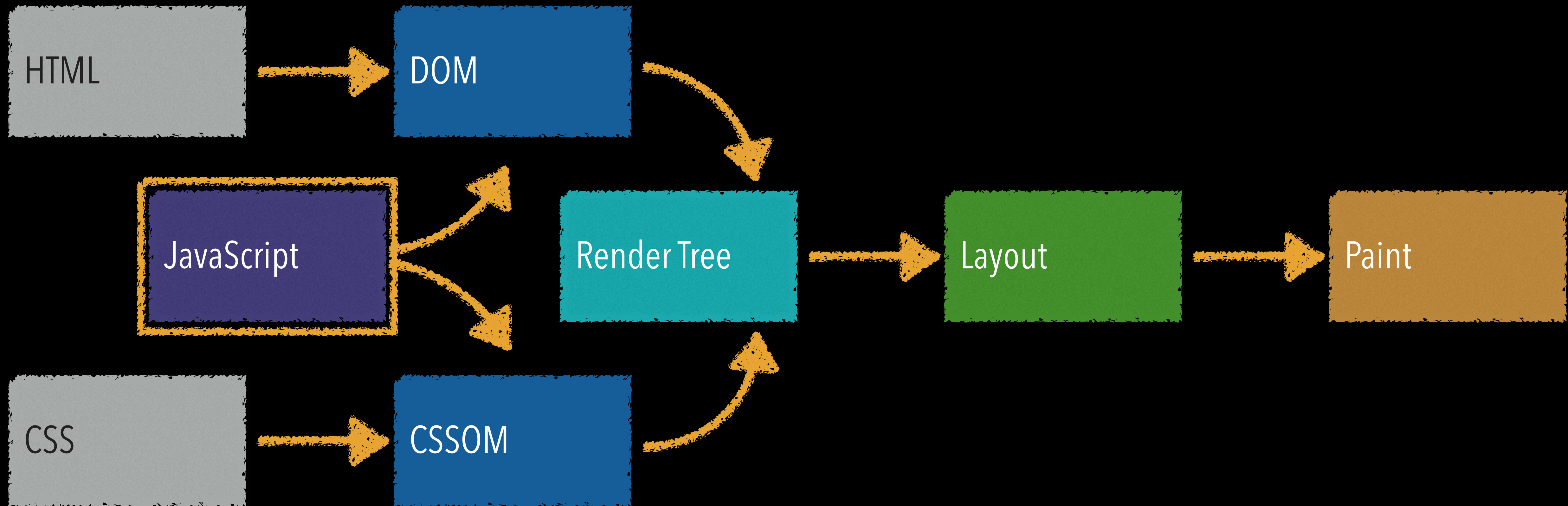


The CSSOM is built by the rendering engine using specified stylesheet rules from:

- » built-in rules that come with the *browser*    
- » rules added by *user*  
- » rules created by CSS *authors* (developers) 

APIs allow developers to manipulate the CSSOM using JavaScript

Bringing It All Together



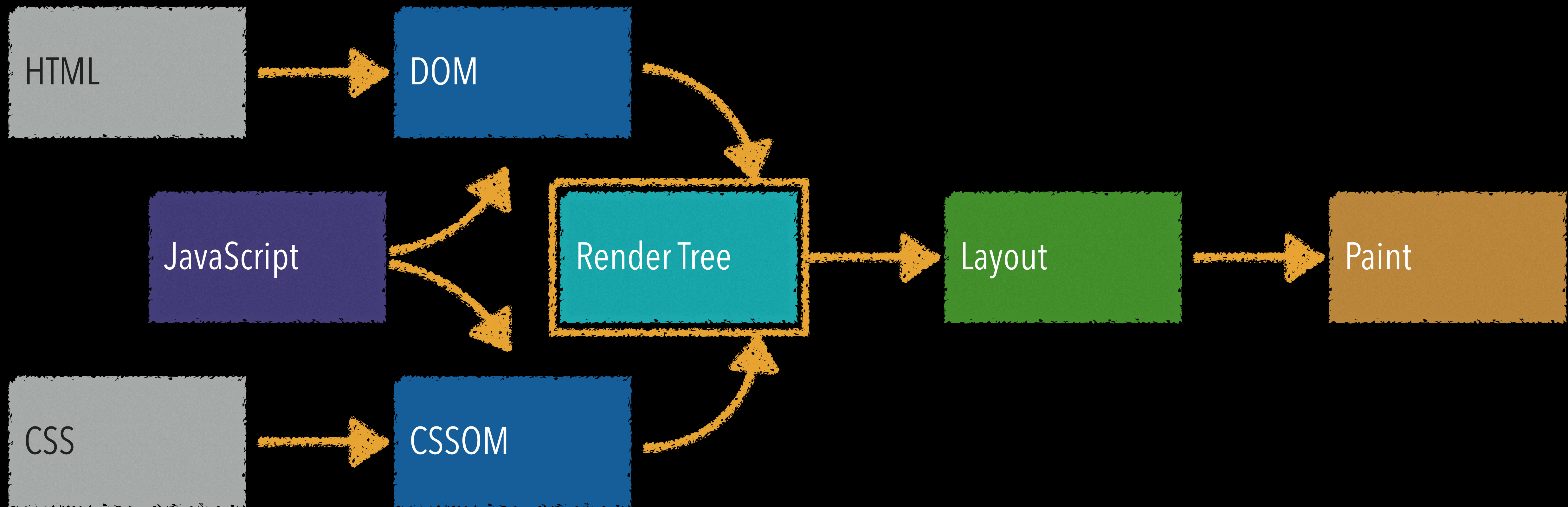
JavaScript can change the DOM & the CSSOM

As of February 2023, there are 140 Web APIs (107 in 2022; 69 in 2020)

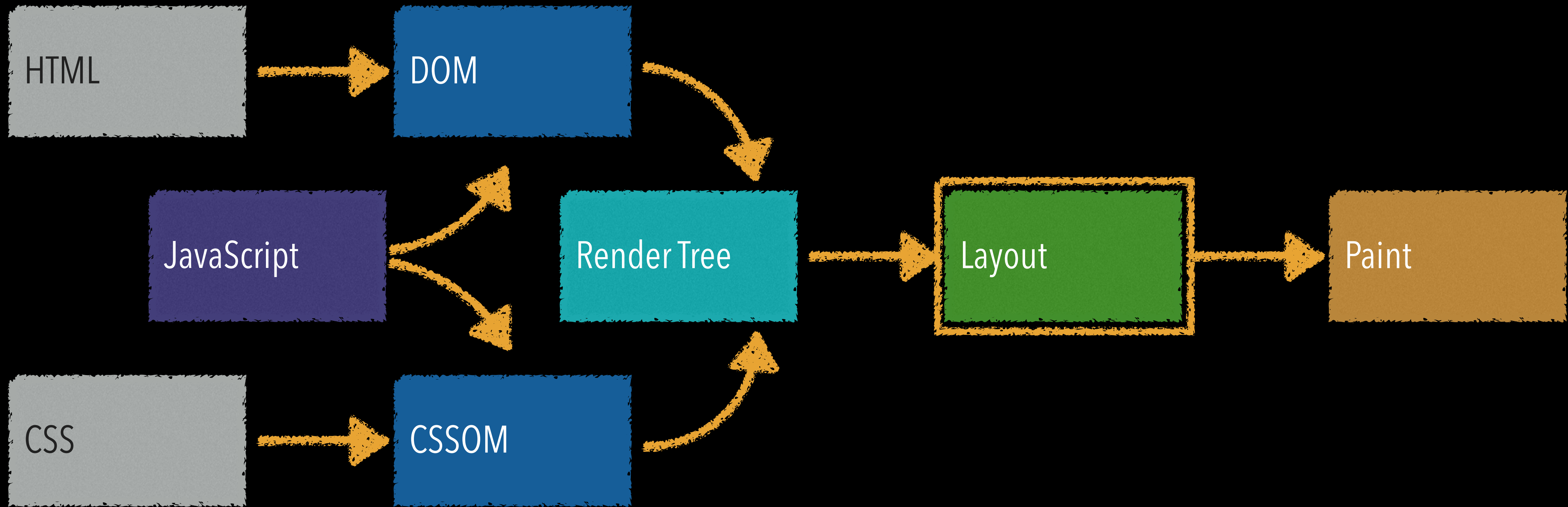
- » 95 available
- » 43 experimental
- » 3 non-standard
- » 1 deprecated

1 non-standard is experimental & 1 is deprecated

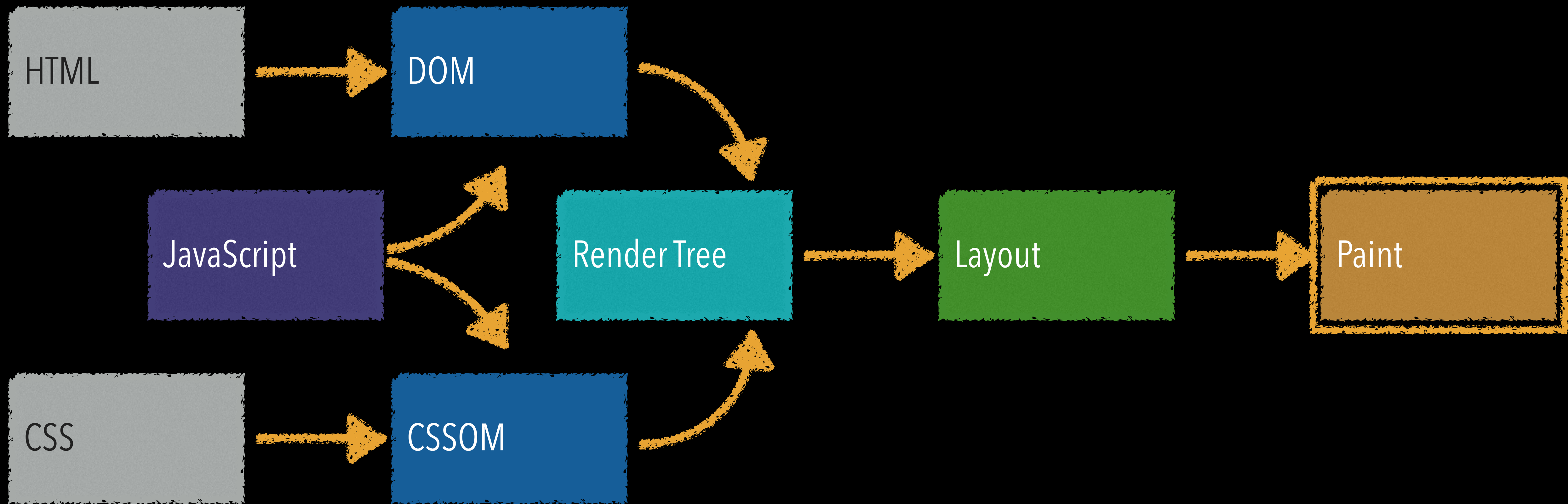
95 Web APIs: Background Synchronization • Background Tasks • Badging • Battery Status • Beacon • Broadcast Channel • CSS Custom Highlight • CSS Font Loading • CSS Properties and Values • CSS Typed Object Model • CSSOM • Canvas • Channel Messaging • Clipboard • Compression Streams • Console • Cookie Store • Credential Management • Document Object Model (DOM) • Device Memory • Device orientation events • Encoding • Encrypted Media Extensions • Fetch • File • File System • File and Directory Entries • Fullscreen • Gamepad • Geolocation • Geometry interfaces • The HTML DOM • HTML Drag and Drop • History • Houdinis • MediaStream Image Capture • IndexedDB • Insertable Streams for MediaStreamTrack • Intersection Observer • Media Capabilities • Media Capture and Streams • Media Session • MediaStream Recording • Network Information • Page Visibility • Payment Request • Performances • Permissions • Picture-in-Picture • Pointer events • Pointer Lock • Popover • Prioritized Task Scheduling • Push • Remote Playback • Resize Observer • SVG • Screen Capture • Screen Orientation • Screen Wake Lock • Selection • Sensors • Server-sent events • Service Worker • Storage • Storage Access • Streams • Touch events • Trusted Types • UI Events • URL • URL Fragment Text Directives • Vibration • View Transition • Visual Viewport • Web Animations • Web Audio • Web Authentication • Web Components • Web Crypto • Web Locks • Web MIDI • Notifications • Web Share • Web Speech • Web Storage • Web Workers • WebCodecs • WebGL • WebOTP • WebRTC • WebSockets • WebTransport • WebVTT • XMLHttpRequest



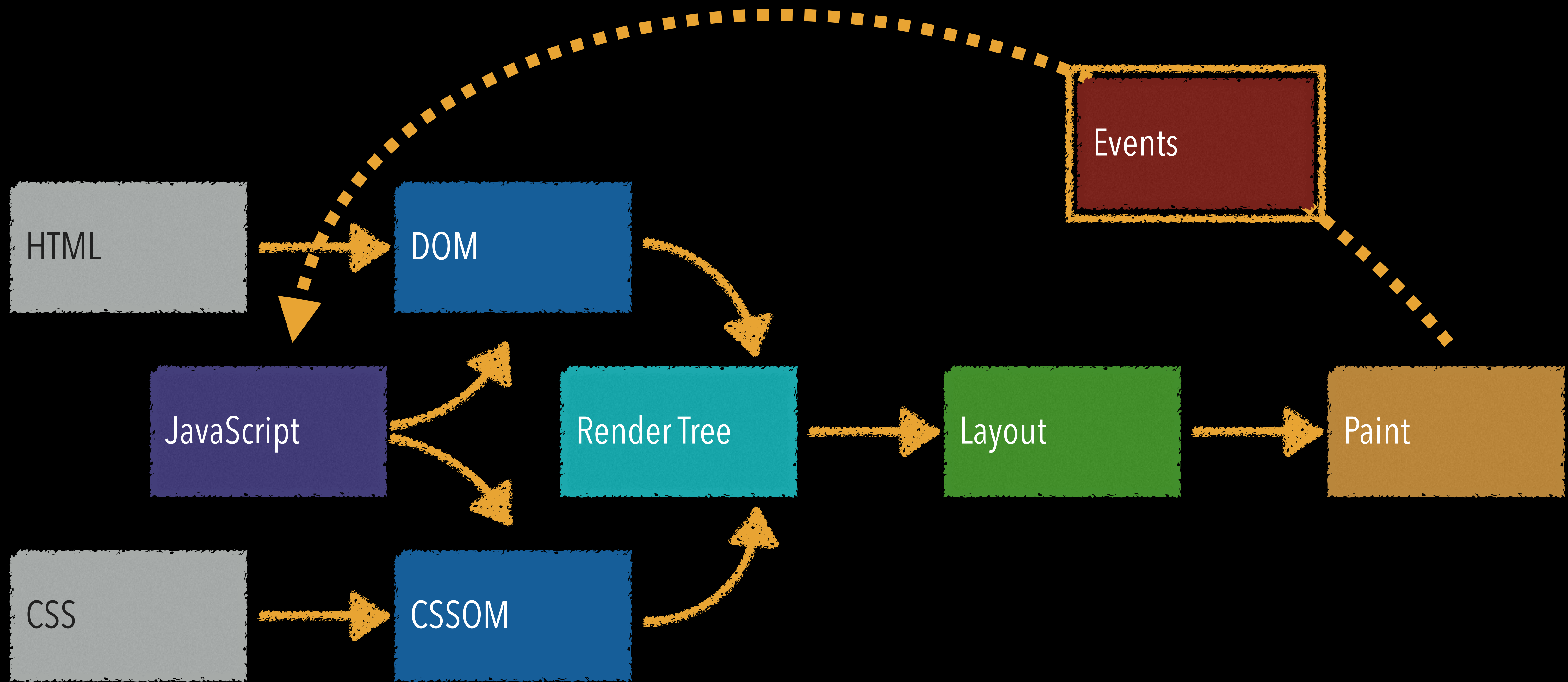
The rendering engine matches HTML element objects with CSS rule objects to generate the Render Tree



The rendering engine uses the Render Tree to lay out boxes in the viewport



The rendering engine displays (*paints*) all the content of the page in the viewport, from back to front



JavaScript events can further change the DOM & the CSSOM

The combination of the DOM, CSSOM, & JavaScript means that webpages are *dynamic* in that JavaScript can:

- » add, change, & remove any DOM elements & attributes
- » change any CSSOM styles
- » react to all existing events
- » create new events

So, when you're working with CSS, you are *styling*
boxes that were rendered from DOM objects

Basic Selectors

A CSS *selector* declares which DOM objects should have particular styles applied to them

The browser's rendering engine...

- » parses the HTML & CSS
- » matches selectors to the appropriate DOM objects
- » applies the CSS style to the rendered DOM objects

HTML	CSS
<code><h1 align="center"></code>	<code>h1 {text-align: center;}</code>
Entire thing is a <i>tag</i>	Entire thing is a <i>style rule</i>
<code>h1</code> is an <i>tag name</i>	<code>h1</code> is a <i>selector</i>
<code>align</code> is an <i>attribute</i>	<code>text-align</code> is a <i>property</i>
<code>center</code> is a <i>value</i>	<code>center</code> is a <i>value</i>
<code>align="center"</code> is an <i>attribute-value pair</i>	<code>text-align: center</code> is a <i>declaration</i>
	Everything inside <code>{ & }</code> is a <i>declaration block</i>

```
/* Everything below is a style rule */  
footer {           /* Start declaration block */  
  padding: 1rem;    /* Declaration */  
  border-top: 1px solid black; /* Declaration */  
}                  /* End declaration block */
```

footer is a *selector*

padding & border-top are *properties*

1rem & 1px solid black are *values*

CSS 1: 10 different selectors (including selectors,
combinators, pseudo-classes, & pseudo-elements)

CSS 2: 13

CSS 3: 22

CSS 4: 53 (as of Nov. 2022)

98 in total

1. Simple selectors

- » Type
- » Class
- » ID
- » Universal
- » Pseudo-classes
- » Pseudo-elements
- » Attribute

2. Compound selectors

3. Complex selectors with combinators

- » Descendant
- » Child
- » Adjacent sibling
- » General sibling

4. Selector list

Simple Selectors

A simple selector describes a single condition on an element

If the condition is true, the element is selected

element: Is this the specific type of element?

.class: Does this element have this class on it?

#ID: Does this element have this ID on it?

***** (universal): Is this an element?

Type

element

Selects *every matching HTML element* (e.g., `<p>`, `<u1>`, or `<h3>`)

AKA Element Selectors

Used when you want to affect every instance of an element (you can then override selected elements with other CSS, e.g., `.class`)

⚙ HTML

```
1 <h3>H. P. Lovecraft's "The Dunwich  
Horror"</h3>  
2 <p>  
3 About 1923, when Wilbur was a boy of  
ten whose mind, voice, stature, and  
bearded face gave all the impressions  
of maturity, a second great siege of  
carpentry went on at the old house. It  
was all inside the sealed upper part,  
and from bits of discarded lumber  
people concluded that the youth and  
his grandfather had knocked out all  
the partitions and even removed the  
attic floor, leaving only one vast  
open void between the ground story and
```

⚙ CSS

```
1 p {  
2   color: red;  
3   font-weight: bold;  
4 }  
5  
6 /* Uninteresting stuff below here */  
7  
8 html {  
9   font-size: 24px;  
10  font-family: "Iowan Old Style",  
    Georgia, serif;  
11 }
```

⚙ JS

H. P. Lovecraft's "The Dunwich Horror"

About 1923, when Wilbur was a boy of ten whose mind, voice, stature, and bearded face gave all the impressions of maturity, a second great siege of carpentry went on at the old house. It was all inside the sealed upper part, and from bits of discarded lumber people concluded that the youth and his grandfather had knocked out all the partitions and even removed the attic floor, leaving only one vast open void between the ground story and the peaked roof. They had torn down the great central chimney, too, and fitted the rusty range with a flimsy outside tin stovepipe.

In the spring after this event Old Whateley noticed the growing number of whippoorwills that would come out of Cold Spring Glen to chirp under his window at night. He seemed to regard the circumstance as one of great significance, and told the loungers at Osborn's that he thought his time had almost come.

Class

`.class`

Selects any *element to which the class has been applied*,
as many times per page as needed

HTML:

```
<p class="intro">..</p>
```

CSS:

```
.intro {  
    font-weight: bold;  
}
```

HTML

1

<p class="attention">

2

Be careful when it is feeding

time for the Yog-Sothoths on

display.

3

</p>

4

<hr>

5

<p>

6

We hope you enjoy your stay at

the Lovecraft Zoo! We have many fun

creatures that the whole family

will enjoy. However, many of our

creatures can cause injury and/or

madness, so <span

class="attention">stay

alert.

7

</p>

CSS

1

.attention {

2

color: red;

3

font-weight: bold;

4

}

JS

Be careful when it is feeding time for the Yog-Sothoths on display.

We hope you enjoy your stay at the Lovecraft Zoo! We have many fun creatures that the whole family will enjoy. However, many of our creatures can cause injury and/or madness, so **stay alert**.

Dot in front of the class name in CSS, but no dot in HTML

Dot labels the class in CSS

The **class** attribute labels the element in HTML

HTML

1

<p class="attention">

2

Be careful when it is feeding

3

time for the Yog-Sothoths on

4

display.

5

</p>

6

<hr>

7

<p>

8

We hope you enjoy your stay at

9

the Lovecraft Zoo! We have many fun

10

creatures that the whole family

11

will enjoy. However, many of our

12

creatures can cause injury and/or

13

madness, so <span

14

class="attention">stay

15

alert.

16

</p>

CSS

1

.attention {

2

color: red;

3

font-weight: bold;

4

}

JS

Be careful when it is feeding time for the Yog-Sothoths on display.

We hope you enjoy your stay at the Lovecraft Zoo! We have many fun creatures that the whole family will enjoy. However, many of our creatures can cause injury and/or madness, so **stay alert**.

The paragraph with **class** applied to it is different from the paragraph that does not have that **class**

HTML

1

<p class="attention">

2

Be careful when it is feeding

time for the Yog-Sothoths on

display.

3

</p>

4

<hr>

5

<p>

6

We hope you enjoy your stay at

the Lovecraft Zoo! We have many fun

creatures that the whole family

will enjoy. However, many of our

creatures can cause injury and/or

madness, so <span

class="attention">stay

alert.

7

</p>

CSS

1

.attention {

2

color: red;

3

font-weight: bold;

4

}

JS

Be careful when it is feeding time for the Yog-Sothoths on display.

We hope you enjoy your stay at the Lovecraft Zoo! We have many fun creatures that the whole family will enjoy. However, many of our creatures can cause injury and/or madness, so **stay alert**.

A class can be applied as many times per page as you wish

You can use multiple (2, 3, 4, whatever) classes on an element when needed

```
HTML
1 <figure>
2   
3   <figcaption class="center image-caption">
4     Libby was a cute puppy
5   </figcaption>
6 </figure>

CSS
1 .center {
2   text-align: center;
3 }
4
5 .image-caption {
6   font-size: .9rem;
7   font-style: italic;
8 }
9

JS
```



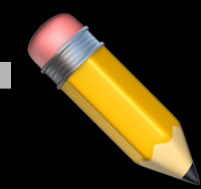
Libby was a cute puppy

Some frameworks, like Tailwind, encourage the use of many classes on elements

```
<div class="p-5 mb-4 bg-light rounded-bottom  
text-light hero maintenance">
```

```
<footer class="container-fluid bg-light  
border-top border-2 border-success mt-auto  
py-3 px-3">
```

This is a bit controversial among developers



SIDE NOTE

A student once did this:

```
<figcaption class="center" class="image-  
caption"> 🤔
```

This will *not* work! No duplicate attributes!

You *must* use:

```
<figcaption class="center image-caption">
```

There is no default list of class names

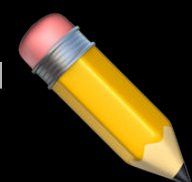
You come up with the class names your project uses (or you use those provided by a framework like Bootstrap)

The one hard rule for class names: no spaces!

Multiple words in a class name?


`.main-content-nav` ← What most developers use
`.main_content_nav`
`.maincontentnav`
`.mainContentNav`
`.MainContentNav`

Just be consistent!




SIDE NOTE


`.main-content-nav` ← kebab-case

`.main_content_nav` ← snake_case 

`.MAIN_CONTENT_NAV` ← SCREAMING_SNAKE_CASE



`.maincontentnav` ← flatcase or lazycase 

`.mainContentNav` ← lowerCamelCase 

`.MainContentNav` ← UpperCamelCase /PascalCase

In fact, you can use emoji for class names (but don't)

```
• 📰 {  
    background-color: hsl(0,0%,76%);  
}
```

```
• 🤮 {  
    font-family: "Comic Sans", cursive;  
}
```

```
<p class="🤮">I have no taste in design!</p>
```

The big rule for class names: *describe function, not appearance*

Not “what does it look like?” but instead “what is it for?”

Bad class names

`.author name`
`.big-red`
`.small`

Good class names

`.author-name`
`.caption`
`.alert`
`.footnote`
`.center`

`.center` is unique, as it is
both a function *and*
appearance

ID

`#id`

Selects any *element to which the ID has been applied*, but each ID can only be used once per page

You can use multiple IDs on a page, but each ID must be unique

`#id` & `.class` share a lot of similarities

- » Names are made up by developers, not specs
- » Names can't have spaces
- » Names should denote purpose, not appearance
- » Names containing multiple words should be combined with a `-` (or something else consistently)

```
HTML
1 <article>
2   <h3>“The Shadow Over Innsmouth” by H. P. Lovecraft</h3>
3   <p></p>
6   <aside id="pull-quote">
7     There were all sorts of
rumours about how Marsh had come
to look.
8   </aside>
9   <p></p>
12 </article>
14 <aside id="sidebar">
15   <h3>August Derleth</h3>
16   <p></p>
19 </aside>
20
21
22
23
24
25
26
27
28

CSS
JS
```

“The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities. His sons had formerly conducted the office in the square, but latterly they had been keeping out of sight a good deal and leaving the brunt of affairs to the younger generation. The sons and their sisters had come to look very queer, especially the elder ones; and it was said that their health was failing.

August Derleth

Best remembered as the first publisher of the writings of H. P. Lovecraft, and for his own contributions to the Cthulhu Mythos genre of horror.

```
HTML
1 <article>
2   <h3>“The Shadow Over Innsmouth” by H. P. Lovecraft</h3>
3   <p></p>
6   <aside id="pull-quote">
7     There were all sorts of
      rumours about how Marsh had come
      to look.
8   </aside>
9   <p></p>
12 </article>
13
14 <aside id="sidebar">

CSS
11 #sidebar {
12   width: 11rem;
13   margin: 0 1em;
14   padding: 1em;
15   box-sizing: border-box;
16 }
17
18 #pull-quote {
19   float: left;
20   width: 12em;
21   margin: .4em .75em 0 0;

JS
```

“The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities. His sons had formerly conducted the office in the square, but latterly they had been keeping out of sight a good deal and leaving the brunt of affairs to the younger generation. The sons had been in the habit of coming to look very queer, especially when they were out, and it was said that their health

August Derleth

Best remembered as the first publisher of the writings of H. P. Lovecraft, and for his own contributions to the Cthulhu Mythos genre of horror.

A specific ID can be applied only *one time per page*

	Class	ID
CSS	<code>.foo</code>	<code>#bar</code>
HTML	<code>class="foo"</code>	<code>id="bar"</code>
Unique on page	No	Yes
	<code>class="foo" 1-∞</code>	<code>id="bar" once</code>
Values per element	Multiple	One
	<code>class="foo bar"</code>	<code>id="baz"</code>
Specificity	10×	100×
URL addressable	No	Yes

Bad ID names

#top nav

#tiny-little-fonts

Good ID names

#sidebar (use <aside>)

#utility-nav (use <nav>)

#site-footer (use
<footer>)

#logo

#legalese (use <small>)

We've stopped using IDs for CSS over the last 10+ years

Use classes instead of IDs for selectors

- » Classes can be reused, while IDs cannot
- » IDs make the *cascade* (more about that soon!) very complicated

Many (most?) frameworks (like Bootstrap) stick to classes entirely

However, even if we don't use IDs for CSS selectors like we used to, they are still necessary

- » Page fragment identifiers
- » Unique hooks for JavaScript

Page Fragment Identifiers

Start with an element that has an `id` on it:

```
<h3 id="the-call-of-cthulhu">
```

Create a link pointing to that fragment identifier:

```
<a href="#the-call-of-cthulhu">
```

Clicking on that link will jump the page down to the

```
<h3>
```

All Presentations

[Home](#) > [Presentations](#) > [All Presentations](#)

On this page...

1. [Web Development](#)

1. [Scott & Jans: An Intro](#)
2. [Web Development Intro for Students: The Absolute Basics](#)
3. [Installing, Configuring, & Using Visual Studio Code](#)
4. [HTML Overview: Elements, Attributes, Values, Entities](#)
5. [HTML Overview Condensed: Elements, Attributes, Values, Entities \(for short courses\)](#)
6. [Bonus: Markdown](#)
7. [It's Boxes All the Way Down!: Elements & Their Relationships](#)
8. [CSS Overview: Selectors, Integration, Inheritance, Cascading](#)
9. [CSS Overview Condensed: Selectors, Integration, Inheritance, Cascading \(for short courses\)](#)

10. [CSS Building Blocks: Selectors](#)

11. [CSS Data Types](#)

12. [CSS Preprocessors: Extending CSS with Sass to Provide Extra](#)

Functionality

13. [CSS Typography: Fonts & Formatting](#)

14. [Tables & Lists: Organized Data](#)

15. [CSS Layout: The Visual Formatting Model](#)

I am currently on www.granneman.com/all-presentations/index.html

All Presentations

[Home](#) > [Presentations](#) > [All Presentations](#)

On this page...

1. [Web Development](#)

1. [Scott & Jans: An Intro](#)
2. [Web Development Intro for Students: The Absolute Basics](#)
3. [Installing, Configuring, & Using Visual Studio Code](#)
4. [HTML Overview: Elements, Attributes, Values, Entities](#)
5. [HTML Overview Condensed: Elements, Attributes, Values, Entities \(for short courses\)](#)
6. [Bonus: Markdown](#)
7. [It's Boxes All the Way Down!: Elements & Their Relationships](#)
8. [CSS Overview: Selectors, Integration, Inheritance, Cascading](#)
9. [CSS Overview Condensed: Selectors, Integration, Inheritance, Cascading \(for short courses\)](#)

10. [CSS Building Blocks: Selectors](#)

11. [CSS Data Types](#)

12. [CSS Preprocessors: Extending CSS with Sass to Provide Extra](#)

Functionality

13. [CSS Typography: Fonts & Formatting](#)

14. [Tables & Lists: Organized Data](#)

15. [CSS Layout: The Visual Formatting Model](#)

I am currently on www.granneman.com/all-presentations/index.html

HTMLTidy×

1<h3 id="toc">Table of Contents</h3>

2

3The Tomb

4Dagon

5Beyond the Walls of Sleep

6

7

8<h3 id="the-tomb">The Tomb [TOC]</h3>

9<p>In relating the circumstances which have led to my confinement...</p>

10

11<h3 id="dagon">Dagon [TOC]</h3>

12<p>I am writing this under an appreciable mental strain...</p>

13

14<h3 id="beyond-the-walls">Beyond the Walls of Sleep [TOC]</h3>

15<p>I have frequently wondered if the majority of mankind...</p>

CSSTidyView Compiled×

JS

Table of Contents

- [The Tomb](#)
- [Dagon](#)
- [Beyond the Walls of Sleep](#)

The Tomb [\[TOC\]](#)

In relating the circumstances which have led to my confinement...

Dagon [\[TOC\]](#)

I am writing this under an appreciable mental strain...

Beyond the Walls of Sleep [\[TOC\]](#)

I have frequently wondered if the majority of mankind...

Nothing in CSS re: `#dagon`!

Say I have this HTML on `http://hpl.com/tales.html`:

```
<h2 id="cthulhu">The Call of Cthulhu</h2>
```

If I want to link to `<h2 id="cthulhu">` from somewhere on `http://hp1.com/tales.html`, I simply use this:

```
<a href="#cthulhu">
```

This means “look on the current page for an element with an `id` of `cthulhu` & go to it”

What if I am on `http://hpl.com/bio.html` & I want to link to `<h2 id="cthulhu">`, which is on `http://hpl.com/tales.html`?

```
<a href="/tales.html#cthulhu">
```

This means “look in the root of this same website for a page named `tales.html`, then look for an element with an `id` of `cthulhu` on that page, & go to it”

What if I am at `https://weirdtales.com/lovecraft.html` (a completely different website) & I want to link to `<h2 id="cthulhu">` on `http://hpl.com/tales.html`?

```
<a href="http://hpl.com/tales.html#cthulhu">
```

This means “go find a website at the domain `hpl.com`, then look in the root of this website for a page named `tales.html`, then look for an element with an `id` of `cthulhu` on that page, & go to it”

JavaScript Hooks

```
HTML
1 <p>
2   <a href="javascript:void(0)" onclick="toggle_show('answer');">
3     Toggle visibility
4   </a>
5 </p>
6 <p id="answer" style="display:none;">
7   Foo
8 </p>

CSS
Tidy View Compiled

JS
1 function toggle_show(id) {
2   var e = document.getElementById(id);
3   if(e.style.display == 'block')
4     e.style.display = 'none';
5   else
6     e.style.display = 'block';
7 }
```

Toggle visibility

This parameter...

...helps the JavaScript...

...target this ID

⚙ HTML

Tidy ✕

```
1 <p>
2   <a href="javascript:void(0)" onclick="toggle_show('answer');">
3     Toggle visibility
4   </a>
5 </p>
6 <p id="answer" style="display:none;">
7   Foo
8 </p>
```

Toggle visibility

Foo

⚙ CSS

Tidy View Compiled ✕

⚙ JS

Tidy ✕

```
1 function toggle_show(id) {
2   var e = document.getElementById(id);
3   if(e.style.display == 'block')
4     e.style.display = 'none';
5   else
6     e.style.display = 'block';
7 }
```

Universal

*

Selects *every HTML element*

(Though later CSS rules can override these declarations)

HTML

1

<h3></h3>

2

<p></p>

5

<p></p>

8

<hr>

9

<p></p>

12

13

Dagon

14

The Terrible Old Man

15

The Colour Out of Space

16

CSS

JS

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

- Dagon
- The Terrible Old Man
- The Colour Out of Space

Once upon a time, Web browsers didn’t always agree on how much **margin** & **padding** to use...

```
HTML
```

```
1 <h3></h3>
2 <p></p>
5 <p></p>
8 <hr>
9 <p></p>
12 <ul>
13   <li>Dagon</li>
14   <li>The Terrible Old
    Man</li>
15   <li>The Colour Out of
    Space</li>
16 </ul>
```

```
* CSS
```

```
1 * {
2   margin: 0;
3   padding: 0;
4 }
```

```
JS
```

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car. There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

Dagon

The Terrible Old Man

The Colour Out of Space

Use *** to *zero out* everything so you can specify **margin** & **padding** for every element

HTML

```
1 <h3></h3>
2 <p></p>
5 <p></p>
8 <hr>
9 <p></p>
12 <ul>
13   <li>Dagon</li>
14   <li>The Terrible Old
    Man</li>
15   <li>The Colour Out of
    Space</li>
16 </ul>
```

* CSS

```
1 * {
2   margin: 0;
3   padding: 0;
4 }
```

JS

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car. There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

Dagon

The Terrible Old Man

The Colour Out of Space

Now the developer has to re-set the **margin** & **padding** for everything, so all browsers are the same

```
HTML
1 <h3></h3>
2 <p></p>
5 <p></p>
8 <hr>
9 <p></p>
12 <ul>
13 <li>Dagon</li>
14 <li>The Terrible Old Man</li>
15 <li>The Colour Out of Space</li>
16 </ul>

CSS
1 * {
2   margin: 0;
3   padding: 0;
4 }
5
6 body {
7   padding: .5em;
8 }
9

JS
```

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

Dagon

The Terrible Old Man

The Colour Out of Space

Let’s start by putting **padding** back on **<body>** so everything isn’t crammed to the edge

```
HTML
1 <h3></h3>
2 <p></p>
5 <p></p>
8 <hr>
9 <p></p>
12 <ul>
13   <li>Dagon</li>
14   <li>The Terrible Old
    Man</li>
CSS
1 * {
2   margin: 0;
3   padding: 0;
4 }
5
6 body {
7   padding: .5em;
8 }
9
10 * + * {
11   margin-bottom: 1em;
12 }
13
JS
```

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

Dagon

The Terrible Old Man

The Colour Out of Space

* + * matches any element that has another sibling element before it

```
HTML
1 <h3></h3>
2 <p></p>
5 <p></p>
8 <hr>
9 <p></p>
12 <ul>
13   <li>Dagon</li>
14   <li>The Terrible Old
    Man</li>
CSS
1 * {
2   margin: 0;
3   padding: 0;
4 }
5
6 body {
7   padding: .5em;
8 }
9
10 * + * {
11   margin-bottom: 1em;
12 }
13
JS
```

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

Dagon

The Terrible Old Man

The Colour Out of Space

`<h3>` doesn’t have a previous sibling, so no margin below it

```
HTML
1 <h3></h3>
2 <p></p>
5 <p></p>
8 <hr>
9 <p></p>
12 <ul>
13 <li>Dagon</li>
14 <li>The Terrible Old
    Man</li>

CSS
1 * {
2   margin: 0;
3   padding: 0;
4 }
5
6 body {
7   padding: .5em;
8 }
9
10 * + * {
11   margin-bottom: 1em;
12 }
13

JS
```

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

Dagon

The Terrible Old Man

The Colour Out of Space

`<p>` has `<h3>` as a previous sibling, so there’s a margin below it

```
HTML
1 <h3></h3>
2 <p></p>
5 <p></p>
8 <hr>
9 <p></p>
12 <ul>
13 <li>Dagon</li>
14 <li>The Terrible Old
    Man</li>
CSS
1 * {
2   margin: 0;
3   padding: 0;
4 }
5
6 body {
7   padding: .5em;
8 }
9
10 * + * {
11   margin-bottom: 1em;
12 }
JS
```

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

Dagon

The Terrible Old Man

The Colour Out of Space

Likewise for the next `<p>`, the `<hr>`, the next `<p>`, & the ``

```
HTML
1 <h3></h3>
2 <p></p>
5 <p></p>
8 <hr>
9 <p></p>
12 <ul>
13   <li>Dagon</li>
14   <li>The Terrible Old
    Man</li>
CSS
1 * {
2   margin: 0;
3   padding: 0;
4 }
5
6 body {
7   padding: .5em;
8 }
9
10 * + * {
11   margin-bottom: 1em;
12 }
13
JS
```

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

Dagon

The Terrible Old Man

The Colour Out of Space

The 1st `` doesn’t have a previous sibling inside ``, so no margin under it ...

```
HTML
1 <h3></h3>
2 <p></p>
5 <p></p>
8 <hr>
9 <p></p>
12 <ul>
13   <li>Dagon</li>
14   <li>The Terrible Old
    Man</li>
CSS
1 * {
2   margin: 0;
3   padding: 0;
4 }
5
6 body {
7   padding: .5em;
8 }
9
10 * + * {
11   margin-bottom: 1em;
12 }
JS
```

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

Dagon

The Terrible Old Man

The Colour Out of Space

...but the 2nd & 3rd ``s *do* have a previous sibling inside ``, so there’s a margin under each ...

```
HTML
1 <h3></h3>
2 <p></p>
5 <p></p>
8 <hr>
9 <p></p>
12 <ul>
13 <li>Dagon</li>
14 <li>The Terrible Old
    Man</li>

CSS
1 * {
2   margin: 0;
3   padding: 0;
4 }
5
6 body {
7   padding: .5em;
8 }
9
10 * + * {
11   margin-bottom: 1em;
12 }

JS
```

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

Dagon

The Terrible Old Man

The Colour Out of Space

... which looks terrible!

How do we fix this?

```
HTML
1 <h3></h3>
2 <p></p>
5 <p></p>
8 <hr>
9 <p></p>

CSS
1 * {
2   margin: 0;
3   padding: 0;
4 }
5
6 body {
7   padding: .5em;
8 }
9
10 * + * {
11   margin-bottom: 1em;
12 }
13
14 * + li {
15   margin-bottom: 0;
16 }
17

JS
```

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

Dagon

The Terrible Old Man

The Colour Out of Space

If any `` has a previous sibling, do *not* put a margin under it

```
HTML
1 <h3></h3>
2 <p></p>
5 <p></p>
8 <hr>
9 <p></p>

CSS
1 * {
2   margin: 0;
3   padding: 0;
4 }
5
6 body {
7   padding: .5em;
8 }
9
10 * + * {
11   margin-bottom: 1em;
12 }
13
14 * + li {
15   margin-bottom: 0;
16 }
17

JS
```

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

Dagon

The Terrible Old Man
The Colour Out of Space

The 1st `` lacks a previous sibling, so our original *zero out* — the `*` in the first rule set — applies

```
HTML
1 <h3></h3>
2 <p></p>
5 <p></p>
8 <hr>
9 <p></p>

CSS
1 * {
2   margin: 0;
3   padding: 0;
4 }
5
6 body {
7   padding: .5em;
8 }
9
10 * + * {
11   margin-bottom: 1em;
12 }
13
14 * + li {
15   margin-bottom: 0;
16 }
17

JS
```

From “The Shadow Over Innsmouth” by H. P. Lovecraft

Of course the town’s real business was the refinery, whose commercial office was on the square only a few doors east of where we stood. Old Man Marsh was never seen, but sometimes went to the works in a closed, curtained car.

There were all sorts of rumours about how Marsh had come to look. He had once been a great dandy, and people said he still wore the frock-coated finery of the Edwardian age, curiously adapted to certain deformities.

After reading, also check out:

Dagon

The Terrible Old Man

The Colour Out of Space

The 2nd & 3rd ``s
have a previous sibling,
so no margin! 😎

It's rare to use `*` by itself as a selector, because you rarely want to change every element

Instead, `*` is more useful when it is *scoped*, or limited to just a certain part of the code

An example is coming up in *Child Combinator*

Other simple selectors

- » Attribute selector
- » Pseudo-class
- » Pseudo-element

Those are covered in *CSS Selectors*

1. Simple selectors

- » Type
- » Class
- » ID
- » Universal
- » Pseudo-classes
- » Pseudo-elements
- » Attribute

2. Compound selectors

3. Complex selectors with combinators

- » Descendant
- » Child
- » Adjacent sibling
- » General sibling

4. Selector list

Compound Selectors

A compound selector describes multiple conditions on an element

If all conditions are true, the element is selected

Consists of a chain of simple selectors connected together with a ., but not connected by a combinator (which is coming up next)

p.alert matches

```
<p class="alert">
```

table.inventory matches

```
<table class="inventory">
```

A compound selector is used to match an element that has two classes on it

```
<p class="center note">
```

```
.center {  
  text-align: center;  
}
```

```
.note {  
  font-size: .9em;  
}
```

```
.center.note {  
  font-style: italic;  
}
```

Selects elements with the
`.center` class

Selects elements with the
`.note` class

Selects elements with the
`.center` *and* `.note` classes

⚙ HTML



```
1 <p class="red">
2   Red
3 </p>
4 <p class="yellow">
5   Yellow
6 </p>
7 <p class="red yellow">
8   Orange
9 </p>
```

⚙ CSS



```
1 .red {
2   background-color: red;
3 }
4
5 .yellow {
6   background-color: yellow;
7 }
8
9 .red.yellow {
10  background-color: orange;
11 }
12
```

⚙ JS



Red

Yellow

Orange

1. Simple selectors

- » Type
- » Class
- » ID
- » Universal
- » Pseudo-classes
- » Pseudo-elements
- » Attribute

2. Compound selectors

3. Complex selectors with combinators

- » Descendant
- » Child
- » Adjacent sibling
- » General sibling

4. Selector list

Complex Selectors Using Combinators

A complex selector uses combinator(s) to combine multiple selectors together into 1 selector

`.ws-gallery img { ... }` ← `└` (space)

`ul > li { ... }` ← `>`

`h2 + p.lead { ... }` ← `+`

Combinator	Name	Ex.	Which B is selected?
└ (space)	Descendant	A B	Any descendant of A
>	Child	A > B	Direct children of A
+	Next Sibling	A + B	Next sibling after A
~ (tilde)	Subsequent Siblings	A ~ B	All siblings after A

A combinator *expresses a relationship between selectors*

Selector Subjects

Rendering engines match selectors from right to left

The right-most part of the selector represents the *subject* of the selector, the actual element being selected

Really helpful when understanding complex selectors

Green indicates selected element(s)

ul li

ul > li

ul > li a[title="home"]

.callOut > p:last-child

.ws-header .nav > li > .current

p code, pre code, blockquote code

Descendant Combinator

`selectorA selectorB`

Selects *all* `selectorB` who have `selectorA` as *an ancestor*

`selectorB` can be a child, grandchild, or later *descendant* of `selectorA`

Any `selectorB` who is *not* a descendant of `selectorA` is unaffected

HTML

87

<footer class="border-top border-success border-2 pt-3 mt-auto">

88

<div class="container">

89

<div class="row column-gap-3">

90

<section class="col-md-4 mb-3 mb-md-

107

0"></section>

108

<section class="col-md-4 mb-3 mb-md-

CSS (SCSS)

1

2

3

4

5

6

7

8

<footer> with
3 <section>s

Contact

10 Barnes St
Providence, RI 02906

1-665-284-8548

shoggoth@herbert-
west.com

Legal

© 2021 Herbert West
Landscaping

Legal info:

- Law firm: Akeley,
Dexter, & Thurston
- Collections officer:
Nyarlahotep

Social





HTML



CSS (SCSS)



JS

```
87 <footer class="border-top border-success border-2  
    pt-3 mt-auto">
```

```
88 <div class="container">
```

```
89 <div class="row column-gap-3">
```

```
90 <section class="col-md-4 mb-3 mb-md-  
0"></section>
```

```
107
```

```
108 <section class="col-md-4 mb-3 mb-md-
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

<footer> with
3 <section>s

Contact

1

10 Barnes St
Providence, RI 02906

1-665-284-8548

shoggoth@herbert-
west.com

Legal

2

© 2021 Herbert West
Landscaping

Legal info:

- Law firm: Akeley,
Dexter, & Thurston
- Collections officer:
Nyarlahotep

Social

3



```
HTML
98      </p>
99      <p>
100        <a href="tel:+1-665-284-8548"
class="text-success text-decoration-none link-
underline-success link-offset-2">1-665-284-8548</a>
101      </p>
102      <p>
103      <a
```

```
CSS (SCSS)
1
2
3
4
5
6
7
8
```

Contact

10 Barnes St
Providence, RI 02906

1-665-284-8548

shoggoth@herbert-
west.com

Legal

© 2021 Herbert West
Landscaping

Legal info:

`<a>` inside `<p>` inside `<address>`
inside `<section>` inside `<div>` inside
`<div>` inside `<footer>`

Social



```
HTML
118 <li>Collections officer: <a
href="https://en.wikipedia.org/wiki/Nyarlathotep"
class="text-success text-decoration-none link-
underline-success link-offset-2">Nyarlathotep</a>
</li>
119 </ul>
120 </section>
121

CSS (SCSS)
1
2
3
4
5
6
7
8
```

Contact

10 Barnes St
Providence, RI 02906

1-665-284-8548

shoggoth@herbert-
west.com

<a> inside inside inside
<section> inside <div> inside <div>
inside <footer>

Legal info:

- Law firm: Akeley,
Dexter, & Thurston
- Collections officer:
Nyarlathotep



HTML



CSS (SCSS)



```
118 <li>Collections officer: <a
href="https://en.wikipedia.org/wiki/Nyarlat
class="text-success text-decoration-none li
underline-success link-offset-2">Nyarlathot
</li>
119 </ul>
120 </section>
121
```

When I hover over an `<a>` in the `<footer>`, I want an underline to appear

Some `<a>`s are in `<p>`s & some `<a>`s are in ``s — how do I target *both* using the least CSS?

Contact

10 Barnes St
Providence, RI 02906

1-665-284-8548

shoggoth@herbert-west.com

Legal

© 2020

Landscape

Legal info:

- Law firm: Akeley, Dexter, & Thurston



HTML

```
118 <li>Collections officer: <a
href="https://en.wikipedia.org/wiki/Nyarlathotep"
class="text-success text-decoration-none link-
underline-success link-offset-2">Nyarlathotep</a>
</li>
119 </ul>
120 </section>
121
```

CSS (SCSS)

```
1 footer a:hover {
2   text-decoration: underline !important;
3 }
```

The answer: use a complex selector with a descendant combinator (the `>`)

Contact

10 Barnes St
Providence, RI 02906

1-665-284-8548

shoggoth@herbert-
west.com

Legal

© 2021 Herbert West

Landscaping

Legal info:

- Law firm

Dexter

I don't know where exactly the `<a>`s will be, but I know they will be descendants somewhere in `<footer>`, & that's what the selector targets

Child Combinator

`selectorA > selectorB`

Selects any `selectorB` who is a *direct child* of `selectorA`, not a grandchild or any other descendants

All siblings who are direct children of `selectorA` are selected

Siblings: 2 or more elements that share a parent

Contrasts with the descendant combinator, which selects both direct children & any descendants, no matter how deep

HTML

1

<table class="movies">

2

<thead>

3

<tr>

4

<th>Movie</th>

5

<th>Genre</th>

6

<th>Hero</th>

7

<th>Year</th>

8

</tr>

9

</thead>

10

<tbody>

11

<tr>

12

<td>Die Hard</td>

13

<td>Action</td>

14

<td>John McClane</td>

15

<td>1988</td>

16

</tr>

17

<tr>

18

<td>Star Wars</td>

19

<td>Sci-Fi</td>

20

<td>Luke Skywalker</td>

21

<td>1977</td>

22

</tr>

23

<tr>

24

<td>John Wick</td>

25

<td>Action</td>

CSS

JS

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

The table *header* cells (`<th>`) are in a single table row (`<tr>`) that is inside `<thead>`

The table *data* cells (`<td>`) are in 3 separate `<tr>`s, all of which are inside `<tbody>`

HTML

```
1 <table class="movies">
2   <thead></thead>
10  <tbody>
11    <tr>
12      <td>Die Hard</td>
13      <td>Action</td>
14      <td>John McClane</td>
15      <td>1988</td>
16    </tr>
17    <tr>
18      <td>Star Wars</td>
19      <td>Sci-Fi</td>
20      <td>Luke Skywalker</td>
21      <td>1977</td>
22    </tr>
23    <tr>
24      <td>John Wick</td>
25      <td>Action</td>
26      <td>John Wick</td>
27      <td>2014</td>
28    </tr>
29  </tbody>
30 </table>
```

CSS

JS

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

<thead> has been collapsed
so you can see all of <tbody>

HTML

1

<table class="movies">

2

<thead>

3

<tr>

4

<th>Movie</th>

5

<th>Genre</th>

6

<th>Hero</th>

7

<th>Year</th>

8

</tr>

9

</thead>

10

<tbody>

11

<tr>

12

<td>Die Hard</td>

13

<td>Action</td>

14

<td>John McClane</td>

15

<td>1988</td>

16

</tr>

17

<tr>

18

<td>Star Wars</td>

19

<td>Sci-Fi</td>

20

<td>Luke Skywalker</td>

21

<td>1977</td>

CSS

1

table.movies > tbody > tr:hover {

2

background-color: pink;

3

}

JS

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

As you hover your mouse over any cell in that `<tr>`, the row turns pink — why?

HTML

```
1 <table class="movies">
2   <thead>
3     <tr>
4       <th>Movie</th>
5       <th>Genre</th>
6       <th>Hero</th>
7       <th>Year</th>
8     </tr>
9   </thead>
10  <tbody>
11    <tr>
12      <td>Die Hard</td>
13      <td>Action</td>
14      <td>John McClane</td>
15      <td>1988</td>
16    </tr>
17    <tr>
18      <td>Star Wars</td>
19      <td>Sci-Fi</td>
20      <td>Luke Skywalker</td>
21      <td>1977</td>
```

CSS

```
1 table.movies > tbody > tr:hover {
2   background-color: pink;
3 }
```

JS

Movie	Genre	Hero	Year
Die Hard	Action	John McClane	1988
Star Wars	Sci-Fi	Luke Skywalker	1977
John Wick	Action	John Wick	2014

Why doesn't that `<tr>` turn pink when you hover your mouse over it?

You can often use a child (or descendant) combinator instead of a class

```
<aside>
  
</aside>
```



```
.headshot {}
```

```
<aside>
  
</aside>
```

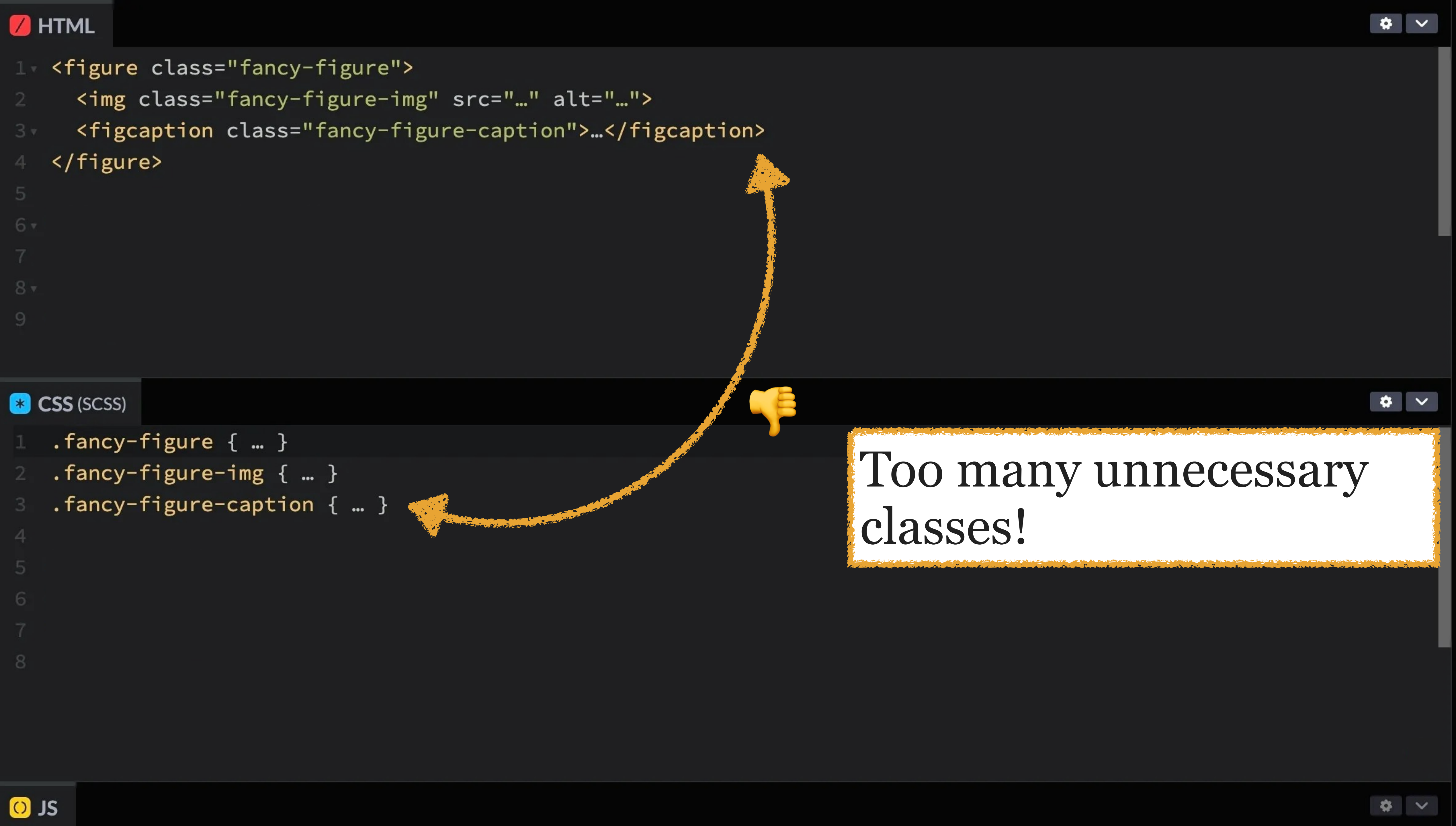


```
aside > img {}
```

No need for a class,
so cleaner HTML

Instead of using a class on every element...

1. place a single class on the *containing* element & then
2. use combinators to select child or descendant type selectors



HTML



```
1 <figure class="fancy-figure">
2   
3   <figcaption class="fancy-figure-caption">...</figcaption>
4 </figure>
```

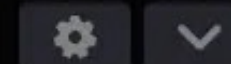
* CSS (SCSS)



```
1 .fancy-figure { ... }
2 .fancy-figure-img { ... }
3 .fancy-figure-caption { ... }
```

Too many unnecessary
classes!

JS



HTML

1<figure class="fancy-figure">

2

3<figcaption class="fancy-figure-caption">...</figcaption>

4</figure>

5

6<figure class="fancy">

7

8<figcaption>...</figcaption>

9</figure>

* CSS (SCSS)

1.fancy-figure { ... }

2.fancy-figure-img { ... }

3.fancy-figure-caption { ... }

4

5figure.fancy { ... }

6figure.fancy > img { ... }

7figure.fancy > figcaption { ... }

8

JS

The 2nd group in both HTML & CSS leverages the HTML structure to make the HTML simpler, while moving any complexity to the CSS

A complex selector allows you to *scope* the universal selector in useful ways

```
HTML
1 <p>I still wonder that we deduced so much in
  the short time at our disposal.</p>
2
3 <blockquote>
4   <p>It may be the effect of this later study
    which has been the immediate source of
    Danforth's present breakdown.</p>
5   <ul>
6     <li>the revived memories and vague
      impressions</li>
7     <li>acting in conjunction with his general
      sensitiveness</li>
8     <li>and with that final supposed horror-
      glimpse whose essence he will not reveal even
      to me</li>
9   </ul>
10 </blockquote>
11
12 <p>Certain lingering influences in that unknown
    antarctic world of disordered time and alien
    natural law make it imperative that further
    exploration be discouraged.</p>

CSS (SCSS)
JS
```

I still wonder that we deduced so much in the short time at our disposal.

It may be the effect of this later study which has been the immediate source of Danforth's present breakdown.

- the revived memories and vague impressions
- acting in conjunction with his general sensitiveness
- and with that final supposed horror-glimpse whose essence he will not reveal even to me

Certain lingering influences in that unknown antarctic world of disordered time and alien natural law make it imperative that further exploration be discouraged.

Let's make it look nice...

```
HTML
2
3 <blockquote>
4   <p>It may be the effect of this later study
   which has been the immediate source of
   Danforth's present breakdown.</p>
5   <ul>
6     <li>the revived memories and vague
       impressions</li>
7     <li>acting in conjunction with his general
       sensitiveness</li>
8     <li>and with that final supposed horror-
       glimpse whose essence he will not reveal even
       to me</li>
9   </ul>
10 </blockquote>
11
12 <p>Certain lingering influences in that unknown
CSS (SCSS)
1 blockquote {
2   background-color: dodgerblue;
3   color: white;
4   padding: 1em;
5 }
```

I still wonder that we deduced so much in the short time at our disposal.

It may be the effect of this later study which has been the immediate source of Danforth's present breakdown.

- the revived memories and vague impressions
- acting in conjunction with his general sensitiveness
- and with that final supposed horror-glimpse whose essence he will not reveal even to me

Certain lingering influences in that unknown antarctic world of disordered time and alien natural law make it imperative that further exploration be discouraged

```
HTML
2
3 <blockquote>
4   <p>It may be the effect of this later study
   which has been the immediate source of
   Danforth's present breakdown.</p>
5   <ul>
6     <li>the revived memories and vague
       impressions</li>
7     <li>acting in conjunction with his general
       sensitiveness</li>
8     <li>and with that final supposed horror-
       glimpse whose essence he will not
       reveal even to me</li>
9   </ul>
10 </blockquote>
11
12 <p>Certain lingering influences in that unknown
    world of disordered time and alien natural law make it
    imperative that further exploration be discouraged</p>
CSS (SCSS)
1 blockquote {
2   background-color: dodgerblue;
3   color: white;
4   padding: 1em;
5 }
JS
```

I don't like that extra space

I still wonder that we deduced so much in the short time at our disposal.

It may be the effect of this later study which has been the immediate source of Danforth's present breakdown.

- the revived memories and vague impressions
- acting in conjunction with his general sensitiveness
- and with that final supposed horror-glimpse whose essence he will not reveal even to me

Certain lingering influences in that unknown antarctic world of disordered time and alien natural law make it imperative that further exploration be discouraged

```
HTML
3 <blockquote>
4   <p>It may be the effect of this later study
  which has been the immediate source of
  Danforth's present breakdown.</p>
5   <ul>
6     <li>the revived memories and vague
  impressions</li>
7     <li>acting in conjunction with his general

CSS (SCSS)
1 blockquote {
2   background-color: dodgerblue;
3   color: white;
4   padding: 1em;
5 }
6
7 blockquote > *:first-child {
8   margin-top: 0;
9 }
10
11 blockquote > *:last-child {
12   margin-bottom: 0;
13 }
```

I still wonder that we deduced so much in the short time at our disposal.

It may be the effect of this later study which has been the immediate source of Danforth's present breakdown.

- the revived memories and vague impressions
- acting in conjunction with his general sensitiveness
- and with that final supposed horror-glimpse whose essence he will not reveal even to me

We scope the `*` so it has limited effect

Certain lingering influences in that unknown antarctic law make it imperative that further exploration be discouraged.

```
HTML
3 <blockquote>
4   <p>It may be the effect of this later study
  which has been the immediate source of
  Danforth's present breakdown.</p>
5   <ul>
6     <li>the revived memories and vague
  impressions</li>
7     <li>acting in conjunction with his general

CSS (SCSS)
1 blockquote {
2   background-color: dodgerblue;
3   color: white;
4   padding: 1em;
5 }
6
7 blockquote > *:first-child {
8   margin-top: 0;
9 }
10
11 blockquote > *:last-child {
12   margin-bottom: 0;
13 }
```

I still wonder that we deduced so much in the short time at our disposal.

It may be the effect of this later study which has been the immediate source of Danforth's present breakdown.

- the revived memories and vague impressions
- acting in conjunction with his general sensitiveness
- and with that final supposed horror-glimpse whose essence he will not reveal even to me

I use `*` because I don't know what the *first* child of `blockquote` might always be

```
HTML
3 <blockquote>
4   <p>It may be the effect of this later study
  which has been the immediate source of
  Danforth's present breakdown.</p>
5   <ul>
6     <li>the revived memories and vague
  impressions</li>
7     <li>acting in conjunction with his general

CSS (SCSS)
1 blockquote {
2   background-color: dodgerblue;
3   color: white;
4   padding: 1em;
5 }
6
7 blockquote > *:first-child {
8   margin-top: 0;
9 }
10
11 blockquote > *:last-child {
12   margin-bottom: 0;
13 }
```

I still wonder that we deduced so much in the short time at our disposal.

It may be the effect of this later study which has been the immediate source of Danforth's present breakdown.

- the revived memories and vague

I use `*` because I don't know what the *last* child of `blockquote` might always be

- and with that final supposed horror-glimpse whose essence he will not reveal even to me

Certain lingering influences in that unknown antarctic world of disordered time and alien natural law make it imperative that further exploration be discouraged.

```
HTML
impressions</li>
7 <li>acting in conjunction with his general
  sensitiveness</li>
8 <li>and with that final supposed horror-
  glimpse whose essence he will not reveal even
  to me</li>
9 </ul>
10 </blockquote>

CSS (SCSS)
1 blockquote {
2   background-color: dodg
3   color: white;
4   padding: 1em;
5 }
6
7 blockquote > *:first-child {
8   margin-top: 0;
9 }
10
11 blockquote > *:last-child {
12   margin-bottom: 0;
13 }
```

C'est parfait!

I still wonder that we deduced so much in the short time at our disposal.

It may be the effect of this later study which has been the immediate source of Danforth's present breakdown.

- the revived memories and vague impressions
- acting in conjunction with his general sensitiveness
- and with that final supposed horror-glimpse whose essence he will not reveal even to me

Certain lingering influences in that unknown antarctic world of disordered time and alien natural law make it imperative that further exploration be discouraged.

An illustration showing the difference between the descendant & child combinators

```
HTML
1 <div>
2   <p></p>
5   <blockquote>
6     <p></p>
9     <p></p>
12  </blockquote>
13  <p></p>
16 </div>
17
18 <!-- From H.P. Lovecraft's "The
    Tomb" at
    http://www.hplovecraft.com
    /writings/texts/fiction/t.aspx
    -->

CSS
1 /* div p:last-child { */
2 div > p:last-child {
3   font-family: Creepster,
  fantasy;
4   font-size: 1.2em;
5 }
6

JS
```

On the following day I was brought to this room with the barred windows, but I have been kept informed of certain things through an aged and simple-minded servitor, for whom I bore a fondness in infancy, and who like me loves the churchyard.

What I have dared relate of my experiences within the vault has brought me only pitying smiles. My father, who visits me frequently, declares that at no time did I pass the chained portal, and swears that the rusted padlock had not been touched for fifty years when he examined it.

He even says that all the village knew of my journeys to the tomb, and that I was often watched as I slept in the bower outside the grim facade, my half-open eyes fixed on the crevice that leads to the interior.

AGAINST THESE ASSERTIONS I HAVE NO TANGIBLE PROOF TO OFFER, SINCE MY KEY TO THE PADLOCK WAS LOST IN THE STRUGGLE ON THAT NIGHT OF HORRORS. THE STRANGE THINGS OF THE PAST WHICH I LEARNT DURING THOSE NOCTURNAL MEETINGS WITH THE DEAD HE DISMISSES AS THE FRUITS OF MY LIFELONG AND OMNIVOROUS BROWSING AMONGST THE ANCIENT VOLUMES OF THE FAMILY LIBRARY.

Using `>` limits the scope of the styles 😊

```
HTML
1 <div>
2   <p></p>
5   <blockquote>
6     <p></p>
9     <p></p>
12  </blockquote>
13  <p></p>
16 </div>
17
18 <!-- From H.P. Lovecraft's "The
    Tomb" at
    http://www.hplovecraft.com
    /writings/texts/fiction/t.aspx
    -->

CSS
1 div p:last-child {
2   /* div > p:last-child { */
3   font-family: Creepster,
    fantasy;
4   font-size: 1.2em;
5 }
6

JS
```

On the following day I was brought to this room with the barred windows, but I have been kept in a room with a single window looking out onto a street with a servitor, for whom I had no use, and a small garden with a single tree in a churchyard.

Using `<p></p>` is not what I wanted! 😡

What I have dared relate of my experiences within the vault has brought me only pitying smiles. My father, who visits me frequently, declares that at no time did I pass the chained portal, and swears that the rusted padlock had not been touched for fifty years when he examined it.

HE EVEN SAYS THAT ALL THE VILLAGE KNEW OF MY JOURNEYS TO THE TOMB, AND THAT I WAS OFTEN WATCHED AS I SLEPT IN THE BOWER OUTSIDE THE GRIM FACADE, MY HALF-OPEN EYES FIXED ON THE CREVICE THAT LEADS TO THE INTERIOR.

AGAINST THESE ASSERTIONS I HAVE NO TANGIBLE PROOF TO OFFER, SINCE MY KEY TO THE PADLOCK WAS LOST IN THE STRUGGLE ON THAT NIGHT OF HORRORS. THE STRANGE THINGS OF THE PAST WHICH I LEARNT DURING THOSE NOCTURNAL MEETINGS WITH THE DEAD HE DISMISSES AS THE FRUITS OF MY LIFELONG AND OMNIVOROUS BROWSING AMONGST THE ANCIENT VOLUMES OF THE FAMILY LIBRARY.

1. Simple selectors

- » Type
- » Class
- » ID
- » Universal
- » Pseudo-classes
- » Pseudo-elements
- » Attribute

2. Compound selectors

3. Complex selectors with combinators

- » Descendant
- » Child
- » Adjacent sibling
- » General sibling

4. Selector list

Selector List

selectorA, selectorB, selectorC

*List selectors that have similar declarations for
simpler & cleaner CSS & HTML*

Don't do this:

```
p {  
  font-family: serif;  
  font-size: 1em;  
}
```

```
blockquote {  
  font-family: serif;  
  font-size: 1em;  
}
```

Do this:

```
blockquote, p {  
  font-family: serif;  
  font-size: 1em;  
}
```

Any selector can be included in a list

```
.emphasis, .title {font-style: italic;}
```

```
blockquote, .title {font-style: italic;}
```

```
nav .menu-btn, nav .menu-item {display: none;}
```

A very common selector list because you want all your headings to use the same font

```
h1, h2, h3, h4, h5, h6 {  
    font-family: "Georgia Pro", Georgia, serif;  
}
```

Group similar selectors, but be specific where needed

CSS is *cumulative* unless overridden

```
blockquote, p {  
    font-family: Verdana, sans-serif;  
}
```

```
p {  
    line-height: 1.5;  
}
```

<p> will use Verdana *and*
have a *line-height* of 1.5

Turn this...

```
h1 {  
    font-weight: normal;  
    font-size: 2.5em;  
    font-family: serif;  
}  
  
h2 {  
    font-weight: normal;  
    border-bottom: 1px  
dotted black;  
    font-family: serif;  
    font-size: 1.8em;  
}
```

...into this:

```
h1, h2 {  
    font-weight: normal;  
    font-family: serif;  
}  
  
h1 {  
    font-size: 2.5em;  
}  
  
h2 {  
    border-bottom: 1px dotted  
black;  
    font-size: 1.8em;  
}
```

Good practice

```
blockquote,  
option,  
p,  
td,  
.info,  
.legalese {  
    font-family: Verdana, sans-serif;  
    font-size: 1em;  
}
```

Each selector on its own line

Elements then classes (no IDs!)

Alphabetical order within
each grouping of selectors

Formatting

Don't do this:

```
h1 {color: dimgray;}
```

```
h1 {font-size: 1.4em;}
```

```
h1 {font-weight: bold;}
```

```
h1 {font-family: Verdana, sans-serif;}
```

Instead, combine related declarations

```
h1 {  
  color: dimgray;  
  font-family: Verdana, sans-serif;  
  font-size: 1.4em;  
  font-weight: bold;  
}
```

Formatting CSS style rules

```
selector {  
    property: value;  
    property: value;  
    property: value;  
    ...  
}
```

The order of declarations in the declaration block
doesn't matter

Do not forget to put ; at the end of every declaration

You do not actually have to put ; at the end of the last line in a style rule, but that is a very bad habit to get into

Use comments in CSS for the same reasons as in HTML

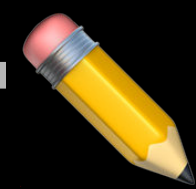
- » *Notes* to yourself & others
- » ~~*Debugging*: comment out troublesome CSS for testing~~
(use your browser's Inspector)

HTML comments

```
<!-- The Doom That Came to Sarnath -->
```

CSS comments

```
/* The Statement of Randolph Carter */
```

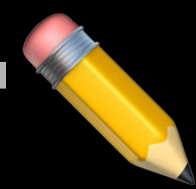


SIDE NOTE

Design Pattern

“a formal way of documenting a solution to a design problem in a particular field of expertise.” —Wikipedia

“Each pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem” —Christopher Alexander, architect & author of *A Pattern Language* (1977)



SIDE NOTE

Navbar

Home

Link

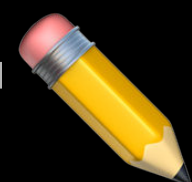
Dropdown ▼

Disabled

HTML



```
<nav class="navbar navbar-expand-lg bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" (
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
```



SIDE NOTE

Image cap

Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

Go somewhere

```
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```



PRO TIP

Here's the order I sometimes shoot for in my stylesheets

```
@font-face
```

```
html
```

```
body
```

```
/* General */
```

```
<type selectors, A→Z>
```

```
<class selectors, A→Z>
```

```
/* <New Section> */
```

```
<type>
```

```
<class>
```

```
...
```


&

<div>

HTML elements “work” without attributes & values,
e.g., `<p>` or `<h2>`

(...with a tiny few exceptions, like `` &
``)

`` & `<div>` are HTML elements that exist solely to work with CSS*

`` & `<div>` by themselves do nothing on a webpage (except draw invisible boxes)

They must use CSS (`class="foo"`) to do anything productive

* & JavaScript

Suddenly the sunshine seemed to lessen without the intervention of **any** discernible cloud. It was a very **peculiar** phenomenon, and was plainly marked by all. A rumbling sound seemed brewing beneath the hills, **mixed strangely with a concordant rumbling** which clearly came from the sky. Lightning flashed aloft, and the wondering crowd **looked in vain** for the portents of storm. The chanting of the men from Arkham now became **unmistakable**, and Wheeler saw through the glass that they were all raising their arms in the rhythmic incantation.

From some farmhouse far away came **the frantic barking of dogs**. The change in the quality of the daylight increased, and the crowd gazed about the horizon in wonder. A **purplish** darkness, born of nothing more than a spectral deepening of the sky's blue, pressed down upon the rumbling hills.

Then the lightning flashed **again**, somewhat brighter than before, and the crowd fancied that it had showed a certain **mistiness** around the altar-stone on the distant height. No one, however, had been using the telescope at that instant.

The whippoorwills continued their **irregular pulsation**, and the men of Dunwich braced themselves tensely against some imponderable menace with which the atmosphere seemed surcharged. Without warning came those deep, cracked, raucous vocal sounds which will **never** leave the memory of the stricken group who heard them.

And because imagination **might** suggest a conjectural source in the world of non-visible beings, the huddled crowd at the mountain's base huddled still closer, and **wincing** as if in expectation of a blow. Ygnailh ... ygnaiih ... thflthkh'ngha ... Yog-Sothoth ... rang the hideous croaking out of space. Y'bthnk ... h'ehye—n'grkdl'lh ...



Block boxes



Inline boxes

` ... `

Creates an *inline box* around a *span of text* so it can be styled with CSS or manipulated with JavaScript

Does *nothing* visually on its own!

Normally you use `` with a class attribute that is then referenced in your CSS

```
<p>
```

```
  The cultists all scream <span class="hpl">Ph'nglui  
mglw'nafh Cthulhu R'lyeh wgah'nagl fhtagn!</span>
```

```
<p>
```

In your CSS you would then refer to `.hpl` & style it:

```
.hpl {  
  font-size: 18px;  
  font-family: Creepster;  
}
```

`` should be used *only* if nothing semantic is appropriate

If you can use `<b class="hpl">` because the text is first bold, or `<i class="hpl">` because the text is first italic, then do so

And so on

```
HTML
4 <p>
5   I heard it, and knew no more. Heard it as I
   sat petrified in that unknown cemetery in the
   hollow, amidst the crumbling stones and the
   falling tombs, the rank vegetation and the
   miasmal vapours. Heard it well up from the
   innermost depths of that damnable open
   sepulchre as I watched amorphous, necrophagous
   shadows dance beneath an accursed waning moon.
   And this is what it said: "<span
   class="eeek">You fool, Warren is dead!</span>"
6 </p>

CSS
1 .eeek {
2   font-family: Creepster, cursive;
3   font-size: 1.2em;
4   /* In case font fails to load */
5   text-transform: uppercase;
6 }
7

JS
```

From H. P. Lovecraft's "The Statement of Randolph Carter":

I heard it, and knew no more. Heard it as I sat petrified in that unknown cemetery in the hollow, amidst the crumbling stones and the falling tombs, the rank vegetation and the miasmal vapours. Heard it well up from the innermost depths of that damnable open sepulchre as I watched amorphous, necrophagous shadows dance beneath an accursed waning moon. And this is what it said: **"YOU FOOL, WARREN IS DEAD!"**

<div>

```
<div> ... </div>
```

Generic container for adding a *division* (a *block box* or new object on the document tree) that can be manipulated with CSS & JavaScript

Does *nothing* visually on its own!

`<div>` is a grouping element that creates a *block box* by default & does nothing else without CSS

Use `<div>` to hold attribute-value pairs relevant to CSS

As a grouping element, it groups other elements, e.g.,
put a `<div>` around 3 `<p>`s & a ``

Use it when other elements (e.g., `<article>`, `<header>`,
or `<aside>`) are not semantically appropriate

```
HTML Tidy x
1 <i>H. P. Lovecraft's </i>The Case of
  Charles Dexter Ward</i>, a
  novella in five parts.
3 </p>
4 <p>
5   I. A Result and a Prologue
6 </p>
7 <p>
8   II. An Antecedent and a Horror
9 </p>
10 <p>
11   III. A Search and an Evocation
12 </p>
13 <p>
14   IV. A Mutation and a Madness
15 </p>
16 <p>
17   V. A Nightmare and a Cataclysm
18 </p>
19 <p>
20   After that, you will write a
    paper looking at the importance
    of place in the novella.

CSS Tidy View Compiled x
JS Tidy x
```

Your assignment is to read H. P. Lovecraft's *The Case of Charles Dexter Ward*, a novella in five parts.

I. A Result and a Prologue

II. An Antecedent and a Horror

III. A Search and an Evocation

IV. A Mutation and a Madness

V. A Nightmare and a Cataclysm

After that, you will write a paper looking at the importance of place in the novella.

I want the table of contents to stand out with a background color, borders, & rounded corners

HTML

Tidy

1

<p>

2

Your assignment is to read H. P. Lovecraft's <i>The Case of Charles Dexter Ward</i>, a novella in five parts.

3

</p>

4

<p class="toc">

5

I. A Result and a Prologue

6

</p>

7

<p class="toc">

8

II. An Antecedent and a Horror

9

</p>

10

<p class="toc">

11

III. A Search and an Evocation

CSS

Tidy

View Compiled

1

.toc {

2

background-color: #E8E8E8;

3

margin: 1em;

4

border: 1px solid #ccc;

5

padding: 0 1em;

6

width: 90%;

7

border-radius: 1em;

8

}

JS

Tidy

Your assignment is to read H. P. Lovecraft's *The Case of Charles Dexter Ward*, a novella in five parts.

I. A Result and a Prologue

II. An Antecedent and a Horror

III. A Search and an Evocation

IV. A Mutation and a Madness

V. A Nightmare and a Cataclysm

After that, you will write a paper looking at the importance of place in the novella.

Well that looks stupid...

```
HTML Tidy x
3 </p>
4 <div class="toc">
5   <p>
6     I. A Result and a Prologue
7   </p>
8   <p>
9     II. An Antecedent and a
10    Horror
11  </p>
12  <p>
13    III. A Search and an
14    Evocation
15  </p>
16  <p>
17    IV. A Mutation and a Madness
18  </p>
19  <p>
20    V. A Nightmare and a
21    Cataclysm
22  </p>
23 </div>
24 <p>
```

```
CSS Tidy View Compiled x
JS Tidy x
```

Your assignment is to read H. P. Lovecraft's *The Case of Charles Dexter Ward*, a novella in five parts.

- I. A Result and a Prologue
- II. An Antecedent and a Horror
- III. A Search and an Evocation
- IV. A Mutation and a Madness
- V. A Nightmare and a Cataclysm

After that, you will write a paper looking at the importance of place in the novella.

```
HTML
1 <p>
2   Your assignment is to read H.
   P. Lovecraft's <i>The Case of
   Charles Dexter Ward</i>, a
   novella in five parts.
3 </p>
4 <div class="toc">
5   <p>
6     I. A Result and a Prologue
7   </p>
8   <p>
9     II. An Antecedent and a
       Horror
10  </p>

```

```
CSS
1 .toc {
2   background-color: #E8E8E8;
3   margin: 1em;
4   border: 1px solid #ccc;
5   padding: 0 1em;
6   width: 90%;
7   border-radius: 1em;
8 }

```

```
JS

```

Your assignment is to read H. P. Lovecraft's *The Case of Charles Dexter Ward*, a novella in five parts.

- I. A Result and a Prologue
- II. An Antecedent and a Horror
- III. A Search and an Evocation
- IV. A Mutation and a Madness
- V. A Nightmare and a Cataclysm

After that, you will write a paper looking at the importance of place in the novella.

Much better—& that is why we have
<div>

```
HTML
4 <ol class="toc">
5   <li>A Result and a Prologue</li>
6   <li>An Antecedent and a Horror</li>
7   <li>A Search and an Evocation</li>
8   <li>A Mutation and a Madness</li>
9   <li>A Nightmare and a Cataclysm</li>
10 </ol>
11 <p>
12   After that, you will write a paper looking at
13   the importance of place in the novella.
14 </p>
```

```
CSS
1 .toc {
2   background-color: #E8E8E8;
3   margin: 1em auto;
4   border: 1px solid #ccc;
5   padding: 1em;
6   width: 80vw;
7   border-radius: 1em;
8   list-style-type: upper-roman;
9 }
10
11 li {
12   margin-left: 1em;
13 }
```

```
JS
```

Your assignment is to read H. P. Lovecraft's *The Case of Charles Dexter Ward*, a novella in five parts.

- I. A Result and a Prologue
- II. An Antecedent and a Horror
- III. A Search and an Evocation
- IV. A Mutation and a Madness
- V. A Nightmare and a Cataclysm

After that, you will write a paper looking at the importance of place in the novella.

Actually, to be semantic, I should really use an ``

This is a useless `<div>`

```
<div class="lead-copy">
```

```
<p>
```

When a traveller in north central Massachusetts takes the wrong fork at the junction of the Aylesbury pike just beyond Dean's Corners he comes upon a lonely and curious country.

```
</p>
```

```
</div>
```

Only use `<div>` around 2 or more elements that create block boxes*

* It's OK to wrap a `<div>` around 1 element in a few cases

```
HTML
1 <div>
2   <p>Then, at last, the truth dawned
  upon me. The curvilinear patterns on
  many of those blocks were closely
  related—parts of one vast decorative
  conception. For the first time in this
  aeon-shaken waste I had come upon a
  mass of masonry in its old position—
  tumbled and fragmentary, it is true,
  but none the less existing in a very
  definite sense.</p>
3 </div>
```

```
* CSS (SCSS)
1 div,
2 p {
3   border: 2px solid dodgerblue;
4 }
5
```

```
JS
```

Then, at last, the truth dawned upon me. The curvilinear patterns on many of those blocks were closely related—parts of one vast decorative conception. For the first time in this aeon-shaken waste I had come upon a mass of masonry in its old position—tumbled and fragmentary, it is true, but none the less existing in a very definite sense.

One reason to use a `<div>` around a single element

Inheritance

⚙ HTML



```
1 <p>
2   And they called him <span class="alert">Seamus</span>...
3 </p>
```

⚙ CSS



```
1 html {
2   font-size: 24px; /* inherited */
3 }
4
5 p {
6   font-family: Georgia, serif; /* inherited */
7 }
8
9 .alert {
10  color: red;
11 }
```

⚙ JS



And they called him **Seamus**...

Why is **Seamus** red?

Why is **Seamus** using the
Georgia font?

Why is **Seamus** 24px?

Some properties, like `font-size` & `color`, are *inherited*: elements with those properties pass those properties down through the DOM to their descendant elements (unless overridden)

Other properties, like `background-image` & `border`, are *not inherited*: elements with those properties do not pass those properties down to their descendent elements

Inheritance is for elements that do not have properties set

⚙ HTML



```
1 <blockquote>
2   <p>
3     Of the five remaining pictures, three were of
      swamp and hill scenes which seemed to bear
      traces of hidden and <em>unwholesome</em>
      tenancy. Another was of a queer mark in the
      ground very near Akeley's house...
4   </p>
5 </blockquote>
```

Of the five remaining pictures, three were of swamp and hill scenes which seemed to bear traces of hidden and *unwholesome* tenancy. Another was of a queer mark in the ground very near Akeley's house...

⚙ CSS



```
1 html {
2   font-size: 24px; /* inherited */
3 }
4
5 blockquote {
6   font-family: "Iowan Old Style", Georgia,
  serif; /* inherited */
7   border: 1px solid black; /* not inherited */
8   padding: 0 1em; /* not inherited */
9 }
10
```

Notice what is inherited in CSS & what is not

It would be pretty ugly if **border** or **padding** was inherited!

⚙ JS



border-collapse	font-style	line-height
border-spacing	font-variant	orphans
caption-side	font	quotes
color	letter-spacing	text-align
cursor	list-style-type	text-edge
direction	list-style-position	text-indent
empty-cells	list-style-image	text-transform
font-family	list-style	visibility
font-size		white-space
font-weight		widows
		word-spacing

Partial list of inherited properties

Cascading Style Sheets

How does the rendering engine know which style to apply to an element?

If an element *does not have properties* set: inheritance
(if the property is inheritable)

If an element *matches* a selector: use that selector's
styles

If an element's CSS rules *conflict* (e.g., CSS tells the
rendering engine to make all `<p>`s black *and* blue): ?



DESIGN FOR THE NEW
FARMERS & MECHANICS BANK OF LA GRANGE

Raising Arizona, 1987



DESIGN FOR THE NEW
FARMERS & MECHANICS BANK OF LA GRANGE

If an element's CSS declaration conflicts with another declaration, the rendering engine uses the *Cascade* to find a winner

In other words, conflicting declarations follow a *cascade*, & the rule with the most *weight* wins

Origin

browser
user
author
author !important
user !important
browser !important

Conflicting CSS declarations follow the cascade until the one with the most weight wins

tied

Inline

style=" "

absent

Specificity

universal | :where()
class | :pseudo-class | [attribute]
type | ::pseudo-element
#ID

tied








Order

first
next
last

CSS Cascade (basic)

Origin
(+ !important)

CSS can originate from 3 places:

- » Browser, aka the *user agent*    
- » User, the person utilizing the user agent  
- » Author, i.e., a developer 

All Web browsers have built-in CSS rules

In Firefox, for example, why does `<p>` have a certain amount of space before & after it?

Because of Firefox's built-in default CSS

[summary](#) | [shortlog](#) | [changelog](#) | [pushlog](#) | [graph](#) | [tags](#) | [bookmarks](#) | [branches](#) | [files](#) | [changeset](#) | file | [latest](#) | [revisions](#) | [annotate](#) | [diff](#) | [comparison](#) | [raw](#) | [help](#)

layout/style/html.css

authorffxbld

Sat, 26 Dec 2015 04:05:31 -0800

changeset 277655c39a71cf4146

parent 275624ddf13e6aa97e

permissions-rw-r--r--



No bug, Automated blocklist update from host bld-linux64-spot-593 - a=blocklist-update

```
1  /* This Source Code Form is subject to the terms of the Mozilla Public
2   * License, v. 2.0. If a copy of the MPL was not distributed with this
3   * file, You can obtain one at http://mozilla.org/MPL/2.0/. */
4
5  @namespace url(http://www.w3.org/1999/xhtml); /* set default namespace to HTML */
6  @namespace xul url(http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul);
7
8  /* bidi */
9
10 [dir="rtl"] {
11     direction: rtl;
12     unicode-bidi: embed;
13 }
14 [dir="ltr"] {
15     direction: ltr;
16     unicode-bidi: embed;
17 }
18
19 bdi:-moz-dir(ltr), [dir="auto"]:-moz-dir(ltr) { direction: ltr; }
20 bdi:-moz-dir rtl), [dir="auto"]:-moz-dir rtl) { direction: rtl; }
21
22 /* To ensure http://www.w3.org/TR/REC-html40/struct/dirlang.html#style-bidi:
23  *
24  * "When a block element that does not have a dir attribute is transformed to
25  * the style of an inline element by a style sheet, the resulting presentation
26  * should be equivalent, in terms of bidirectional formatting, to the
27  * formatting obtained by explicitly adding a dir attribute (assigned the
28  * inherited value) to the transformed element."
29  *
```



```
109 hgroup,
110 html,
111 main,
112 nav,
113 section {
114     display: block;
115 }
116
117 body {
118     display: block;
119     margin: 8px;
120 }
121
122 p, dl, multicol {
123     display: block;
124     margin-block-start: 1em;
125     margin-block-end: 1em;
126 }
127
128 dd {
129     display: block;
130     -moz-margin-start: 40px;
131 }
132
133 blockquote, figure {
134     display: block;
135     margin-block-start: 1em;
136     margin-block-end: 1em;
137     -moz-margin-start: 40px;
138     -moz-margin-end: 40px;
139 }
140
141 address {
142     display: block;
```

Equivalent to `margin-top: 1em`
& `margin-bottom: 1em`



WebKit (Safari)

Search

[Register](#) | [Lost Password](#) | [Login](#)

	Wiki	Timeline	Browse Source	Search
--	------	----------	---------------	--------

[← Previous Revision](#) | [Next Revision →](#) | [Blame](#) | [Revision Log](#)

source: [trunk](#) / [Source](#) / [WebCore](#) / [css](#) / [html.css](#) @ 194427

Visit:

View revision:

Revision **191751**, 30.6 KB checked in by [wenson_hsieh@apple.com](#), 2 months ago ([diff](#))

Date input values should not overlap with menu list dropdown button on iOS

➡ https://bugs.webkit.org/show_bug.cgi?id=150681

<rdar://problem/19965078>

Reviewed by Zalan Bujtas.

Source/WebCore:

Adds a right margin on iOS date and time inputs so that the inner div does not render the value of the date on top of the dropdown button on the right of the menu list.

Tests: fast/forms/date/date-input-rendering-basic.html

fast/forms/time/time-input-rendering-basic.html

- css/html.css:
(input::-webkit-date-and-time-value):
LayoutTests:
Tests that date and time inputs render properly. Namely, the text should not be rendered such that it covers the dropdown button on the right of the menu list.

- fast/forms/date/date-input-rendering-basic-expected.txt: Added.
- fast/forms/date/date-input-rendering-basic.html: Added.
- fast/forms/time/time-input-rendering-basic-expected.txt: Added.
- fast/forms/time/time-input-rendering-basic.html: Added.

Property **svn:eol-style** set to *native*

Line	
1	/*
2	* The default style sheet used to render HTML.
3	*
4	* Copyright (C) 2000 Lars Knoll (knoll@kde.org)
5	* Copyright (C) 2003-2011, 2014 Apple Inc. All rights reserved.
6	*
7	* This library is free software; you can redistribute it and/or
8	* modify it under the terms of the GNU Library General Public
9	* License as published by the Free Software Foundation; either
10	* version 2 of the License, or (at your option) any later version.

Safari's built-in CSS

```

26 html {
27     display: block;
28 }
29
30 /* children of the <head> element all have display:none */
31 head, link, meta, script, style, title {
32     display: none;
33 }
34
35 /* generic block-level elements */
36
37 body {
38     display: block;
39     margin: 8px;
40 }
41
42 p {
43     display: block;
44     -webkit-margin-before: 1__qem;
45     -webkit-margin-after: 1__qem;
46     -webkit-margin-start: 0;
47     -webkit-margin-end: 0;
48 }
49
50 address, article, aside, div, footer, header, hgroup, layer, main, nav, section {
51     display: block;
52 }
53
54 marquee {
55     display: inline-block;
56     overflow: -webkit-marquee;
57 }
58
59 blockquote {

```

Equivalent to `margin-top: 1em`
& `margin-bottom: 1em`

HTML

Living Standard — Last Updated 24 December 2015



[← 13 The XHTML syntax](#) — [Table of Contents](#) — [15 Obsolete features](#) →

14 Rendering

14.1 Introduction

14.2 The CSS user agent style sheet and presentational hints

14.3 Non-replaced elements

14.3.1 Hidden elements

14.3.2 The page

14.3.3 Flow content

14.3.4 Phrasing content

14.3.5 Bidirectional text

14.3.6 Quotes

14.3.7 Sections and headings

14.3.8 Lists

14.3.9 Tables

14.3.10 Margin collapsing quirks

14.3.11 Form controls

14.3.12 The `hr` element

14.3.13 The `fieldset` and `legend` elements

WHATWG's standard

[File an issue about the selected text](#)

HTML

Living Standard — Last Updated 24 December 2015



← 13 The XHTML syntax — Table of Contents — 15 Obsolete features →

14 Rendering

14.1 Introduction

“The CSS rules given in these subsections are, except where otherwise specified, expected to be used as part of the user-agent level style sheet defaults for all documents that contain HTML elements.”

14.2 The CSS user-agent style sheet and presentational hints

14.3 Non-replaced elements

14.3.1 Hidden elements

14.3.2 The page

14.3.3 Flow content

14.3.4 Formatting content

14.3.5 Bidirectional text

14.3.6 Quotes

14.3.7 Sections and headings

14.3.8 Lists

14.3.9 Tables

14.3.10 Margin collapsing quirks

14.3.11 Form controls

14.3.12 The `hr` element

14.3.13 The `fieldset` and `legend` elements

WHATWG's standard

[File an issue about the selected text](#)

14.3.3 Flow content

Spec bugs: [27673](#)

```
CSS @namespace url(http://www.w3.org/1999/xhtml);
```

```
address, blockquote, center, div, figure, figcaption, footer, form, header, hr,  
legend, listing, main, p, plaintext, pre, summary, xmp {  
    display: block;  
}
```

```
blockquote, figure, listing, p, plaintext, pre, xmp {  
    margin-top: 1em; margin-bottom: 1em;  
}
```

```
blockquote, figure { margin-left: 40px; margin-right: 40px; }
```


```
address { font-style: italic; }  
listing, plaintext, pre, xmp {  
    font-family: monospace; white-space: pre;  
}
```

```
dialog:not([open]) { display: none; }
```

```
dialog {  
    position: absolute;  
    left: 0; right: 0;  
    width: fit-content;  
    height: fit-content;  
    margin: auto;  
    border: solid;
```

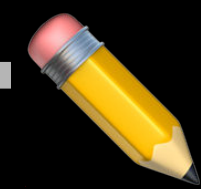
Users can specify CSS rule sets too

Why? Accessibility & other reasons

- » A more readable font
- » All fonts are at least a certain size
- » Always enable `text-decoration: underline` so links are obvious
- » See outlines around elements with keyboard focus using `outline: solid`

Firefox has always allowed users to create their own styles in a file called `userContent.css` that goes into your Firefox Profile

(Look in the slides' Notes for more info about your Profile & `userContent.css`)

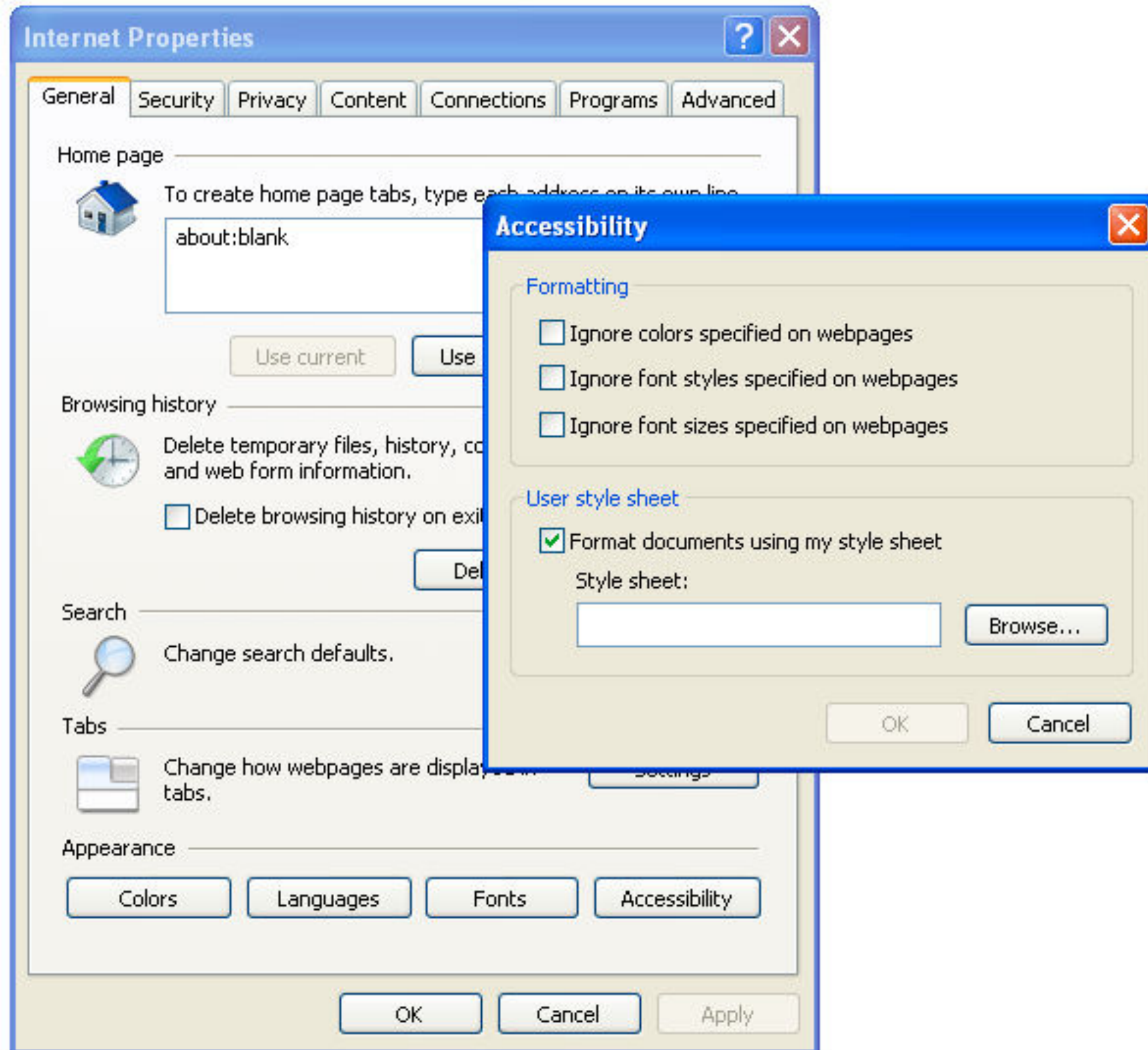


SIDE NOTE

As of Firefox 69, `userContent.css` isn't supported by default unless users first enable the feature

1. Type `about:config` in the Firefox address bar & select Enter
2. Click the button that confirms you Accept the Risk and Continue
3. In the box at the top, search for `toolkit.legacyUserProfileCustomizations.stylesheets`
4. Double-click on the resulting line to toggle to `true`
5. Restart Firefox

Internet Explorer actually had good support for user CSS





Chrome 33 (2014) dropped support for a user styles file



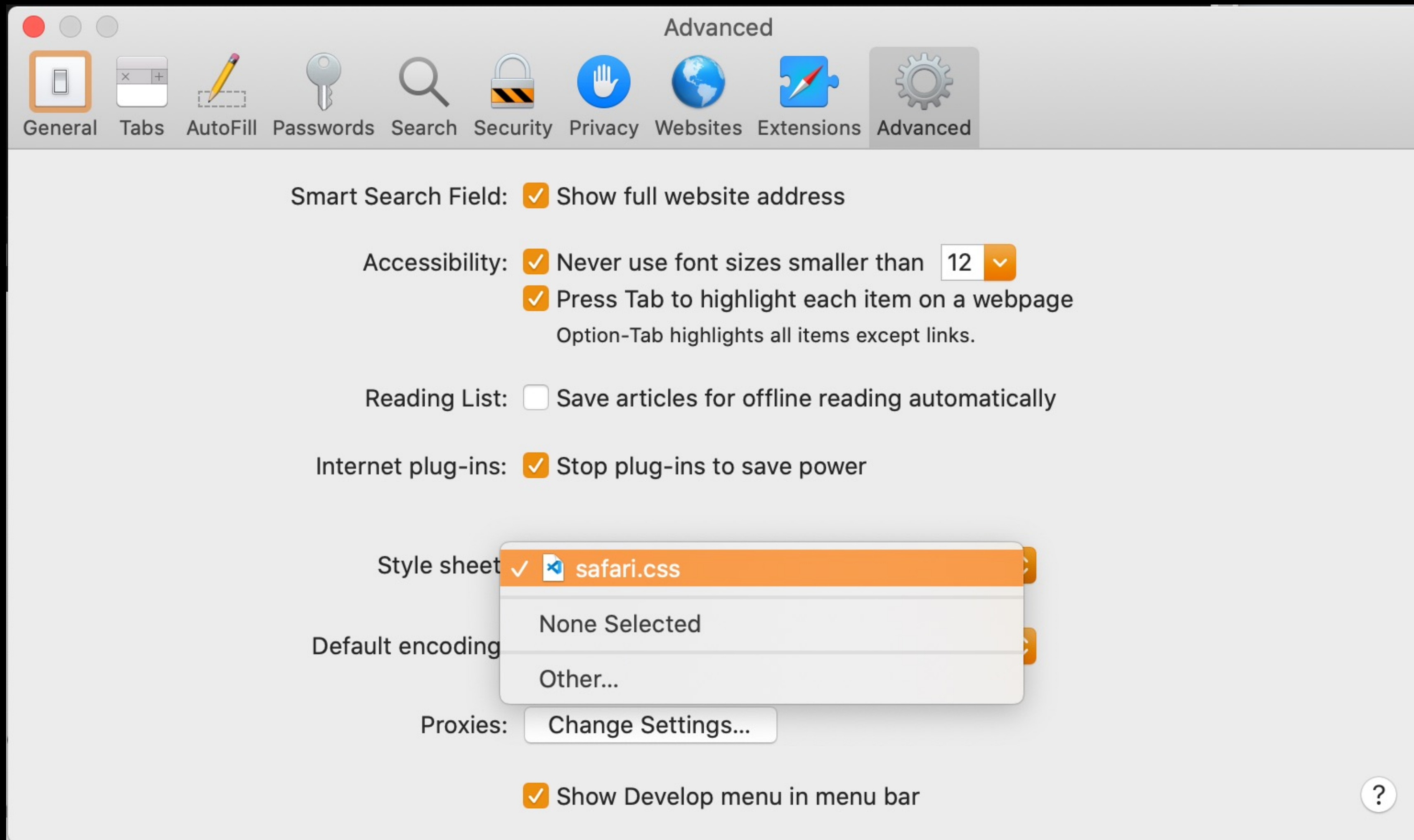
Chrome 33 (2014) dropped support for a user styles file



Edge: *Never supported* a user styles file!



Edge: *Never supported* a user styles file!



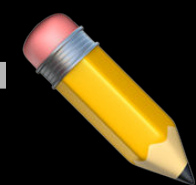
Safari — I created `safari.css` for my own use

My safari.css file

```
html {  
    font-family: "Source Sans Pro", sans-serif;  
}
```

```
h1, h2, h3, h4, h5, h6 {  
    font-family: "Georgia Pro", serif;  
}
```

```
code, kbd, pre, samp, tt, var {  
    font-family: "Source Code Pro", monospace;  
}
```



SIDE NOTE

Browser makers' response when asked about missing support for a user styles file: "Get an extension"

Do get:

- » Stylus for Chromium-based browsers & Firefox (which is excellent & removes all analytics, telemetry, & data collection)
- » Cascadea for Safari

Do *not* get: Stylish 🚫 (was good, but now it's spyware)

Default, without Stylus

Page issues

Article **Talk**

文A Language

★ Watch

 History

 [Edit](#)

More

Not to be confused with the psychotherapy of Fritz Perls, [Gestalt therapy](#).

Gestalt psychology, gestaltism or configurationism^[1] is a [school of psychology](#) that emerged in the early twentieth century in Austria and Germany as a theory of [perception](#) that was a rejection of basic principles of [Wilhelm Wundt](#)'s and [Edward Titchener](#)'s elementalist and [structuralist psychology](#).^{[2][3][4]}

As used in Gestalt psychology, the German word *Gestalt* (/gəˈʃtælt, -ˈʃtɑːlt, -ˈʃtɔːlt, -ˈstaːlt, -ˈstɔːlt/ gə-SHTA(H)LT, -STAHLT, -S(H)TAWLT,^{[5][6]} German: [gəˈʃtalt] ( listen); meaning "form"^[7]) is interpreted as "pattern" or "configuration".^[8] Gestalt psychologists emphasized that organisms perceive entire patterns or configurations, not merely individual components.^[8] The view is sometimes summarized using the adage, "the whole is more than the sum of its parts."^[9]:13 Gestalt principles, proximity, similarity, figure-ground, continuity, closure, and connection, describe how humans perceive visuals in connection

Part of a series on

Psychology.

$$\Psi$$

[Outline](#) · [History](#) · [Subfields](#)

Basic types

Abnormal • Behavioral • Behavioral genetics •
Biological • Cognitive/Cognitivism •
Comparative • Cross-cultural • Cultural •

Page issues

Article Talk

文A Language

★ Watch

 History [Edit](#)

⋮ More

Not to be confused with the psychotherapy of Fritz Perls, [Gestalt therapy](#).

Gestalt psychology, gestaltism or configurationism^[1] is a [school of psychology](#) that emerged in the early twentieth century in Austria and Germany as a theory of [perception](#) that was a rejection of basic principles of [Wilhelm Wundt](#)'s and [Edward Titchener](#)'s elementalist and [structuralist psychology](#).^{[2][3][4]}

As used in Gestalt psychology, the German word *Gestalt* (/gəˈʃtælt, -ˈʃtɑːlt, -ˈʃtɔːlt, -ˈstɑːlt, -ˈstɔːlt/ gə-SHTA(H)LT, -STAHLT, -S(H)TAWLT,^{[5][6]} German: [gəˈʃtalt] (🔊 listen); meaning "form"^[7]) is interpreted as "pattern" or "configuration".^[8] Gestalt psychologists emphasized that organisms perceive entire patterns or configurations, not merely individual components.^[8] The view is sometimes summarized using the adage, "the whole is more than the sum of its parts."^[9]:13 Gestalt principles, proximity, similarity, figure-ground, continuity, closure, and connection, describe how humans perceive visuals in connection

Part of a series on

Psychology

$$\Psi$$

[Outline](#) · [History](#) · [Subfields](#)

Basic types

Abnormal • Behavioral • Behavioral genetics •
Biological • Cognitive/Cognitivism •
Comparative • Cross-cultural • Cultural •

Gestalt psychology

Page issues

Article **Talk**

文A Language

★ Watch

 History. Edit

More

Not to be confused with the psychotherapy of Fritz Perls, [Gestalt therapy](#).

Gestalt psychology, gestaltism or configurationism^[1] is a [school of psychology](#) that emerged in the early twentieth century in Austria and Germany as a theory of [perception](#) that was a rejection of basic principles of [Wilhelm Wundt](#)'s and [Edward Titchener](#)'s elementalist and [structuralist psychology](#).^{[2][3][4]}

As used in Gestalt psychology, the German word *Gestalt* (/gəˈʃtælt, -ˈʃtaːlt, -ˈʃtɔːlt, -ˈstɑːlt, -ˈstɔːlt/ gə-SHTA(H)LT, -STAHLT, -S(H)TAWLT,^{[5][6]} German: [gəˈʃtalt] (🔊 listen); meaning "form"^[7]) is interpreted as "pattern" or "configuration".^[8] Gestalt psychologists emphasized that organisms perceive entire patterns or configurations, not merely individual components.^[8] The

Part of a series on

Psychology.

$$\Psi$$

[Outline](#) · [History](#) · [Subfields](#)

Basic types

Abnormal • Behavioral • Behavioral genetics •

Page issues

Article **Talk**

文A Language

★ Watch

 History [Edit](#)

⋮ More

Not to be confused with the psychotherapy of Fritz Perls, [Gestalt therapy](#).

Gestalt psychology, gestaltism or configurationism^[1] is a [school of psychology](#) that emerged in the early twentieth century in Austria and Germany as a theory of [perception](#) that was a rejection of basic principles of [Wilhelm Wundt](#)'s and [Edward Titchener](#)'s elementalist and [structuralist psychology](#).^{[2][3][4]}

As used in Gestalt psychology, the German word *Gestalt* (/gəˈʃtælt, -ˈʃtaːlt, -ˈʃtɔːlt, -ˈstɑːlt, -ˈstɔːlt/ gə-SHTA(H)LT, -STAHLT, -S(H)TAWLT,^{[5][6]} German: [gəˈʃtalt] (🔊 listen); meaning "form"^[7]) is interpreted as "pattern" or "configuration".^[8] Gestalt psychologists emphasized that organisms perceive entire patterns or configurations, not merely individual components.^[8] The

Part of a series on

Psychology

$$\Psi$$

[Outline](#) · [History](#) · [Subfields](#)

Basic types

Abnormal • Behavioral • Behavioral genetics •

Installed Styles

▼Filters:

☐ Only enabled styles ▼
☐ Only locally created styles ▼
☐ Only Usercss styles ▼

Search

Metadata ▼

ⓘ

Title ▼

ⓘ

▼Actions

Check all styles for updates 📄🕒

Write new style ☐ as Usercss ↗

Options

▼Backup

Select a file or drag and drop to this page.

Export Import Sync to cloud

Get and share styles ⓘ |
Get styles ⓘ | Get help | Wiki |
Translate

<input type="checkbox"/> +All	×	4y	Everything
<input checked="" type="checkbox"/> +Granneman	×	9m	granneman.com finn.gabefinn.com wu.granneman.com
<input checked="" type="checkbox"/> +NYT	×	2y	nytimes.com nymag.com
<input checked="" type="checkbox"/> +Printing	×	2y	Everything
<input checked="" type="checkbox"/> +Wikis	×	17h	wikipedia.org wikimedia.org mediawiki.org
<input checked="" type="checkbox"/> admin.mailroute.net	×	3y	admin.mailroute.net
<input checked="" type="checkbox"/> allmusic.com	×	1y	allmusic.com allmovie.com
<input checked="" type="checkbox"/> Amazon	×	3y	amazon.com
<input checked="" type="checkbox"/> archive.org	×	5m	archive.org
<input checked="" type="checkbox"/> atariarchives.org	×	6m	atariarchives.org
<input checked="" type="checkbox"/> brado.net	×	10m	brado.net
<input checked="" type="checkbox"/> cloudform.com	×	4y	blog.cloudform.org
<input checked="" type="checkbox"/> community.folivora.ai	×	4y	community.folivora.ai
<input checked="" type="checkbox"/> corecode io	×	3y	corecode io

Beautify

Publish style

Add another section Clone Beautify

Edit Style

allmusic.com

☒ Enabled ☒ Live preview

Save Beautify Back to manage

Import Export ⓘ

Options

- ☒ Word wrap
- ☒ Use smart indentation
- ☐ Use tabs with smart indentation
- ☒ Autoclose brackets and quotes
- ☐ Autocomplete on typing
- ☒ Double-clicking selects tokens
- ☒ Colorpickers for CSS colors ⚙

Tab size

Keymap ⓘ

Theme

Highlight

CSS Linter ⚙

Publish

Using [userstyles.world](#)

Publish style

Code 1:

```
1 html {
2   font-size: 16px !important;
3 }
4
5 section.biography .text,
6 .description,
7 .review p,
8 .user_review div.middle {
9   font-family: "Source Sans 3" !important;
10  font-size: 1.2em !important;
11  line-height: 1.5 !important;
12 }
13
14 header.biography {
15   font-size: 1.2em !important;
16 }
17
18 .header-main-nav .max-header .main-nav .site-nav,
19 .header-main-nav .max-header-min .main-nav .site-nav {
20   font-size: 1rem;
21 }
22
23 tabs tab a {
```

Applies to ⓘ - +

- +

Add another section Clone Beautify

Edit Style

google.com

☒ Enabled ☒ Live preview

Save Beautify Back to manage

Import Export ⓘ

Options

- ☒ Word wrap
- ☒ Use smart indentation
- ☐ Use tabs with smart indentation
- ☒ Autoclose brackets and quotes
- ☐ Autocomplete on typing
- ☒ Double-clicking selects tokens
- ☒ Colorpickers for CSS colors ⚙

Tab size 2

Keymap default ⓘ

Theme default ▾

Highlight Token under cursor ▾

CSS Linter CSSLint ⚙

Publish

Using [userstyles.world](#)

Publish style

Code 1:

```
1 #taw {
2   display: none;
3 }
```

Applies to ⓘ URLs on the domain ▾ google.com - +

Add another section Clone Beautify

Edit Style

interaction-design.org

☒ Enabled ☒ Live preview

Save Beautify Back to manage

Import Export ⓘ

Options

- ☒ Word wrap
- ☒ Use smart indentation
- ☐ Use tabs with smart indentation
- ☒ Autoclose brackets and quotes
- ☐ Autocomplete on typing
- ☒ Double-clicking selects tokens
- ☒ Colorpickers for CSS colors ⚙

Tab size

Keymap ⓘ

Theme

Highlight

CSS Linter ⚙

Publish

Using [userstyles.world](#)

Publish style

Code 1:

```
1 .navigationHeader {
2   position: static !important;
3 }
4
5 .navigationHeader__secondaryNavigationBar,
6 .ux-share-panel-bottom,
7 .uxArticleWithSidebar__sidebar {
8   display: none !important;
9 }
10
```

Applies to ⓘ - +

Add another section Clone Beautify

Edit Style

webster.edu

☒ Enabled ☒ Live preview

Save Beautify Back to manage

Import Export ⓘ

Options

- ☒ Word wrap
- ☒ Use smart indentation
- ☐ Use tabs with smart indentation
- ☒ Autoclose brackets and quotes
- ☐ Autocomplete on typing
- ☒ Double-clicking selects tokens
- ☒ Colorpickers for CSS colors ⚙

Tab size

Keymap ⓘ

Theme

Highlight

CSS Linter ⚙

Publish

Using [userstyles.world](#)

Publish style

Code 1:

```
1 h1, h2, h3, h4, h5, h6 {
2   font-family: "Source Sans 3" !important;
3   border-bottom: 0;
4   margin-top: 1rem;
5   margin-bottom: 0;
6   padding-top: 0;
7   padding-bottom: 0;
8 }
9
10 /*
11 p {
12   margin-top: 0;
13 }
14 */
```

Applies to ⓘ URLs on the domain - +

Add another section Clone Beautify



UserStyles.world

Free and open-source, community-driven platform for sharing and browsing UserCSS userstyles, and a replacement for [UserStyles.org](#), made by the userstyles community.

[Explore userstyles](#)[Create an account](#)

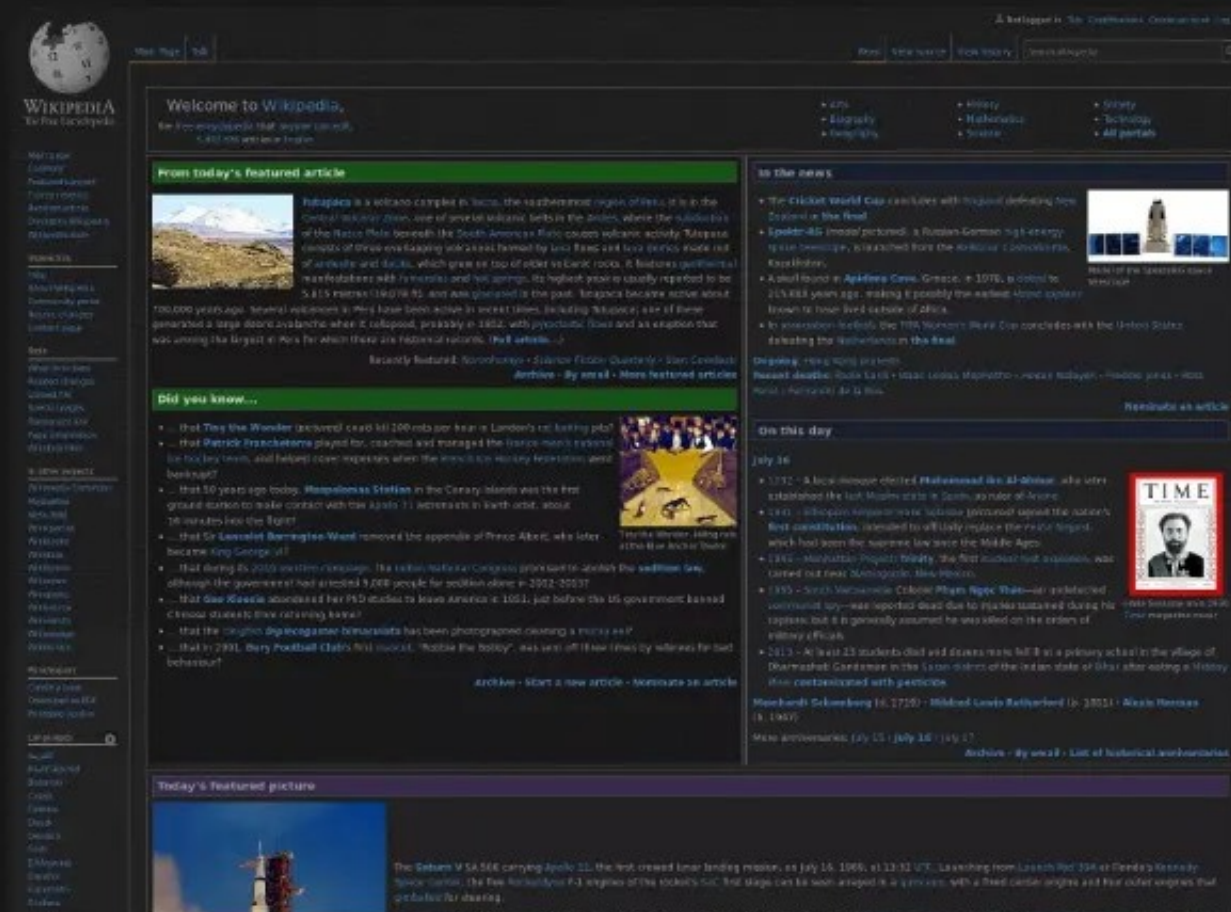
Featured userstyles

Explore

Browse all available userstyles.

Sort by Most installs

Sort



Updated December 25, 2021 00:30

Wikipedia Dark
by stylishthemes

1.6k views

186.9k installs

-/5



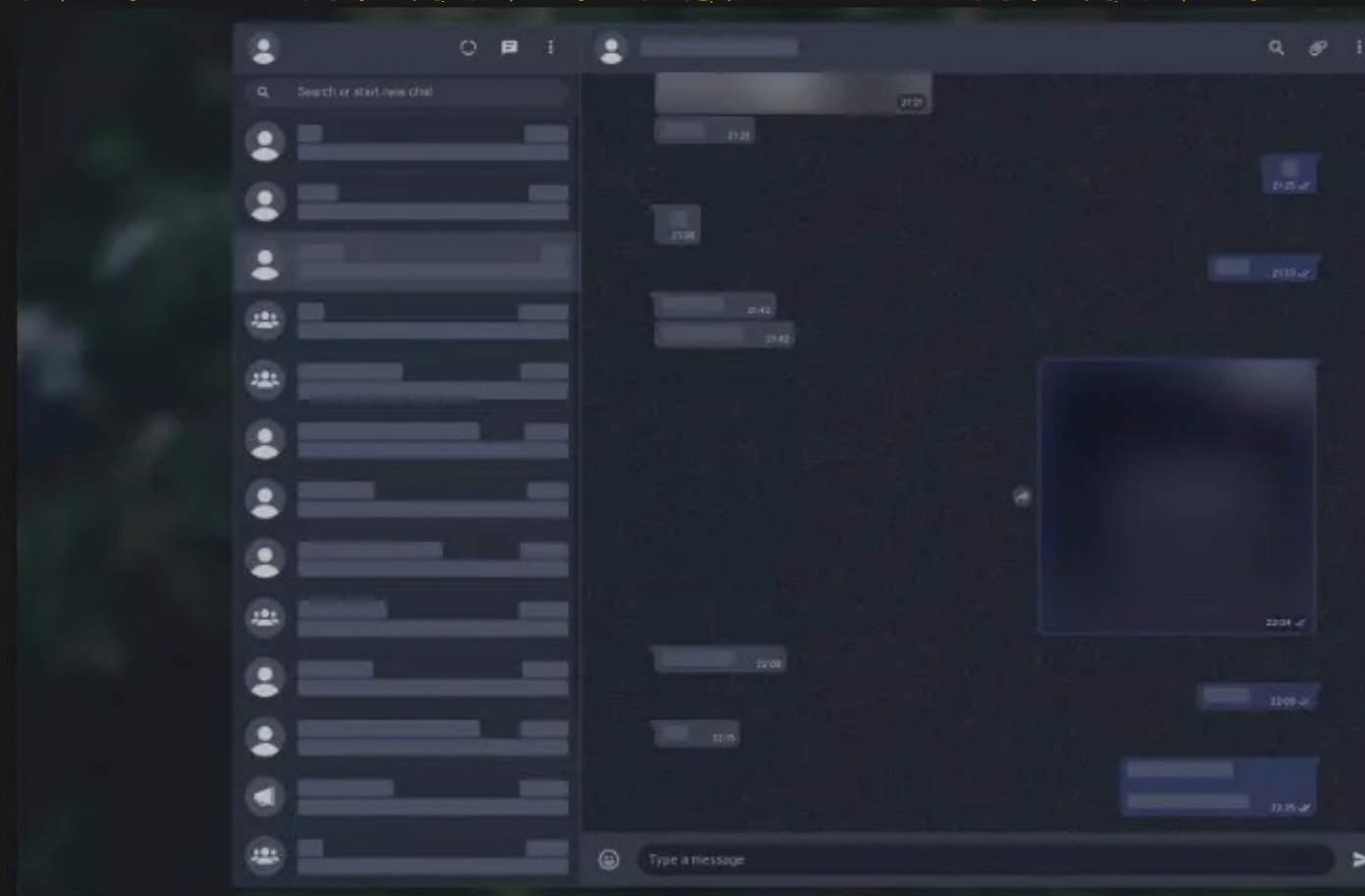
Updated July 18, 2021 00:56

Dark-WhatsApp
by vednoc

1.5k views

57.3k installs

5/5



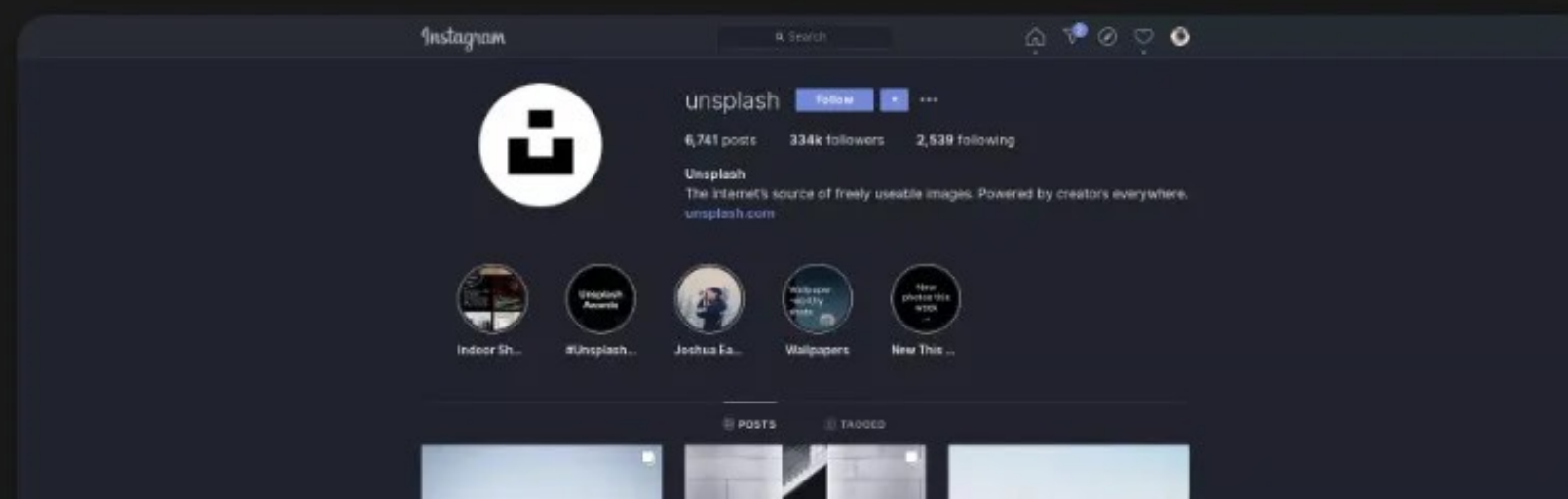
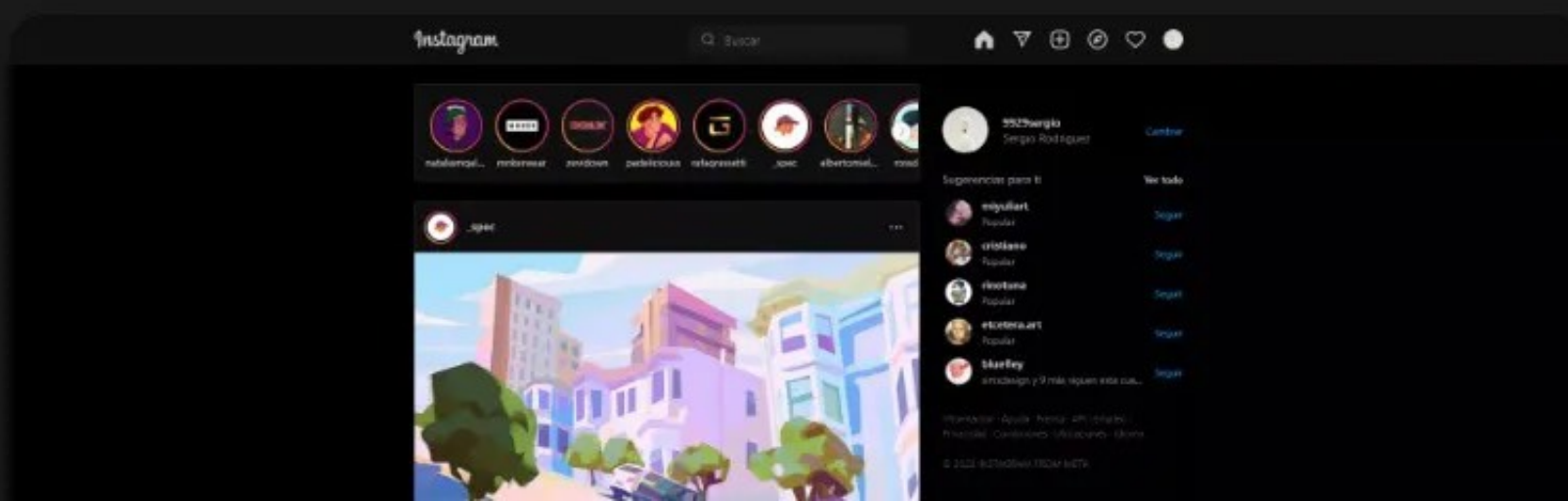
Updated April 7, 2022 06:55

Google Dark Theme
by blyad

4k views

37.2k installs

5/5



Explore

Browse all available userstyles.

Sort by Most views

Sort



Updated April 8, 2022 13:09

osu color changer
by cyperdark

4.9k views

9.2k installs

☆ -/5



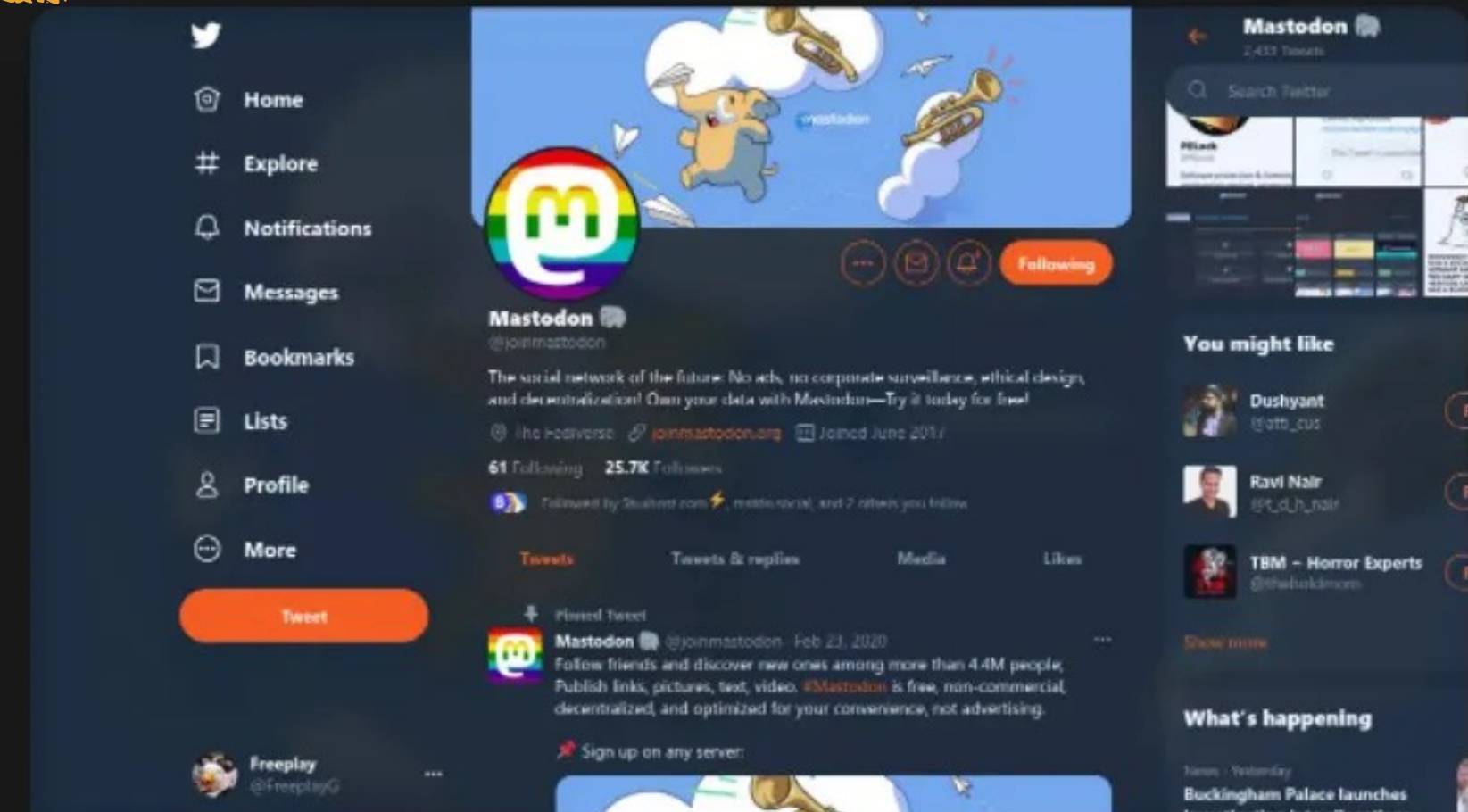
Updated April 7, 2022 06:55

Google Dark Theme
by blyad

4k views

37.2k installs

☆ 5/5



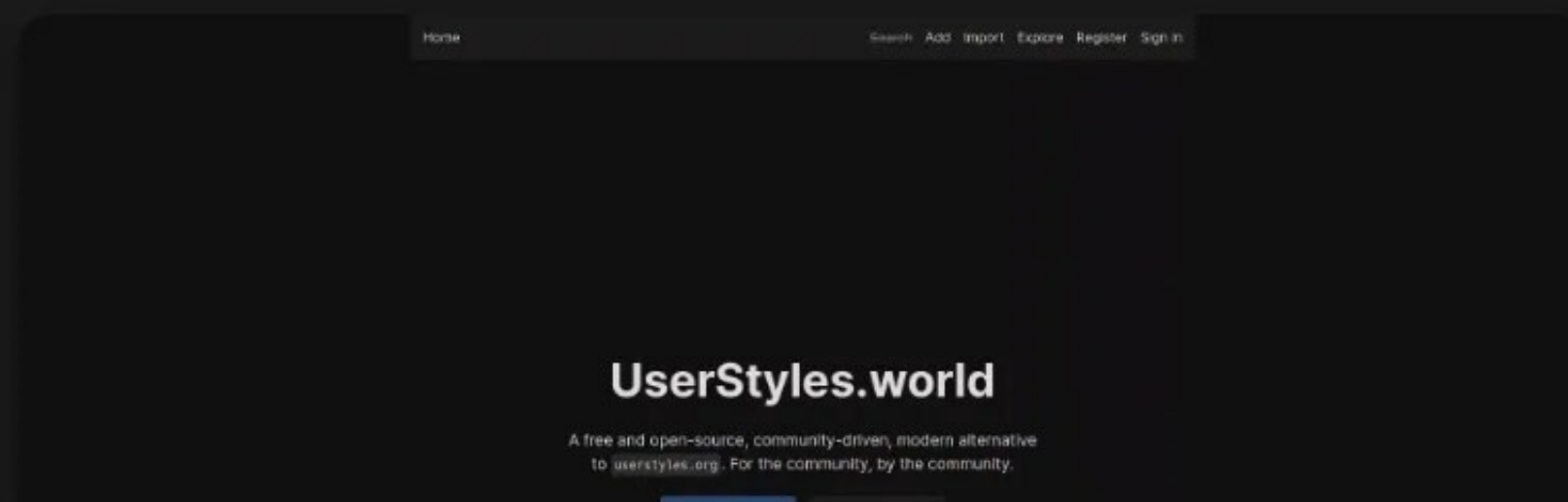
Updated March 1, 2022 23:30

Twitter Tweaks
by freeplay

3.1k views

8.1k installs

☆ 5/5



Search

Sorted from most to least relevant results.

Search keyword

duckduckgo

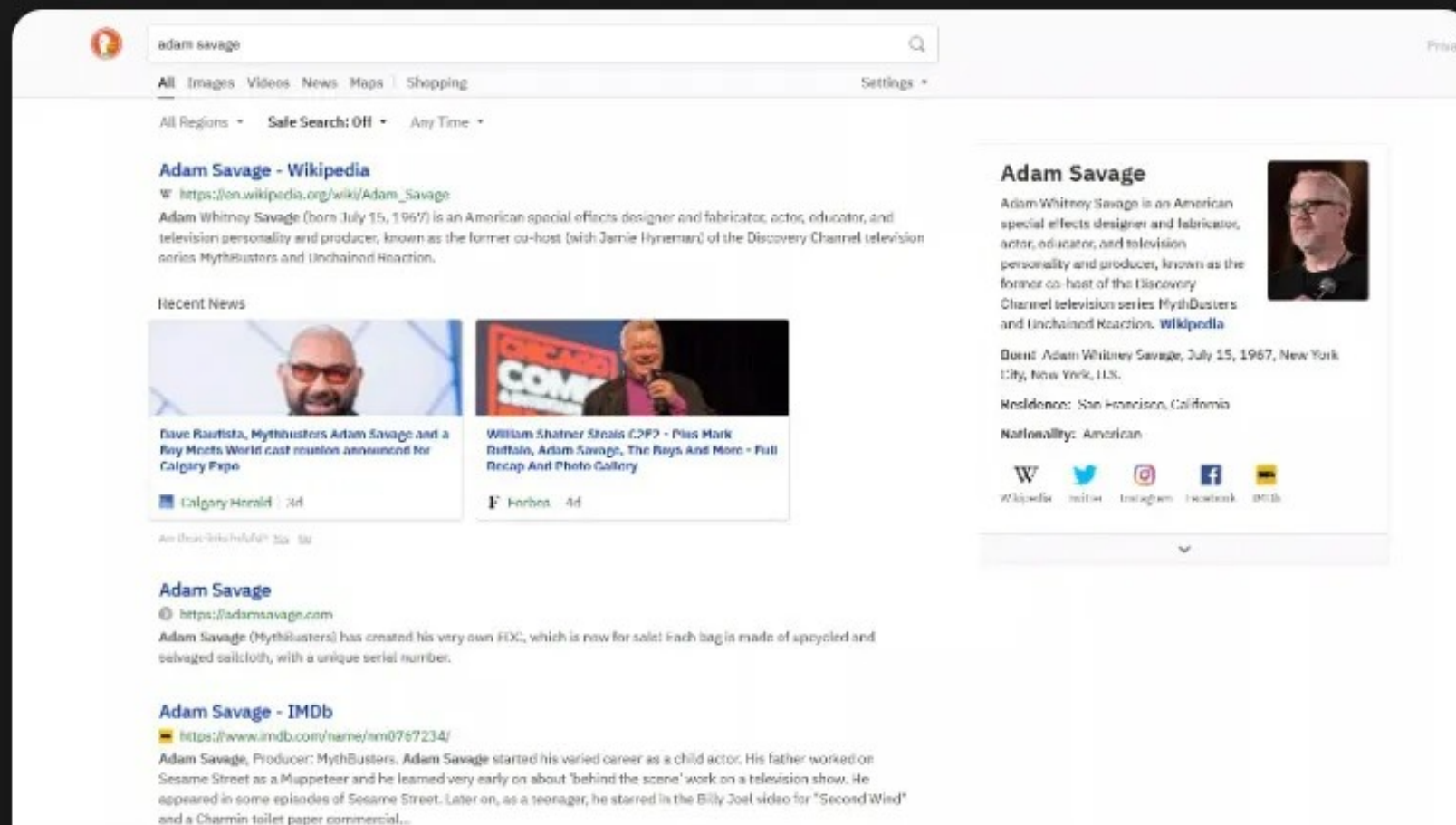
Sort by

Default



Search

Metrics of search engine: Hits: 25 · Time Spent: 2.145ms



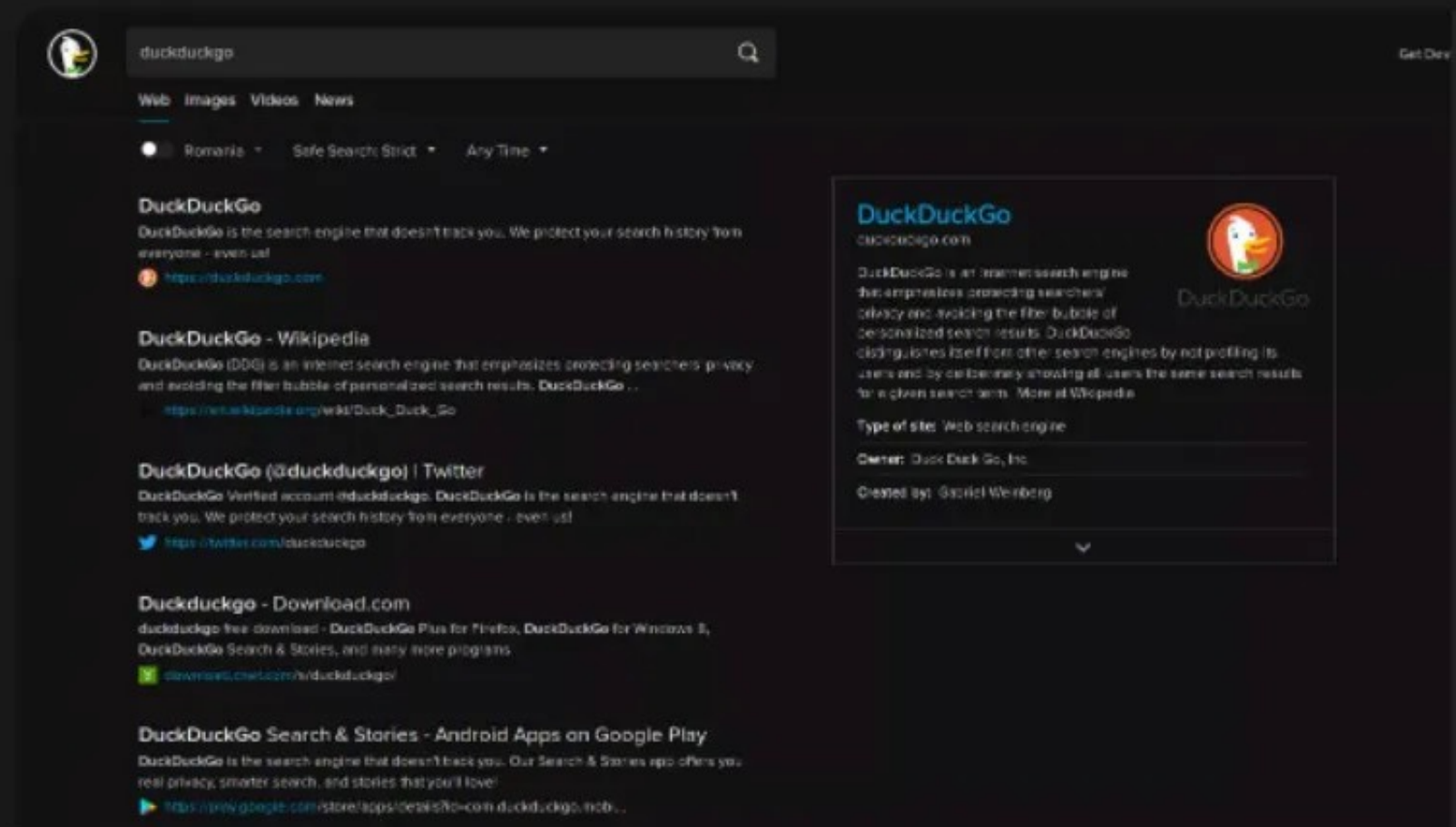
Updated January 17, 2022 19:00

DuckDuckGo - Wider, Prettier and Cu...
by krisu

388 views

1.4k installs

-/5



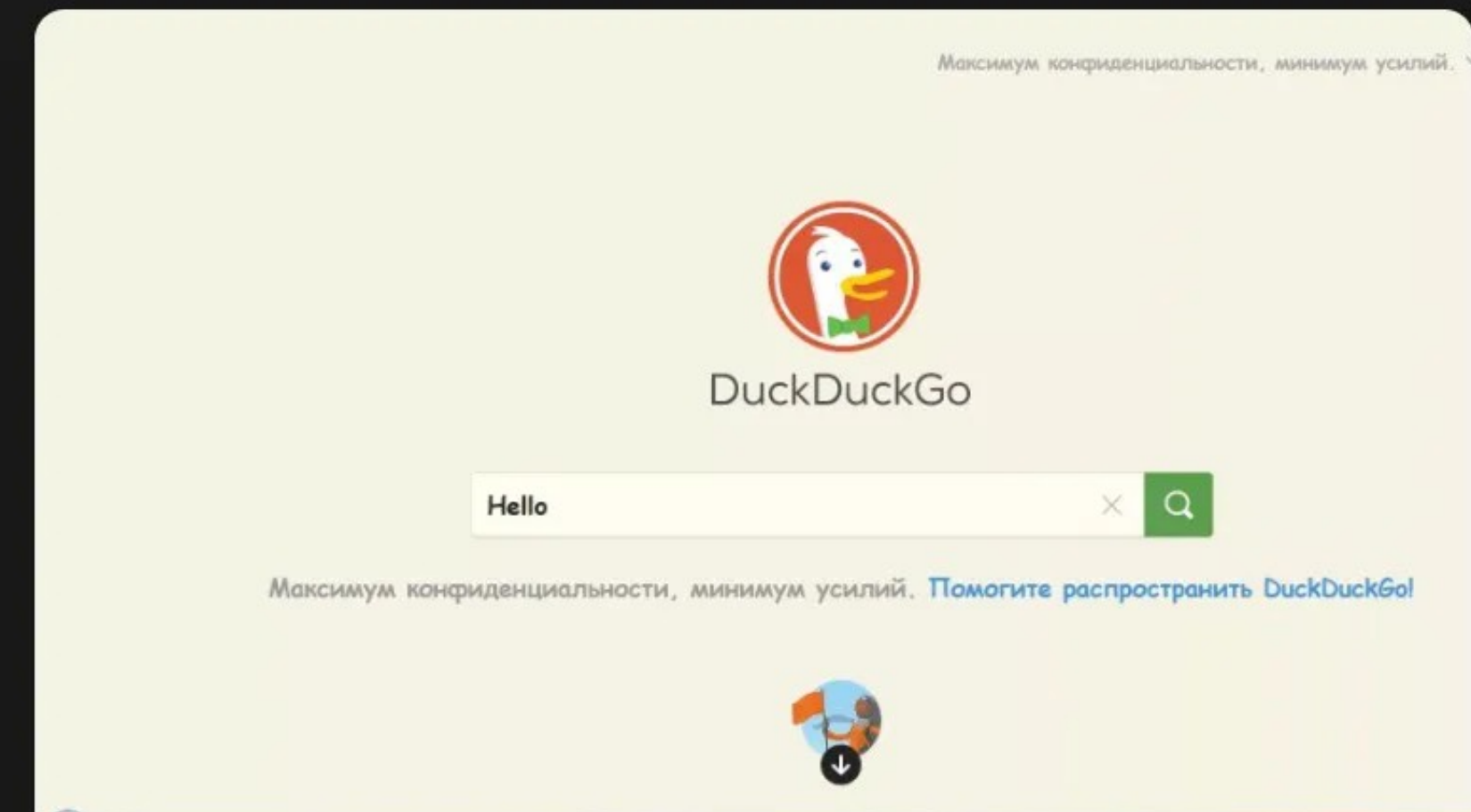
Updated September 24, 2021 11:00

DuckDuckGo DeepDark
by raitarohikami

606 views

1.2k installs

-/5



Updated April 8, 2022 16:27

Beige Duckduckgo
by odd

82 views

173 installs

-/5

Authors (developers) can specify CSS (obviously!)

If a user's CSS contradicts a author's CSS, the user can make sure hers “wins”

Use `!important` after a declaration & before the `;`

```
p {  
    font-size: 36px !important;  
}
```

However, CSS authors can do the same thing!

My safari.css file

Notice: no !important

```
html {  
    font-family: "Source Sans Pro", sans-serif;  
}
```

```
h1, h2, h3, h4, h5, h6 {  
    font-family: "Georgia Pro", serif;  
}
```

```
code, kbd, pre, samp, tt, var {  
    font-family: "Source Code Pro", monospace;  
}
```



The order of origin & importance

AKA

The order in which stylesheets are *weighted*, from lightest to heaviest

Order of
origin &
importance,
from lightest
to heaviest

Browser



User



Author



Author *!important*



User *!important*



Browser *!important*

Order of
origin &
importance,
from lightest
to heaviest

Browser



User



Author



Author !important



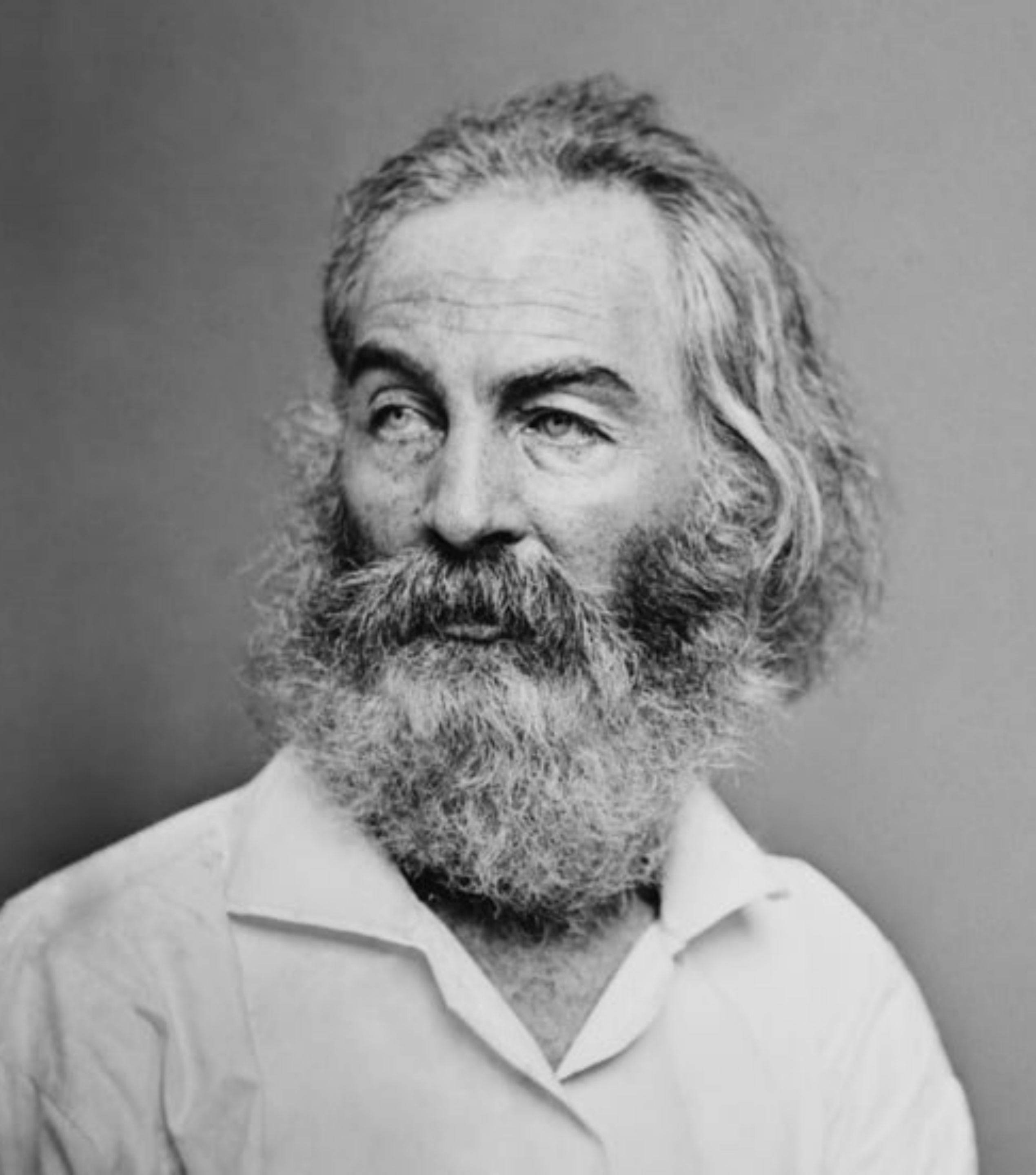
User !important



Browser !important

Author & Author !important?!

Why are you contradicting yourself?



Do I contradict myself?
Very well then I contradict
myself,
(I am large, I contain
multitudes.)

From Walt Whitman's
"Song of Myself" in *Leaves
of Grass*

You're not contradicting *yourself* (hopefully!)

What if you're using Bootstrap & its default CSS?

```
<link rel="stylesheet" href="bootstrap.css">
```

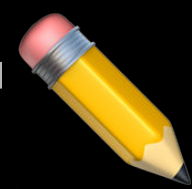
Is that CSS coming from the browser, user, or author?

You will want to override some of Bootstrap's choices

```
<link rel="stylesheet" href="bootstrap.css">  
<link rel="stylesheet" href="me.css">
```

Author vs. author

Sometimes, you will have to use `!important` to beat the *other* author

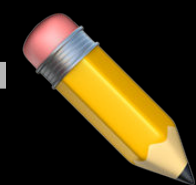


SIDE NOTE

The browser can use `!important`, & it outweighs everyone else

Fortunately, browsers only use `!important` on very few declarations

For instance, I looked through Firefox's default CSS & found only 11 uses of `!important`



SIDE NOTE


```
audio:not([controls]) {  
  display: none !important;  
}
```

This would match `<audio src="cthulhu.mp3">`

The CSS means that if `<audio>` does not have the `controls` attribute, do not display the `<audio>` (which makes sense — if there are no controls, how else are you going to view or interact with it?)

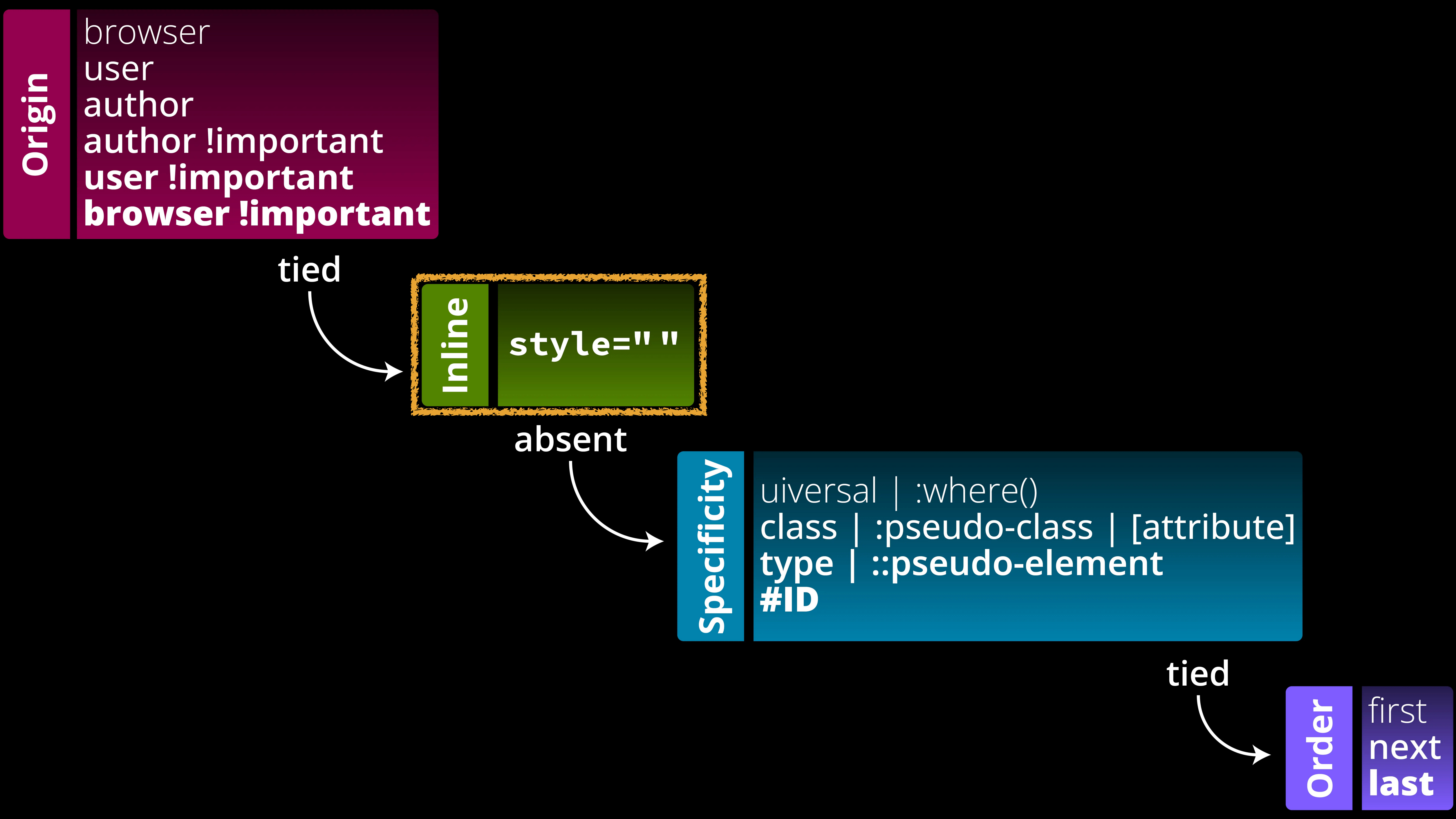
Be careful using `!important`

“`!important`: 3 seconds to type, 3 years to remove.”
—Harry Roberts

“`!important` is the Hammer of Thor, smiting everything in its path.” —Jans Carton 

Only use it as a last resort

Inline
(style="...")



Remember at the beginning how we talked about inline styles (`<p style="font-size: 18px;">`) & we gave several reasons why you should avoid them?

Here's another reason: they hamper the Cascade

In the Cascade, inline styles are always assigned to the *author* in terms of Origin

And in terms of Specificity, coming up next, inline styles outweigh anything else

So inline styles really short circuit the Cascade — yet another great reason to avoid them

Specificity

Origin

browser
user
author
author !important
user !important
browser !important

tied

Inline

style=" "

absent

Specificity

universal | :where()
class | :pseudo-class | [attribute]
type | ::pseudo-element
#ID

tied

Order

first
next
last

Specificity asks how *specific* is each selector?

Each simple selector is assigned a weight value

The greater the weight, the more specific the selector

In other words, with great weight comes great specificity 🕷️ 🤪

Selector(s)		Column
1 ×	#ID	a
1 ×	.class, :pseudo-class, & [attribute]	b
1 ×	type & ::pseudo-element	c

Take the numbers & place them **a,b,c**, e.g., **0,2,1**

	a	b	c	Total
*				0,0,0
li			1	0,0,1
.foo		1		0,1,0
#chapter1	1			1,0,0

Remember, compound & complex selectors are made up of multiple simple selectors

Simply add their simple selector weight values together

`p.foo.bar` (compound) is made up of 1 type & 2 classes, so `0,2,1`

`.foo > cite` (complex) is composed of 1 class & 1 type, so `0,1,1`

The following are *ignored* when calculating specificity:

- » Combinators: `└`, `>`, `+`, `~`
- » Universal selector: `*`
- » `:where()` & its contents
- » `:is()`, `:not()`, or `:has()` (but not the contents — the most specific selector is counted!): `:not(.but-this-part-is-counted)`

```
HTML
1 <ul>
2   <li>Great Old Ones
3     <ul>
4       <li>Abholos</li>
5       <li>Alala</li>
6       <li>Ammutseba</li>
7       <li>Amon-Gorloth</li>
8       <li>Aphoom-Zhah</li>
9       <li>Apocolothoth</li>
10      <li>Arwassa</li>
11      <li>Atlach-Nacha</li>
12      <li>Ayi'ig</li>
13      <li>Aylith</li>
14      <li>Baoh't Z'uqqa-Mogg</li>
15      <li>Basatan</li>
16      <li>B'gnu-Thun</li>
17      <li>Bokrug</li>

CSS
2
3 * ul > li { /* Specificity = 0,0,2 */
4   color: blue;
5 }
6
7 ul li { /* Specificity = 0,0,2 */
8   color: red;
9 }
10

JS
```

- Great Old Ones
 - Abholos
 - Alala
 - Ammutseba
 - Amon-Gorloth
 - Aphoom-Zhah
 - Apocolothoth
 - Arwassa
 - Atlach-Nacha
 - Ayi'ig
 - Aylith
 - Baoh't Z'uqqa-Mogg
 - Basatan
 - B'gnu-Thun
 - Bokrug
 - Bugg-Shash
 - Byagoona
 - Byatis
 - Chaugnar Faugn
 - Coatlicue

	a	b	c	Total
*				0,0,0
li			1	0,0,1
.foo		1		0,1,0
#chapter1	1			1,0,0
ul li			1 × 2	0,0,2
.foo > li		1	1	0,1,1
ul ol li.steps		1	1 × 3	0,1,3
li.steps.mech		1 × 2	1	0,2,1
style="foo"				∞

```
HTML
1 <div class="a b c d e f g h i j k l m n o p q r s t u v
  w x y z" id="alphabet">🎵 Let's call the whole thing off...
  🎵</div>

* CSS (SCSS)
1 // Specificity: 0,0,1
2 div {
3   color: seagreen;
4   font-size: 48px;
5 }
6
7 // Specificity: 0,26,0
8 .a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.q.r.s.t.u.v.w.x.y.z {
9   color: dodgerblue;
10 }
11
12 // Specificity: 1,0,0
13 #alphabet {
14   color: tomato;
15 }
```

🎵 Let's call the whole thing off... 🎵

Any value > 0 in **a**
outweighs any value > 0 in
b, which outweighs any
value > 0 in **c**

$(a > 0) > (b > 0) > (c > 0)$

1 in **a** outweighs the 26 in **b**

Visual Studio Code shows you the specificity of a selector when you hover over it in your CSS & what it means visually

EXPLORER

...

index-01.htmlmain.css

OPEN EDITORS

BOOTSTRAP-4-101

css

main.css

fonts

images

videos

index-01.html

index-02-jumbotron.h...

index-02.1-jumbotron...

index-02.2-jumbotron...

index-03-breadcrumb...

index-04-nav-click-to...

index-04.1-nav-no-dr...

index-05-fonts.html

OUTLINE

NPM SCRIPTS

TEMPLATES

css > main.css > ...

```
68  *****/
69
70  /* For video background */
71
72  .jumbotron-video {
73      /* must set this so we can use position: absolute for the video
       inside it, but didn't move it because we don't want it to */
74      position: relative;
75      /* don't let video exceed the boundaries we set */
76      --video-width: 100%;
77      ...
78      <video>
79
80      Selector Specificity: (0, 1, 1)
81  .jumbotron-video video {
82      /* ↓ take video out of flow; relied on containing block (.
       jumbotron-rsg-video) using position: relative */
83      position: absolute;
84      /* ↓ make sure video is behind content, which has z-index of 20 */
85      z-index: 10;
86      /* ↓ place video in area */
```

<element class="jumbotron-video">

...

<video>

Selector Specificity: (0, 1, 1)

.jumbotron-video video {

/* ↓ take video out of flow; relied on containing block (.
jumbotron-rsg-video) using position: relative */

position: absolute;

/* ↓ make sure video is behind content, which has z-index of 20 */

z-index: 10;

/* ↓ place video in area */

EXPLORER

> OPEN EDITORS

✓ BOOTSTRAP-4-101

css

main.css

> fonts

> images

> videos

index-01.html

index-02-jumbotron.h...

index-02.1-jumbotron...

index-02.2-jumbotron...

index-03-breadcrumb...

index-04-nav-click-to...

index-04.1-nav-no-dr...

index-05-fonts.html

> OUTLINE

> NPM SCRIPTS

> TEMPLATES

index-01.html

main.css

css > main.css > .breadcrumb-item.active

```
105  /* For breadcrumb */
106
107  .jumbotron-breadcrumb {
108    padding: 0 2rem;
109    margin-top: -2rem;
110    border-top: 1px dotted lightgrey;
111  }
112
113  .breadcrumb-item + .breadcrumb-item::before {
114    <element class="breadcrumb-item active">
115
116    Selector Specificity: (0, 2, 0)
117  .breadcrumb-item.active {
118    color: #333;
119  }
120
121  /* Footer */
122
123  footer {
124    text-align: center;
125    border-top: 1px dotted darkgrey;
```

<<

⊗ 0 ⚠ 0

Live Share

⚡

GIST [github]

Assignments > bootstrap-4-101

CSS

Go Live

Rsync: ⓘ

[off]

👤

🔔

EXPLORER

> OPEN EDITORS

✓ BOOTSTRAP-4-101

- css
 - main.css
- fonts
- images
- videos
- index-01.html
- index-02-jumbotron.h...
- index-02.1-jumbotron...
- index-02.2-jumbotron...
- index-03-breadcrumb...
- index-04-nav-click-to...
- index-04.1-nav-no-dr...
- index-05-fonts.html

> OUTLINE

> NPM SCRIPTS

> TEMPLATES

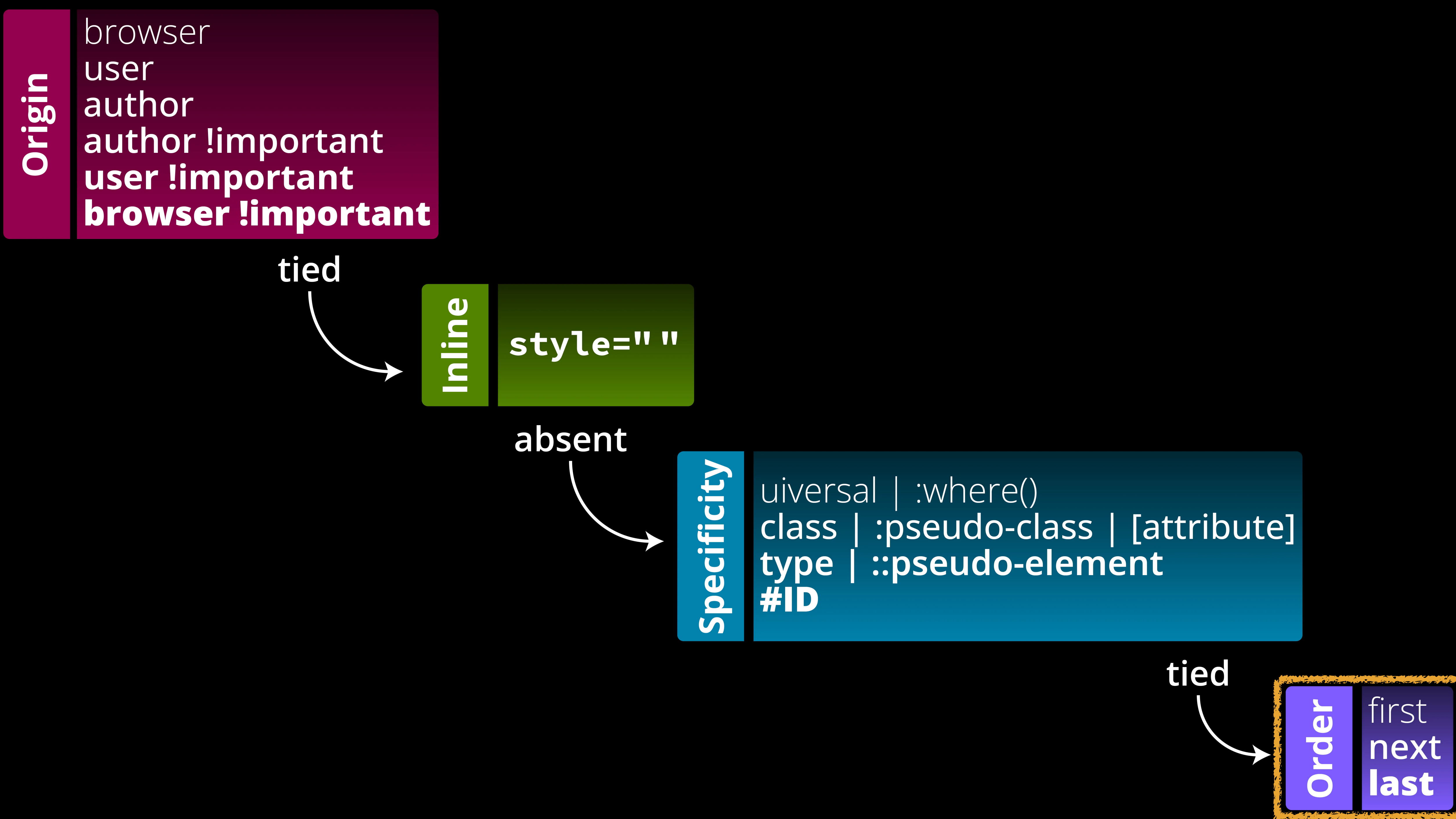
index-01.html

main.css

css > main.css > .breadcrumb-item.active

```
105  /* For breadcrumb */
106
107  .jumbotron-breadcrumb {
108      padding: 0 2rem;
109
110      <element class="breadcrumb-item">
111      <element class="breadcrumb-item" ::before>
112
113      Selector Specificity: (0, 2, 1)
114      .breadcrumb-item + .breadcrumb-item::before {
115          content: ">";
116      }
117
118      .breadcrumb-item.active {
119          color: #333;
120      }
121
122  /* Footer */
123
124  footer {
125      text-align: center;
126      border-top: 1px dotted darkgrey;
```

Order



Later CSS in the stylesheet wins over earlier CSS

```
main.css:
```

```
.blue {color: blue}  
.red {color: red}
```

```
index.html:
```

```
<p class="red blue">  
  What color am I?  
</p>
```

main.css:

```
.blue {color: blue}  
.red {color: red}
```

index.html:

```
<p class="red blue">  
  What color am I?  
</p>
```

main.css:

```
.red {color: red}  
.blue {color: blue}
```

index.html:

```
<p class="blue red">  
  What color am I?  
</p>
```

main.css:

```
.red {color: red}  
.blue {color: blue}
```

index.html:

```
<p class="blue red">  
  What color am I?  
</p>
```

The Cascade

Origin

browser
user
author
author !important
user !important
browser !important

Conflicting CSS declarations follow the cascade until the one with the most weight wins

tied

Inline

style=""

absent

Specificity

universal | :where()
class | :pseudo-class | [attribute]
type | ::pseudo-element
#ID

tied

Order

first
next
last


CSS Cascade (basic)


Let me walk you through a real example that happened to me in which knowing how the Cascade works made my life a lot easier

Here's some code I have on a website:

```
<div class="callOut">  
  <p>  
    For the next 2 weeks...  
  </p>  
</div>
```

```
.callOut {  
  background-color: #E6E8F2;  
  margin: 1em 1em 2em 1em;  
  padding: 1em;  
  border: 1px #ccc solid;  
  border-radius: 1em;  
}
```

Source: [xkcd](#) 

For the next 2 weeks, in addition to learning about wikis, you're also going to edit articles at Wikipedia. In order to fulfill this assignment, please [create an account at Wikipedia](#)  if you do not already have one. Do this as soon as possible, as you must have an account in existence for a few days before you can create/edit certain articles at Wikipedia.

Before you begin your reading, please read & comment: [Wiki & Wikipedia usage among students](#).

Definitions

Read all of these.

- [Computerworld Defines WikiWikiWeb](#)
Wiki defined.
- [Wikis in Plain English](#)
A short movie that explains the concept of wikis and how they work.


Why & how do wikis work?


Read all of these.

- [The Wiki Way](#)
A review of a book about Wikis explains their philosophy and why they work.
- [Some useful & interesting wikis \(besides Wikipedia\)](#)
A list of informational & useful wikis, besides Wikipedia.

The result!
However...

- Broadcast Flag
- Bullying
- Business-Education Partnerships
- Business Intelligence
- Business School
- Cameras
- Capitalism
- Cartels
- CBDTPA
- Cell Phones
- Climate Change
- Cloud Computing
- Collaboration
- Comic Books
- Communications
- Concepts
- Consolidation
- Content Management Systems
- Conveniences
- Copy Protection
- Copyright
- Cracking
- Craigslist
- Creative Commons
- Crime
- Cultural Differences
- Customer Tracking
- Database
- Death
- DeCSS
- Definition
- Demographics
- Design
- DeskSwap
- Developing Nations

Source: [xkcd](#) 

For the next 2 weeks, in addition to learning about wikis, you're also going to edit articles at Wikipedia. In order to fulfill this assignment, please [create an account at Wikipedia](#)  if you do not already have one. Do this as soon as possible, as you must have an account in existence for a few days before you can create/edit certain articles at Wikipedia.

Before you begin your reading, please read & comment: [Wiki & Wikipedia usage among students](#).

Definitions

Read all of these.

- [Computerworld Defines WikiWikiWeb](#)
Wiki defined.
- [Wikis in Plain English](#)
A short movie that explains the concept of wikis and how they work.

Why & how do wikis work?

Read all of these.

- [The Wiki Way](#)
A review of a book about Wikis explains their philosophy and why they work.
- [Some useful & interesting wikis \(besides Wikipedia\)](#)
A list of informational & useful wikis, besides Wikipedia.

- Broadcast Flag
- Bullying
- Business-Education Partnerships
- Business Intelligence
- Business School
- Cameras
- Capitalism
- Cartels
- CBDTPA
- Cell Phones
- Climate Change
- Cloud Computing
- Collaboration
- Comic Books
- Communications
- Concepts
- Consolidation
- Content Management Systems
- Conveniences
- Copy Protection
- Copyright
- Cracking
- Craigslist
- Creative Commons
- Crime
- Cultural Differences
- Customer Tracking
- Database
- Death
- DeCSS
- Definition
- Demographics
- Design
- DeskSwap
- Developing Nations

For the next 2 weeks, in addition to learning about wikis, you're also going to edit articles at Wikipedia. In order to fulfill this assignment, please [create an account at Wikipedia](#) if you do not already have one. Do this as soon as possible, as you must have an account in existence for a few days before you can create/edit certain articles at Wikipedia.

- Broadcast Flag
- Bullying
- Business-Education Partnerships
- Business Intelligence
- Business School
- Cameras
- Capitalism
- Cartels
- CDDPA

46

For the next 2 weeks, in addition to learning about wikis, you're also going to edit articles at Wikipedia. In order to fulfill this assignment, please [create an account at Wikipedia](#) if you do not already have one. Do this as soon as possible, as you must have an account in existence for a few days before you can create/edit certain articles at Wikipedia.

61

Why & how do wikis work?

Read all of these

- [The Wiki Way](#)
A review of a
- [Some useful](#)
A list of informational & useful wikis, besides Wikipedia.

That extra space at the bottom really bothers me

- Cultural Differences
- Customer Tracking
- Database
- Death
- DeCSS
- Definition
- Demographics
- Design
- DeskSwap
- Developing Nations


To fix it, I put this in my CSS at lines 194–196:

```
.callOut > p:last-child {  
    margin-bottom: 0;  
}
```

Let me explain what that means...

> is a *combinator* that selects the direct children of (*not* descendants of) the `.callOut` class

`p:last-child` is a *pseudo-class* that means the `<p>` that is the last child of (*not* descendant of) the `.callOut` class



```
.callOut > p:last-child {  
    margin-bottom: 0;  
}
```

So this selects the `<p>` that is the *last direct child* of the `.callOut` class

EXAMPLE

```
.callOut > p:last-child {  
  margin-bottom: 0;  
}
```

```
<div class="callOut">  
  <p>foo</p>  
  <p>bar</p>  
  <p>baz</p>  
  <p>quz</p>  
  <blockquote>  
    <p>quux</p>  
  </blockquote>  
</div>
```

← Selected!

← Not selected, as this **<p>** is
not the last direct child

Why use this?

```
.callOut > p:last-child {  
    margin-bottom: 0;  
}
```

Why not just use this?

```
.callOut > p {  
    margin-bottom: 0;  
}
```


Because sometimes there are 2 or more paragraphs inside `.callOut` — & I don't want to select *every* paragraph

So back to where I was... I put this in my CSS at line 194:

```
.callOut > p:last-child {  
    margin-bottom: 0;  
}
```

Let's check the webpage...

46

For the next 2 weeks, in addition to learning about wikis, you're also going to edit articles at Wikipedia. In order to fulfill this assignment, please [create an account at Wikipedia](#)  if you do not already have one. Do this as soon as possible, as you must have an account in existence for a few days before you can create/edit certain articles at Wikipedia.

61

Nothing changed! 

Why didn't it work?

Let's open the Inspector & find out why

Source: [xkcd](#)

For the next 2 weeks, in addition to learning about wikis, you're also going to edit articles at Wikipedia. In order to fulfill this assignment, please [create an account at Wikipedia](#) if you do not already have one. Do this as soon as possible, as you must have an account in existence for a few days before you can create/edit certain articles at Wikipedia.

p 574px × 120px

Before you begin your reading, please read & comment: [Wiki & Wikipedia usage among students](#).

Definitions

- Broadcast Flag
- Bullying
- Business-Education Partnerships
- Business Intelligence
- Business School
- Cameras
- Capitalism
- Cartels
- CBDTPA
- Cell Phones
- Climate Change
- Cloud Computing
- Collaboration
- Comic Books
- Communications
- Concepts
- Consolidation
- Content Management Systems

×

📄

📑

↺

⬇

📄 72

🛒

🕒

💬 0

🚫 11

⚠ 0

🎯

🔍 Search

🧱 Elements

🌐 Network

📄 Resources

🕒 Timelines

🐛 Debugger

💾 Storage

📝 Console

+

E > E > E > E > E > E > E

div#post-10096.post-10096.post.type-post.status-publish.... > div.entry-content > div.callOut > p

⏏

🔍

🖋

📏

<script>onmousedown=function(){document.getElementById('primary').style.display='block';}else{document.getElementById('primary').style.display='none';document.getElementById('content').style.margin='0px 200px;'}}Toggle Visibility</script>

<div class="entry-meta">...</div>

<!-- .entry-meta -->

<div class="entry-content">

<p>...</p>

<div class="callOut">

<p>...</p> = \$0

</div>

<p>...</p>

<h2>Definitions</h2>

<p>Read all of these.</p>

...

<h2>Why & how do wikis work?</h2>

<n>Read all of these.</n>

Node

Styles

Layers

Computed

Rules

Metrics

#content p — style.css:146


margin-bottom: 12px;


.callOut > p:last-child — style.css:194

margin-bottom: 0;

Filter Styles

Main Frame

```
 #content p — style.css:146  
margin-bottom: 12px;
```

```
 .callout > p:last-child — style.css:194  
margin-bottom: 0;
```

Hmmm... mine is being beaten by earlier code

Lines 146–148:

```
#content p {  
    margin-bottom: 12px;  
}
```

Lines 194–196:

```
.callOut > p:last-child {  
    margin-bottom: 0;  
}
```

Why isn't mine winning since it's later in order?

Lines 146–148:

```
#content p {  
    margin-bottom: 12px;  
}
```

Lines 194–196:

```
.callOut > p:last-child {  
    margin-bottom: 0;  
}
```

Is *origin & importance* causing the 1st declaration block to win? 🙋

Lines 146–148:

```
#content p {  
    margin-bottom: 12px;  
}
```

Lines 194–196:

```
.callOut > p:last-child {  
    margin-bottom: 0;  
}
```

Is *specificity* causing the 1st declaration block to win? 🤔

```
#content p {  
    margin-bottom: 12px;  
}
```

Specificity

$(\text{\#content} = 1 \times a) +$
 $(p = 1 \times c) =$
 $1, 0, 1$

```
.callout > p:last-  
child {  
    margin-bottom: 0;  
}
```

Specificity

$(.callout = 1 \times b) +$
 $(p = 1 \times c) +$
 $(:last-child = 1 \times b) =$
 $0, 2, 1$

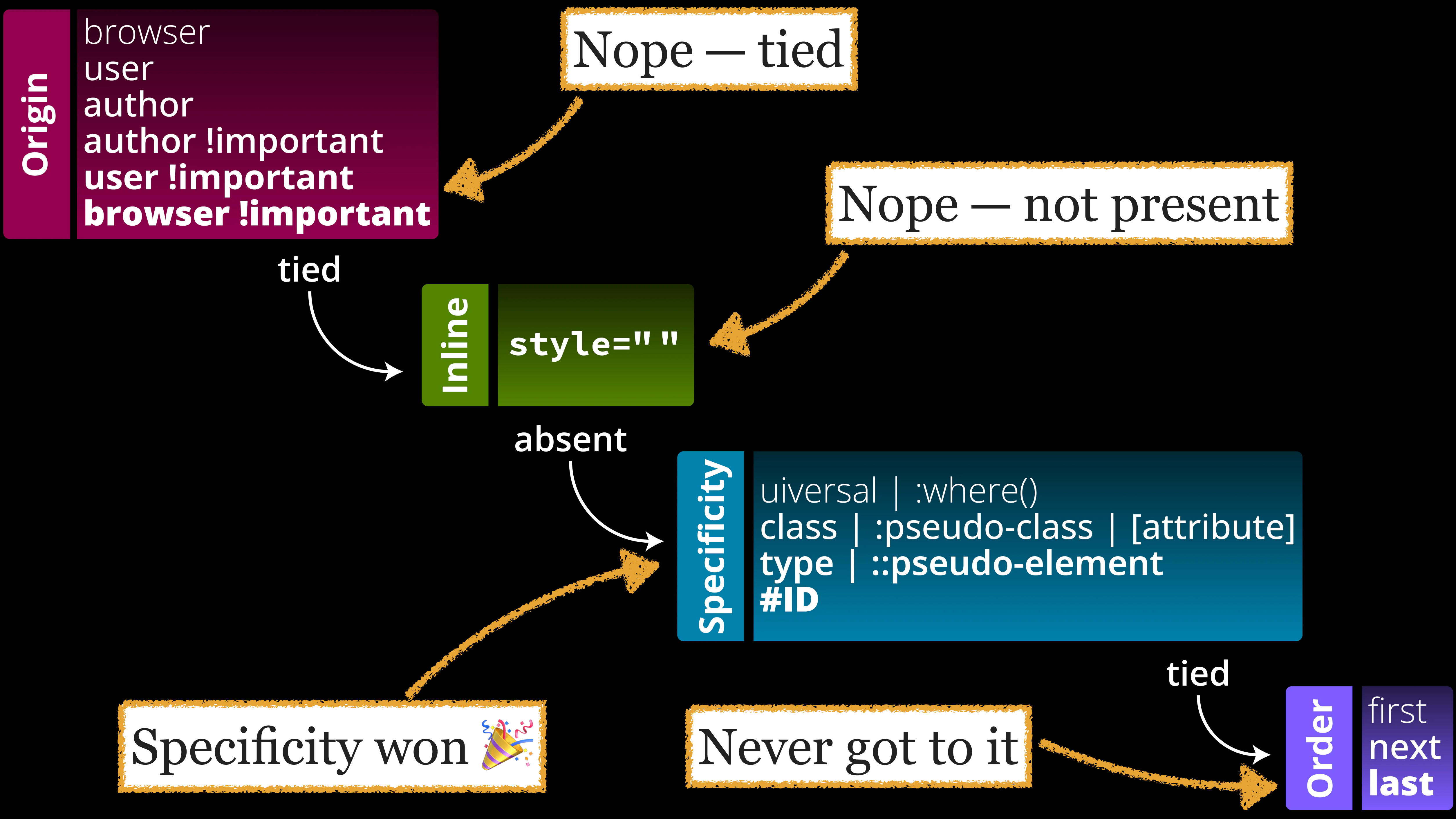
Lines 146–148:

```
#content p {                                1,0,1
  margin-bottom: 12px;
}
```

Lines 194–196:

```
.callOut > p:last-child {
  margin-bottom: 0;                          0,2,1
}
```

Because *specificity* meant the 1st declaration block had the most weight, *order* never entered into the picture



There are 3 solutions to this problem

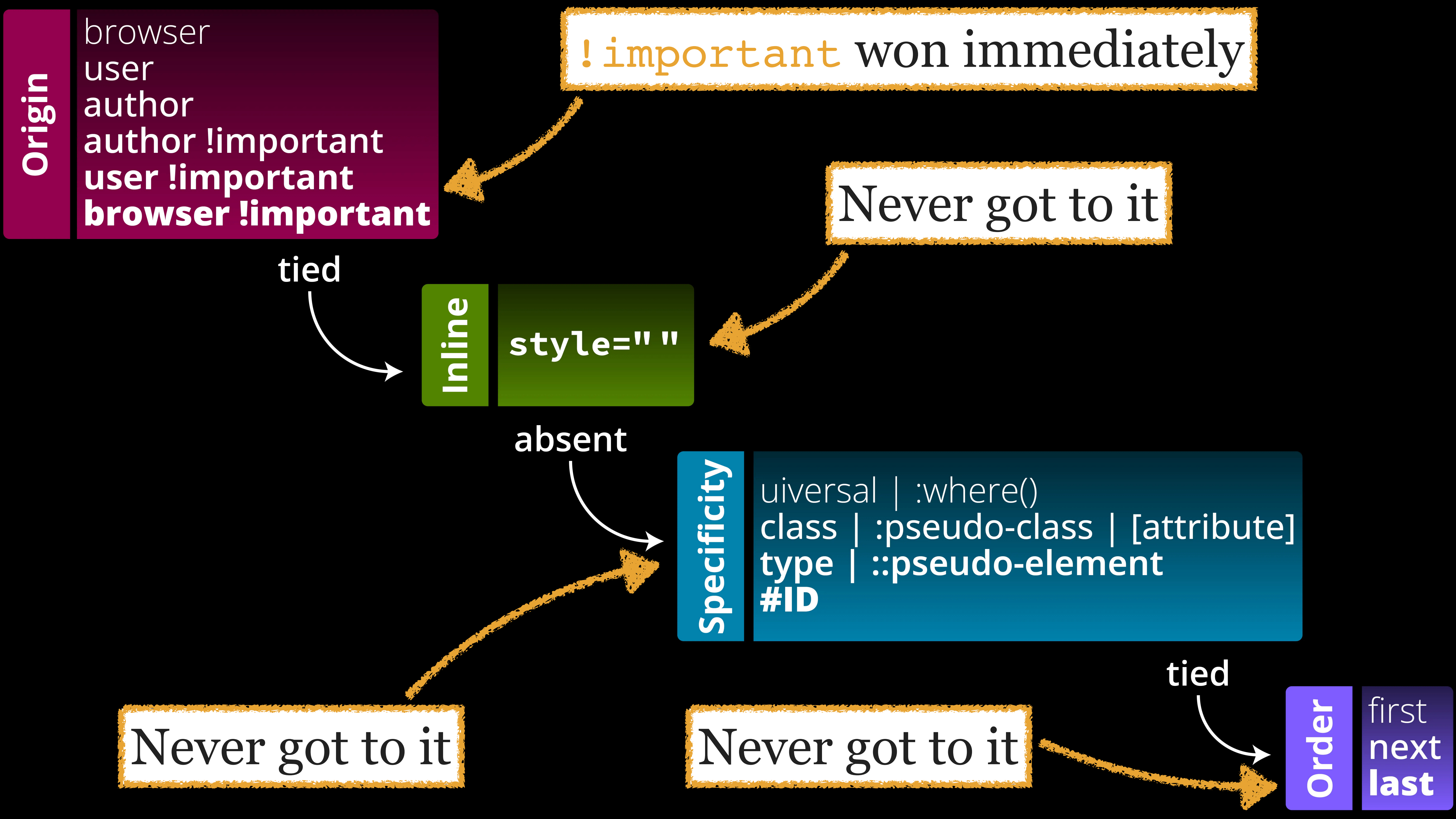
Lines 146–148:

```
#content p {  
    margin-bottom: 12px;  
}
```

Lines 194–196:

```
.callOut > p:last-child {  
    margin-bottom: 0 !important;  
}
```

Adding `!important` makes the 2nd style rule win due to *origin & importance*, so *specificity & order* never come into play



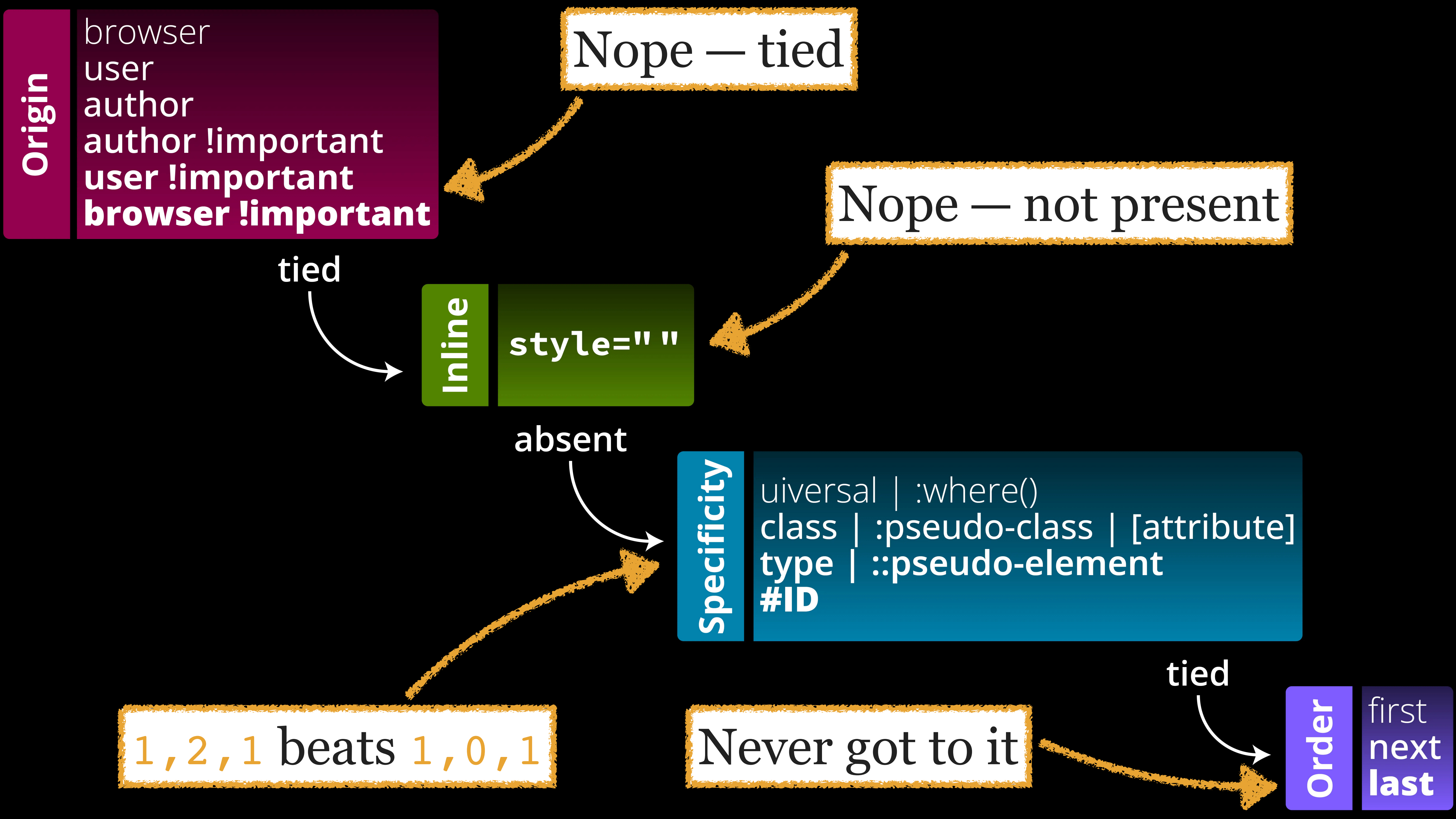
Lines 146–148:

```
#content p {                                1,0,1
    margin-bottom: 12px;
}
```

Lines 194–196:

```
#content .callOut > p:last-child {
    margin-bottom: 0;
                                           1,2,1
}
```

Adding `#content` to line 194 makes the 2nd style rule win due to *specificity*, so *order & importance* never comes into play




Lines 146–148:


```
#content p {  
    margin-bottom: 12px;           0,0,1  
}
```

Lines 194–196:

```
.callOut > p:last-child {  
    margin-bottom: 0;             0,2,1  
}
```

Deleting `#content` on line 146 makes the 2nd style rule win due to *specificity*, but I can't do that or things will break!

Source: [xkcd](#) 

For the next 2 weeks, in addition to learning about wikis, you're also going to edit articles at Wikipedia. In order to fulfill this assignment, please [create an account at Wikipedia](#)  if you do not already have one. Do this as soon as possible, as you must have an account in existence for a few days before you can create/edit certain articles at Wikipedia.

Before you begin your reading, please read & comment: [Wiki & Wikipedia usage among students](#).

Definitions

Read all of these.

- [Computerworld Defines WikiWikiWeb](#)
Wiki defined.
- [Wikis in Plain English](#)
A short movie that explains the concept of wikis and how they work.

Why & how do wikis work?

Read all of these.

- [The Wiki Way](#)
A review of a book about Wikis explains their philosophy and why they work.
- [Some useful & interesting wikis \(besides Wikipedia\)](#)
A list of informational & useful wikis, besides Wikipedia.

Much better!

- Broadcast Flag
- Bullying
- Business-Education Partnerships
- Business Intelligence
- Business School
- Cameras
- Capitalism
- Cartels
- CBDTPA
- Cell Phones
- Climate Change
- Cloud Computing
- Collaboration
- Comic Books
- Communications
- Concepts
- Consolidation
- Content Management Systems
- Conveniences
- Copy Protection
- Copyright
- Cracking
- Craigslist
- Creative Commons
- Crime
- Cultural Differences
- Customer Tracking
- Database
- Death
- DeCSS
- Definition
- Demographics
- Design
- DeskSwap
- Developing Nations

Source: [xkcd](#)

For the next 2 weeks, in addition to learning about wikis, you're also going to edit articles at Wikipedia. In order to fulfill this assignment, please [create an account at Wikipedia](#) if you do not already have one. Do this as soon as possible, as you must have an account in existence for a few days before you can create/edit certain articles at Wikipedia.

- Broadcast Flag
- Bullying
- Business-Education Partnerships
- Business Intelligence
- Business School
- Cameras
- Capitalism
- Cartels
- CBDTPA
- Cell Phones

46

For the next 2 weeks, in addition to learning about wikis, you're also going to edit articles at Wikipedia. In order to fulfill this assignment, please [create an account at Wikipedia](#) if you do not already have one. Do this as soon as possible, as you must have an account in existence for a few days before you can create/edit certain articles at Wikipedia.

37

Why & How do Wikis Work.

Read all of these.

- [The Wiki Way](#)
A review of a book about Wikis explains their philosophy and why they work.
- [Some useful & interesting wikis \(besides Wikipedia\)](#)
A list of informational & useful wikis, besides Wikipedia.

- Cultural Differences
- Customer Tracking
- Database
- Death
- DeCSS
- Definition
- Demographics
- Design
- DeskSwap
- Developing Nations

Bonus question: how do we get rid of those extra pixels at the top?

46

For the next 2 weeks, in addition to learning about wikis, you're also going to edit articles at Wikipedia. In order to fulfill this assignment, please [create an account at Wikipedia](#) if you do not already have one. Do this as soon as possible, as you must have an account in existence for a few days before you can create/edit certain articles at Wikipedia.

37

Read all of these.

- [The Wiki Way](#)
A review of a book about Wikis explains their philosophy and why they work.
- [Some useful & interesting wikis \(besides Wikipedia\)](#)
A list of informational & useful wikis, besides Wikipedia.

- Broadcast Flag
- Bullying
- Business-Education Partnerships

- Cell Phones

- Cultural Differences
- Customer Tracking
- Database
- Death
- DeCSS
- Definition
- Demographics
- Design
- DeskSwap
- Developing Nations

Then I realized that `<div class="callOut">` doesn't always end with `<p>`, so I fixed that:

```
#content .callOut > :last-child {  
    margin-bottom: 0;  
}
```

Now it's weighted to win *and* selecting the correct DOM objects

Remember, `:last-child` is the same as `*:last-child`

Tools

Books



Jon Duckett's *HTML & CSS: Design & Build Websites*

Great overview of (some) HTML5 & CSS2 (& some CSS3)

Getting pretty out of date, though (published in 2011), but still good for the basics

References

RELATED TOPICS

Learn CSS

- ▶ CSS first steps
- ▶ CSS building blocks
- ▶ Styling text
- ▶ CSS layout

Reference

- ▶ Modules
- ▶ Properties
- ▶ Selectors
- ▶ Pseudo-classes
- ▶ Pseudo-elements
- ▶ At-rules
- ▶ Types

Guides


CSS reference

Use this **CSS reference** to browse an [alphabetical index](#) of all of the standard [CSS](#) properties, [pseudo-classes](#), [pseudo-elements](#), [data types](#), [functional notations](#) and [at-rules](#). You can also browse [key CSS concepts](#) and a list of [selectors organized by type](#). Also included is a brief [DOM-CSS / CSSOM reference](#).

Basic rule syntax

Style rule syntax

```
style-rule ::=
  selectors-list {
    properties-list
  }
```



IN THIS ARTICLE

- Basic rule syntax
- Index
- Selectors
- Concepts
- DOM-CSS / CSSOM
- See also
- External Links

696 terms as of April 2022!

Fresh Article July 6, 2020

Bootstrap 5



Author

Chris Coyier

Comments

Start Conversation →

It's always notable when the world biggest CSS framework goes up a major version (it's in alpha now).

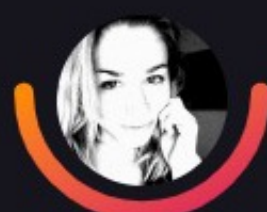
It has dropped jQuery and IE, started using some CSS custom properties, gone fully customized with form elements, started to embrace utility classes, and includes a massive icon set you can use via SVG sprite. Sweet....

[Read article](#)

Popular this month
Sponsored by

Article Jun 26, 2020

In Defense of a Fussy Website



Author
Sarah
Drasner

CARE

INTENTION

PORTFOLIO

UX

Article Jun 17, 2020

Advice for Complex CSS Illustrations



Author
Jhey
Tompkins

ART

CSS SHAPES

Article Jul 1, 2020

A Complete Guide to Dark Mode on the Web



Author
Adhuam

DARK MODE

Article Jun 15, 2020

What is Developer Experience (DX)?



Author
Chris Coyier

DX

UX

Article Jun 12, 2020

Striking a Balance Between Native and Custom Select Elements



Author
Sandra
Pereira

ACCESSIBILITY

FORMS

SELECT

Some of the Guides on CSS Tricks (+ many more!):

- » A Complete Guide to Flexbox ★
- » Media Queries for Standard Devices
- » A Complete Guide to Grid ★
- » A Complete Guide to the Table Element
- » Centering in CSS: A Complete Guide
- » A Complete Guide to CSS Cascade Layers
- » A Nerd's Guide to Color on the Web
- » A Complete Guide to Data Attributes

Can I use

[Settings](#)

CSS

- `::placeholder` CSS pseudo-element
- `::selection` CSS pseudo-element
- `:placeholder-shown` CSS pseudo-class
- `@font-face` Web fonts
- Blending of HTML/SVG elements
- `calc()` as CSS unit value
- 2.1 selectors
- `background-attachment`
- `background-blend-mode`
- `background-position` edge offsets
- `box-decoration-break`
- `clip-path` property
- Counter Styles
- Counters
- `currentColor` value
- Device Adaptation
- Exclusions Level 1
- Feature Queries
- Filter Effects

HTML5

- Audio element
- Audio Tracks
- Autofocus attribute
- Canvas (basic support)
- Canvas blend modes
- `classList` (`DOMTokenList`)
- Color input type
- `contenteditable` attribute (basic support)
- Custom Elements
- Custom protocol handling
- Datalist element
- `dataset` & `data-*` attributes
- Date and time input types
- Details & Summary elements
- Dialog element
- `disabled` attribute of the `fieldset` element
- Download attribute
- Drag and Drop
- Email, telephone & URL input types

SVG

- Inline SVG in HTML5
- SVG (basic support)
- SVG effects for HTML
- SVG favicons
- SVG filters
- SVG fonts
- SVG fragment identifiers
- SVG in CSS backgrounds
- SVG in HTML `img` element
- SVG SMIL animation
- **All SVG features**

JS API

- Ambient Light API
- Base64 encoding and decoding
- Battery Status API
- Blob constructing
- Blob URLs

Can I use grid

? ⚙ Settings

2 results found

#

CSS Grid Layout 📄 - CR

Global

69.53% + 5.62% = 75.15%

unprefixed:

69.53%

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for layout into columns and rows using a set of predictable sizing behaviors

Current aligned

Usage relative

Date relative

Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			1 49			10.2			
	2 15	55	60	10.1	47	10.3		4.4	
2 11	16	56	61	11	48	11	all	56	61
		57	62	TP	49				
		58	63		50				
		59	64						

Notes

Known issues (2)

Resources (12)

Feedback

```
/* Styles */

@import "extra.css";

ul li
{
    list-style: square;
    margin: 2em 20% 15px 0;
}

#content
{
    -webkit-font-smoothing: antialiased;
    background: url(img/gradient.png);
    background: linear-gradient(to bottom, red, rgba(255,0,0,0));
}

@media only screen and (min-width: 35em)
{
    #content
    {
        width: 50%;
    }
}

.item + .item,
.item ~ .item
```

apps.workflower.fi/vocabs/css/

Comment

Statement

Rule-set

At-rule

At-keyword

Media query

Media query list

Media type

Expression

Media feature

Block

Declaration block

Selector

Simple selector

Type selector

Universal selector

Compound selector

Class selector

Attribute selector

```
/* Styles */
```

```
@import "extra.css";
```

```
ul li  
{  
    list-style: square;  
    margin: 2em 20% 15px 0;  
}
```

```
#content  
{  
    -webkit-font-smoothing: antialiased;  
    background: url(img/gradient.png);  
    background: linear-gradient(to bottom, red, rgba(255,0,0,0));  
}
```

```
@media only screen and (min-width: 35em)  
{  
    #content  
    {  
        width: 50%;  
    }  
}
```

```
.item + .item,
```

```
.item ~ .item
```

CSS VOCABULARY

?

Comment

Statement

Rule-set

At-rule

At-keyword

Media query

Media query list

Media type

Expression

Media feature

Block

Declaration block

Selector

Simple selector

Type selector

Universal selector

ID selector

Class selector

Attribute selector

```

/* Styles */

@import "extra.css";

ul li
{
    list-style: square;
    margin: 2em 20% 15px 0;
}

#content
{
    -webkit-font-smoothing: antialiased;
    background: url(img/gradient.png);
    background: linear-gradient(to bottom, red, rgba(255,0,0,0));
}

@media only screen and (min-width: 35em)
{
    #content
    {
        width: 50%;
    }
}

.item + .item,
.item ~ .item

```

- Class selector
- Attribute selector
- Pseudo-class
- Pseudo-element
- Combinator
 - Descendant combinator
 - Child combinator
 - Adjacent sibling combinator
 - General sibling combinator
- Declaration
- Property
- Value
- Function
- Keyword
- String
- URL
- Number
- Percentage
- Length
- Unit
- Color
- Vendor prefix

rsgranne + css 192

RSS

Search All

« earlier

About CodePen

CodePen is a playground for the front end side of the web. It's all about inspiration, education, and sharing.

Need to build a reduced test case to demonstrate and figure out a bug? CodePen is great for that. Want to show off your latest creation and get feedback from your peers? CodePen is great for that. Want to find example of a particular design pattern for you project? CodePen is great for that.

CodePen PRO takes things further with features like Collab Mode which allows you to pair program in real time and Professor Mode for allowing a group of students to follow you as you teach code and chat with each other.

code css design javascript webdev html collaboration service websanity teaching webster

24 days ago by rsgranne copy to mine

Download Google Fonts - Fonts.com

Sync Google Fonts to your Desktop with SkyFonts

We are proud to have teamed up with Google to offer desktop versions of their popular Google Fonts free of charge. Offered for use in print, these fonts are delivered using SkyFont's patent-pending font delivery technology and can be used anywhere.

Each time a font is updated — such as when new characters are added — SkyFonts will automatically update the font on your device. Syncing Google Fonts with SkyFonts will also improve your web browsing experience, by cutting the time spent downloading fonts.

fonts google typography css

24 days ago by rsgranne copy to mine

BuiltWith Web Technology Usage Statistics

Amazingly useful set of statistics about hundreds of trends in code, design, & technology usage on the Web.

statistics technology web webdev websanity html css

5 weeks ago by rsgranne copy to mine

related tags

addons + adobe + advertising + ajax + animation

+ api + apple + art + assignment + blog +

bookmarks + books + browser + business +

characterentities + chart + cheatsheet +

chrome + cms + code + collaboration +

collection + color + commercial + compression

+ conversion + creativity + CSS - demo +

design + dhtml + diagnostic + dictionary +

directory + div + documentation + download

editing + education + email +

encmeyer + examples + extensions +

firefox + flash + flickr + fonts + framework

+ free + friends + fun + generator +

pinboard.in/u:rsgranne/t:css

Browser Tools

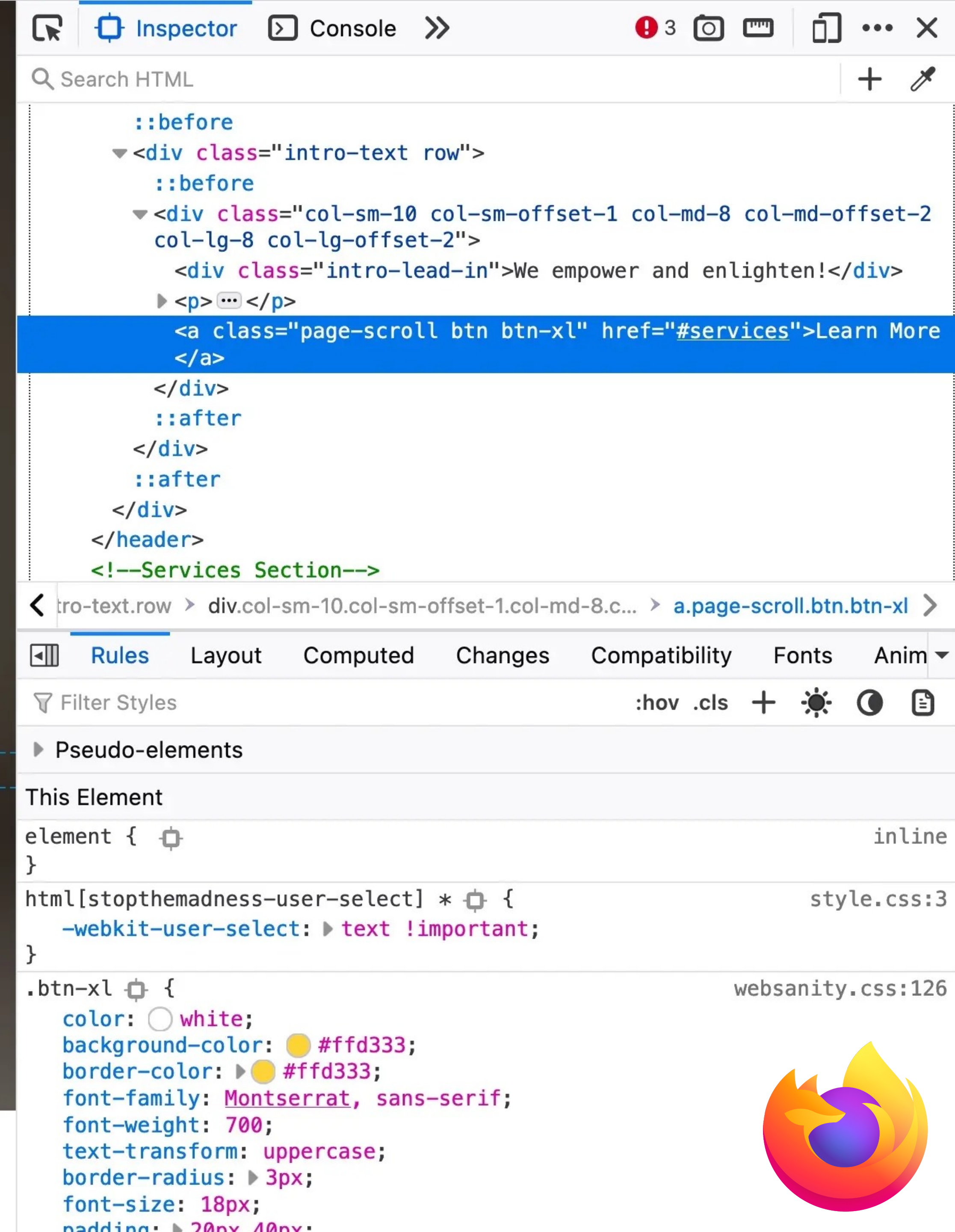
Built-in developer tools (aka DevTools or the Inspector)
are excellent

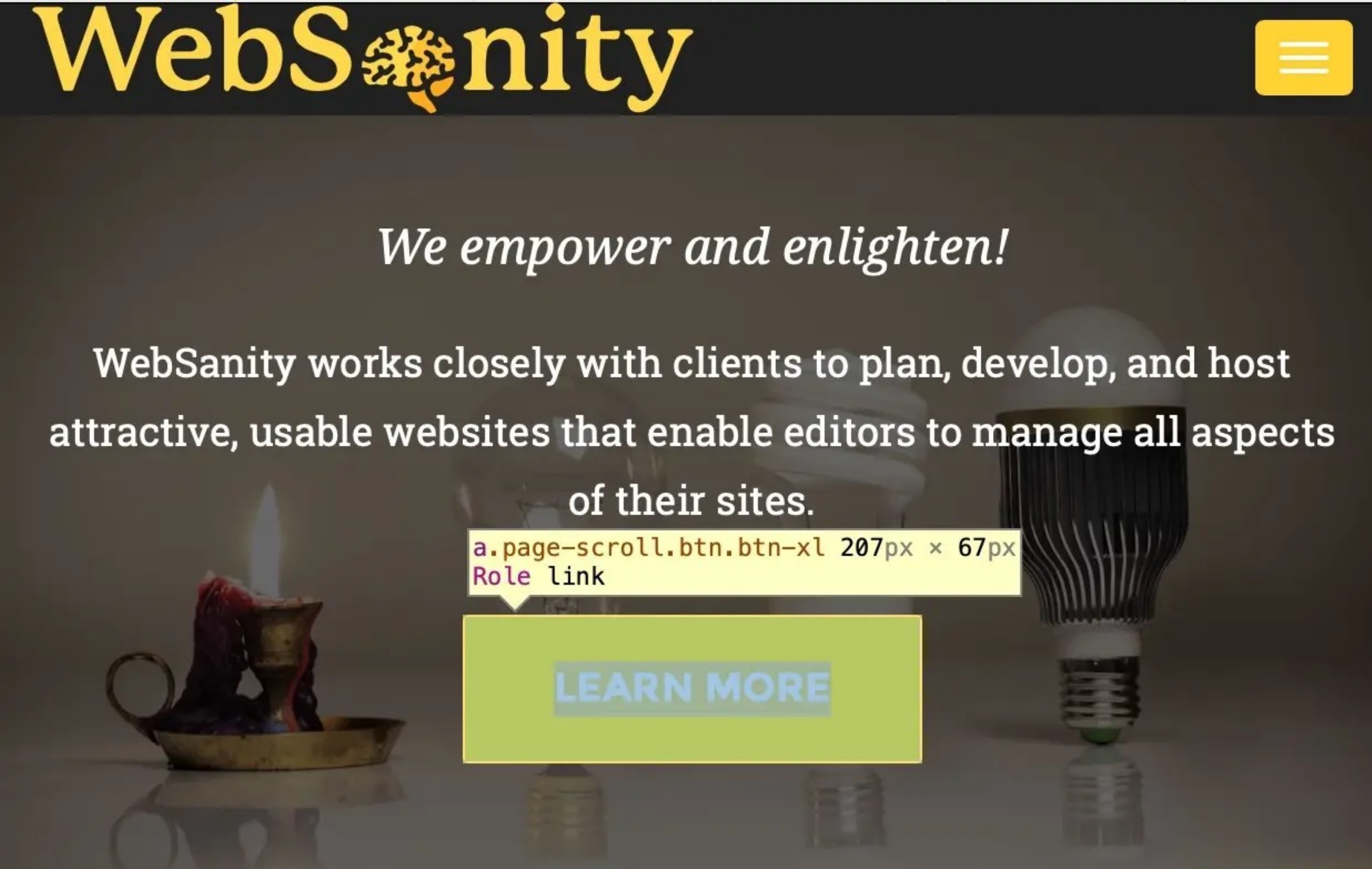
We empower and enlighten!

WebSanity works closely with clients to plan, develop, and host attractive, usable websites that enable editors to manage all aspects of their sites.

a.page-scroll.btn.btn-xl	206.817 × 67.7167
--------------------------	-------------------

LEARN MORE





HOW WE SERVE YOU

We help you build the perfect website.



Plan

You bring industry knowledge, professional expertise, and insight unique to

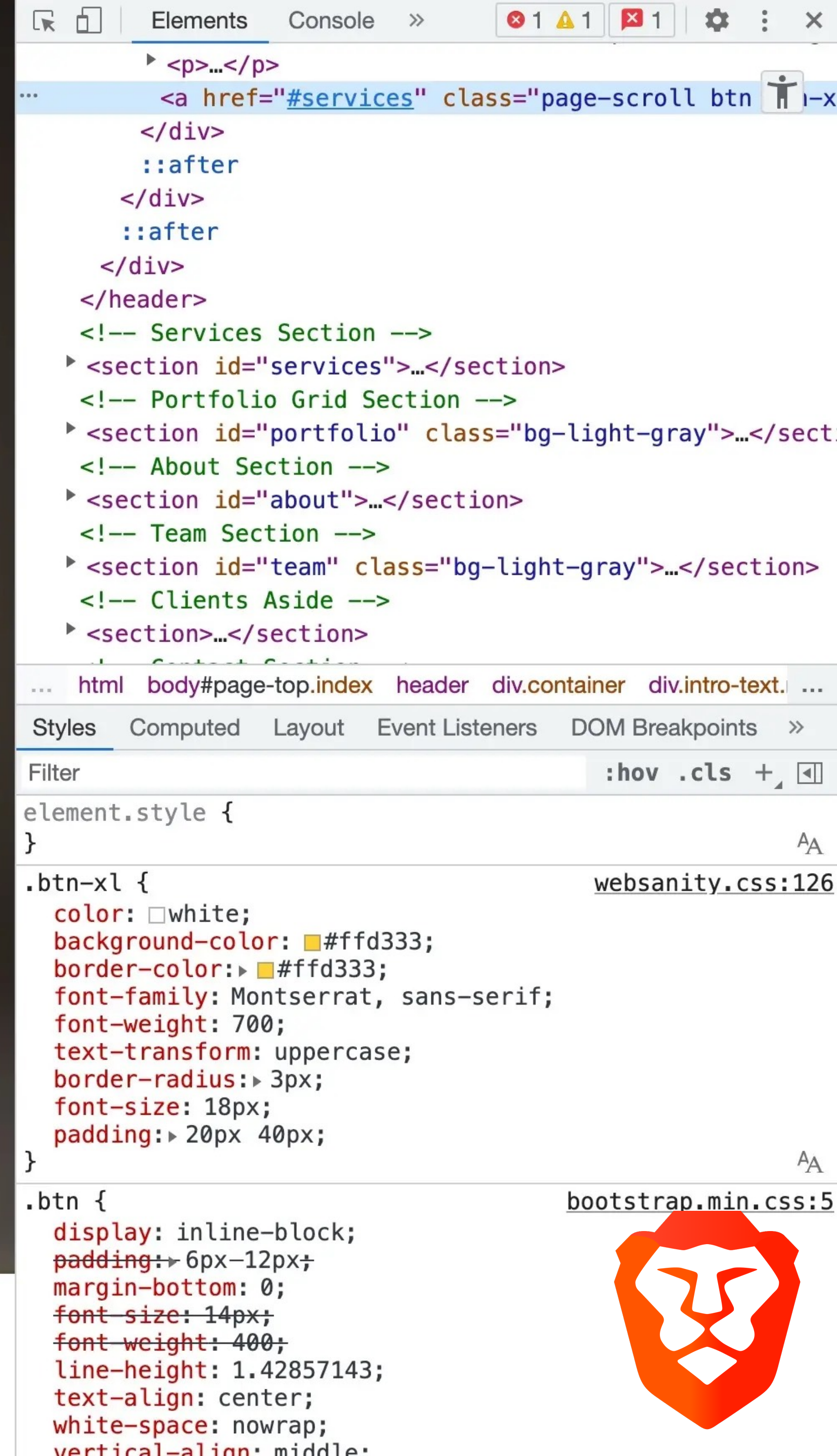
<

We empower and enlighten!

WebSanity works closely with clients to plan, develop, and host attractive, usable websites that enable editors to manage all aspects of their sites.

a.page-scroll.btn.btn-xl 206.85 x 67.71

LEARN MORE



Color Pickers



System Color Picker by Sindre Sorhus

Color picker

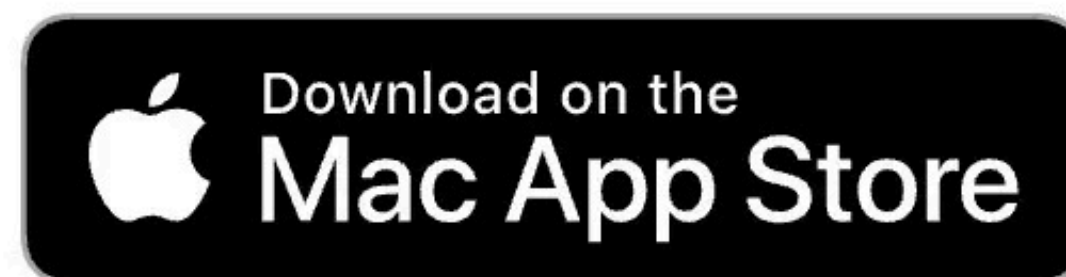
sindresorhus.com/system-color-picker

\$0 in the Mac App Store (& open source!)



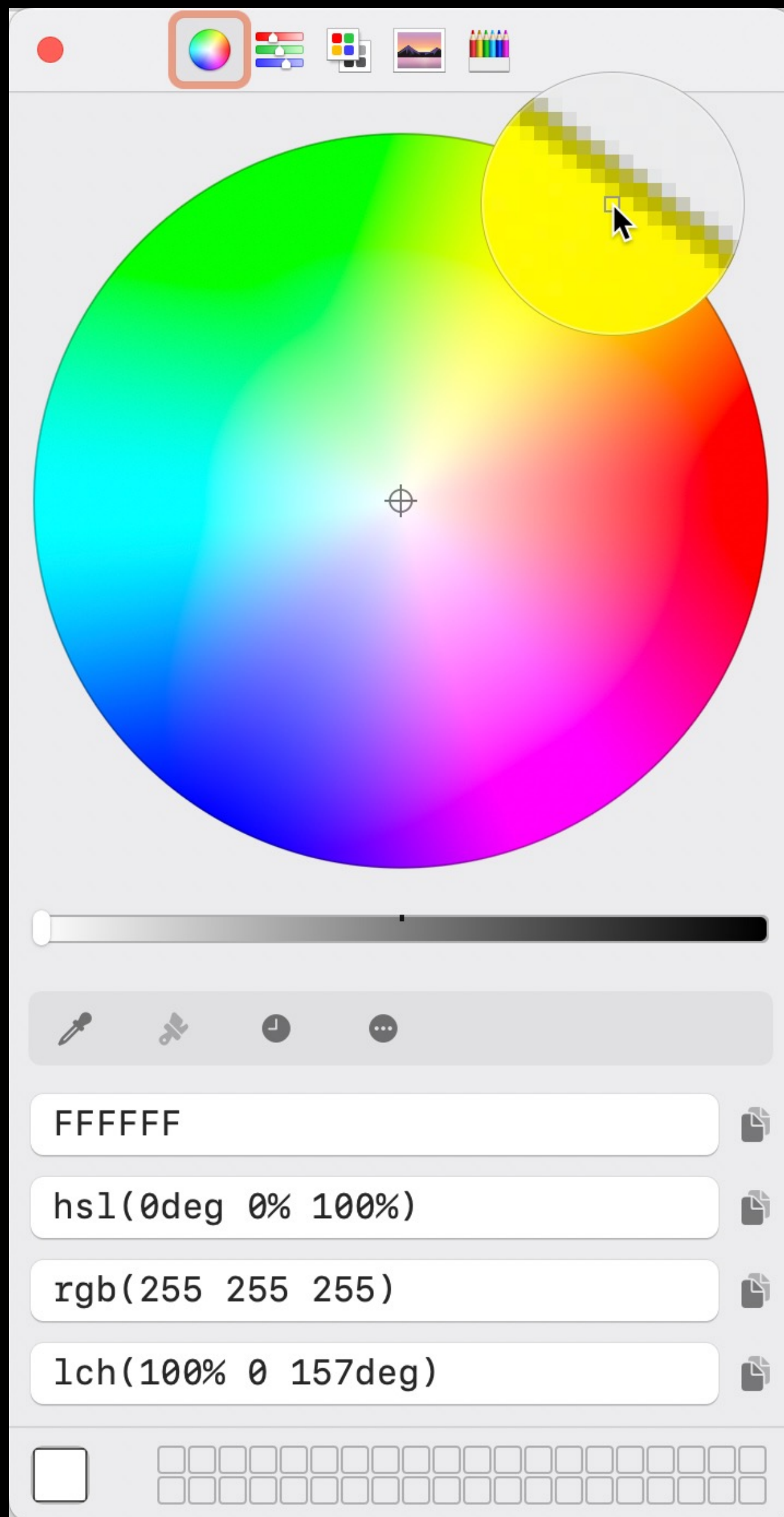
System Color Picker

The familiar color picker supercharged

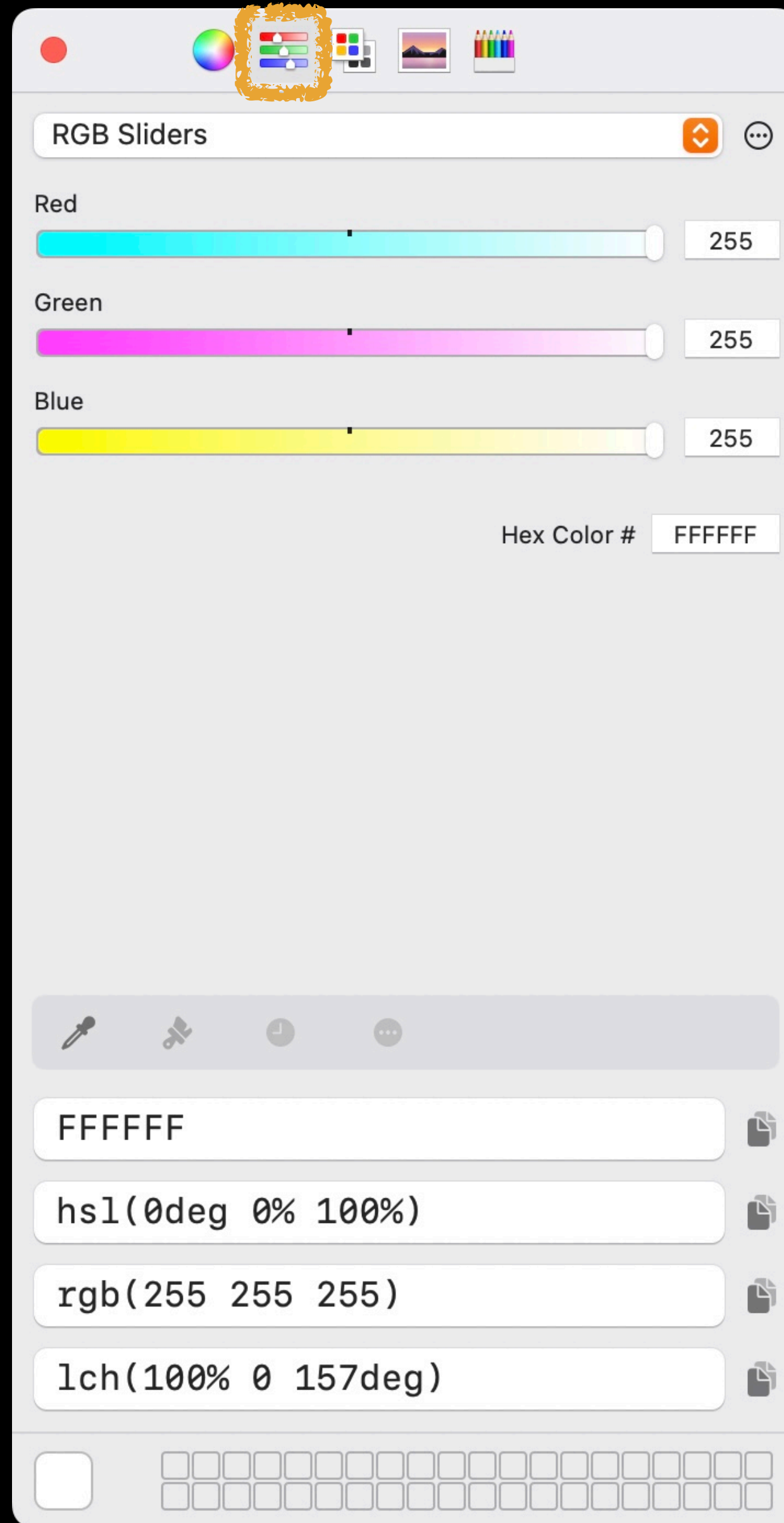


[Read more](#)

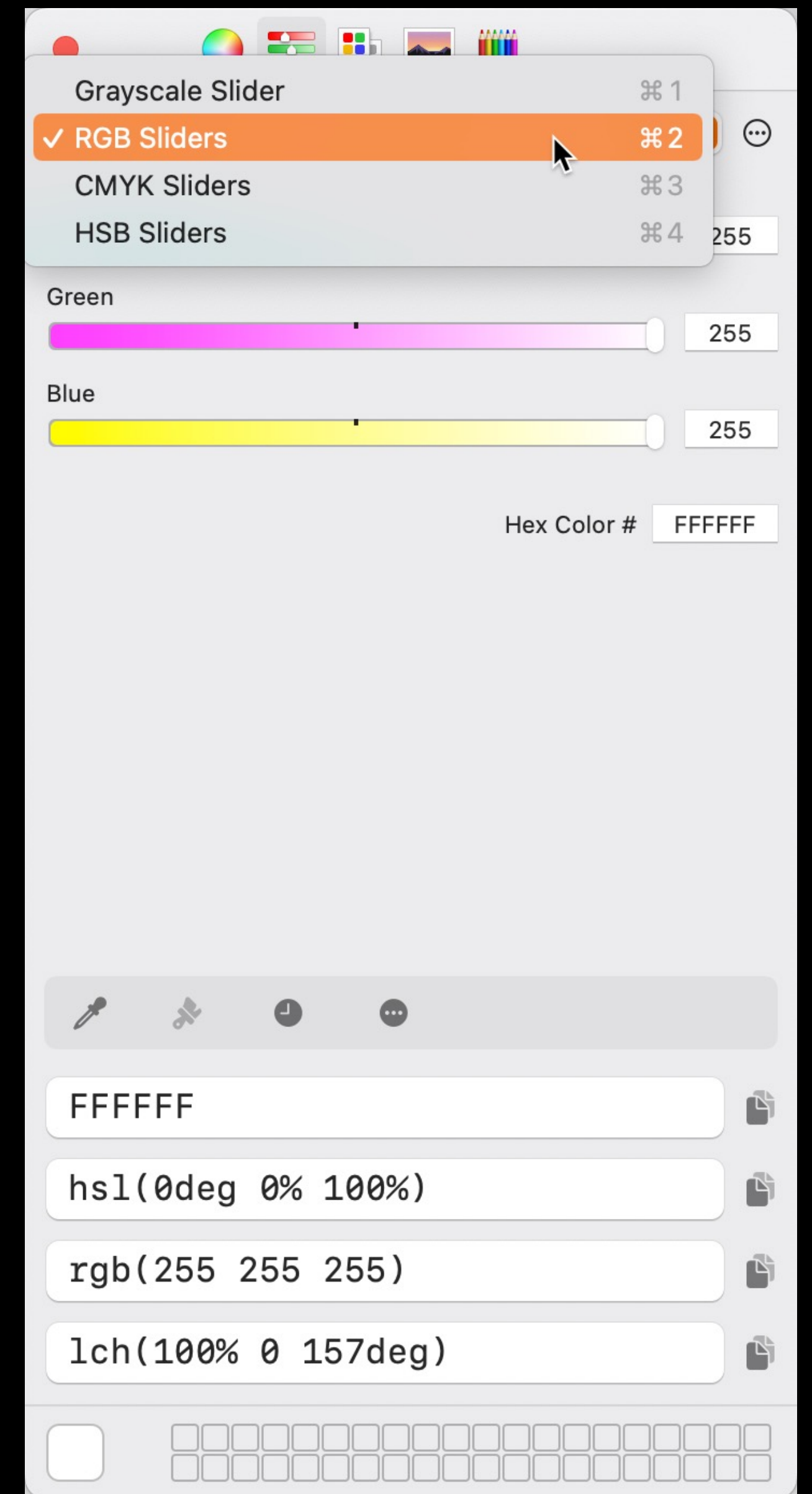
[Source on GitHub](#)

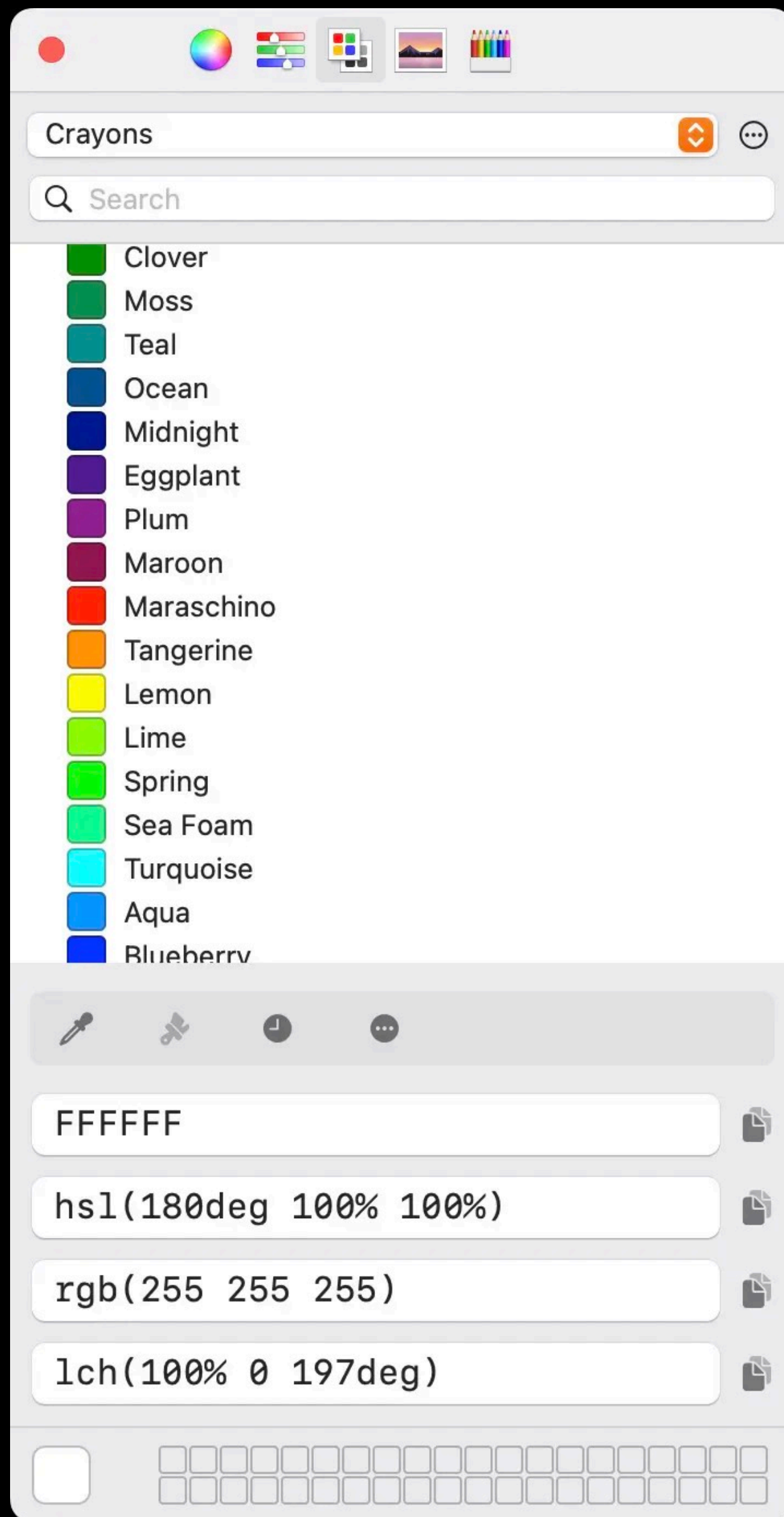


Color Wheel

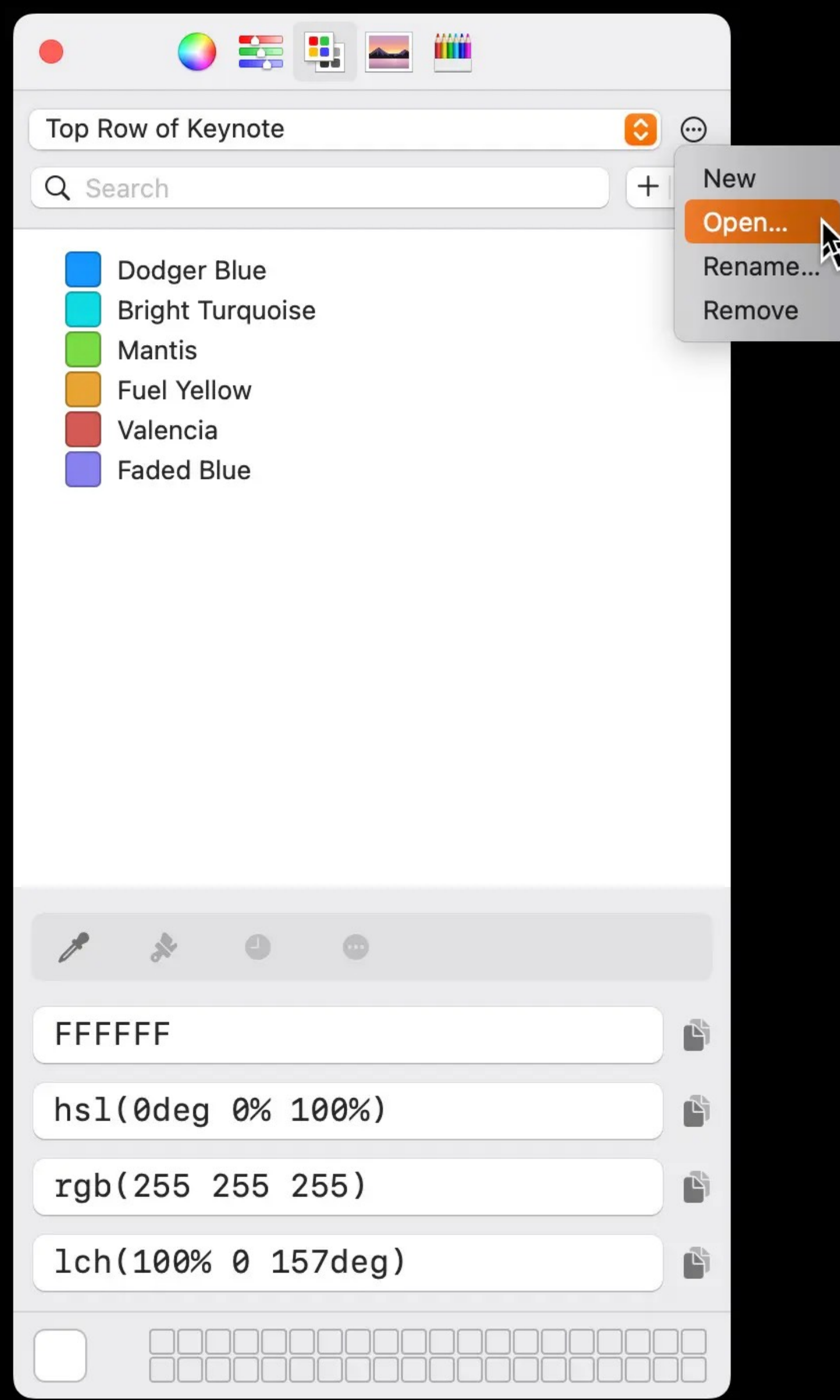


Sliders

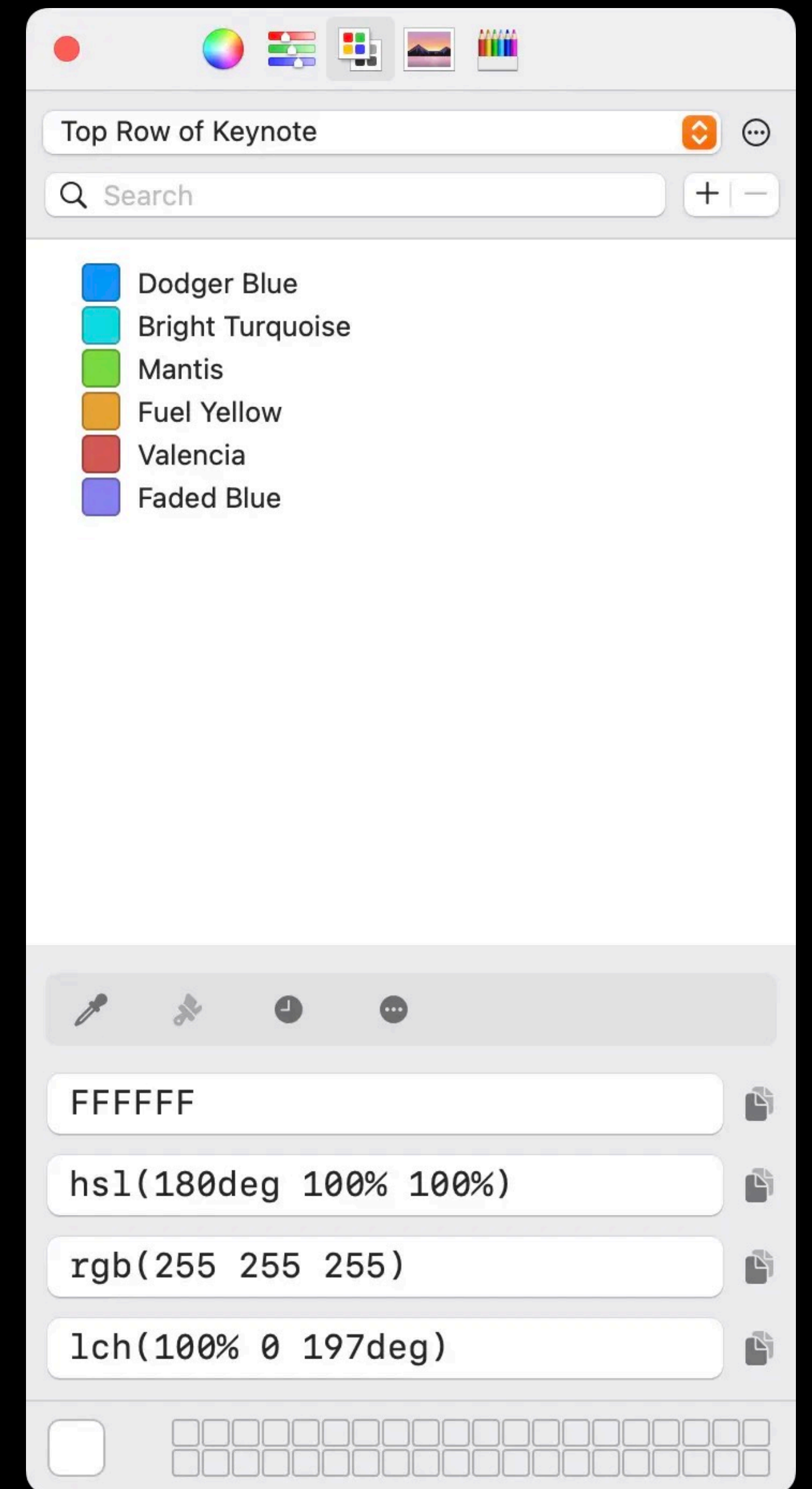




Color Palettes

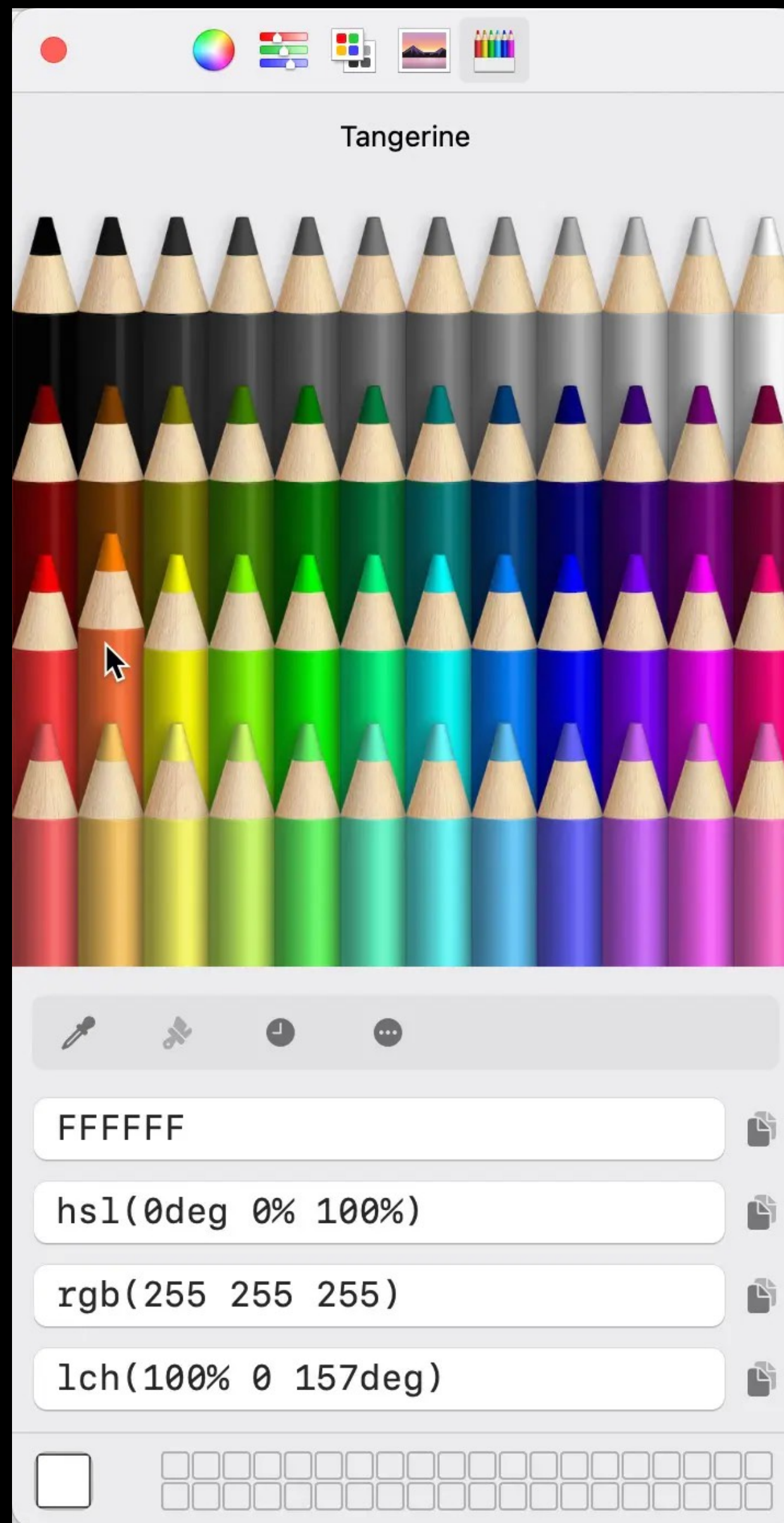


Open palettes generated elsewhere

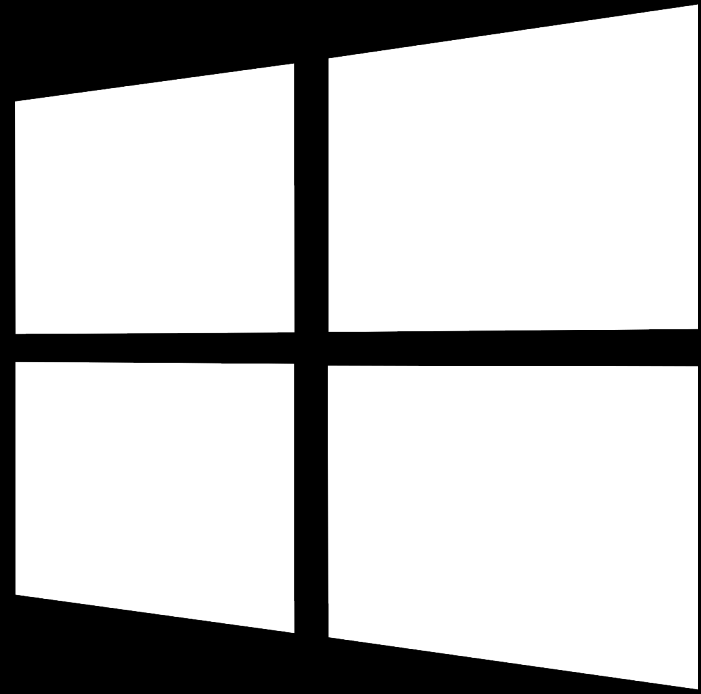


Add favorites

Pencils



Supports
hexadecimal, HSL,
RGB, & OKLCH



Color Picker, from Microsoft PowerToys

One of many small, very useful tools

\$0

docs.microsoft.com/en-us/windows/powertoys/

Windows App Development

[Explore](#) ▾[Development](#) ▾[Platforms](#) ▾[Resources](#) ▾[Dashboard](#) Filter by title[Windows development environment](#)▾ [Developer tools](#)[Windows Subsystem for Linux](#)[Windows Terminal](#)[Windows Package Manager](#)▾ [PowerToys](#)▾ [Overview](#)[About PowerToys](#)[Install PowerToys](#)[Run in admin mode](#)[Report a Bug](#)[Always on Top](#)[PowerToys Awake](#)[Color Picker](#)[FancyZones](#) [Download PDF](#)[...](#) / [PowerToys](#) /

Microsoft PowerToys: Utilities to customize Windows

Article • 07/06/2022 • 4 minutes to read • [10 contributors](#)

Microsoft PowerToys is a set of utilities for power users to tune and streamline their Windows experience for greater productivity.

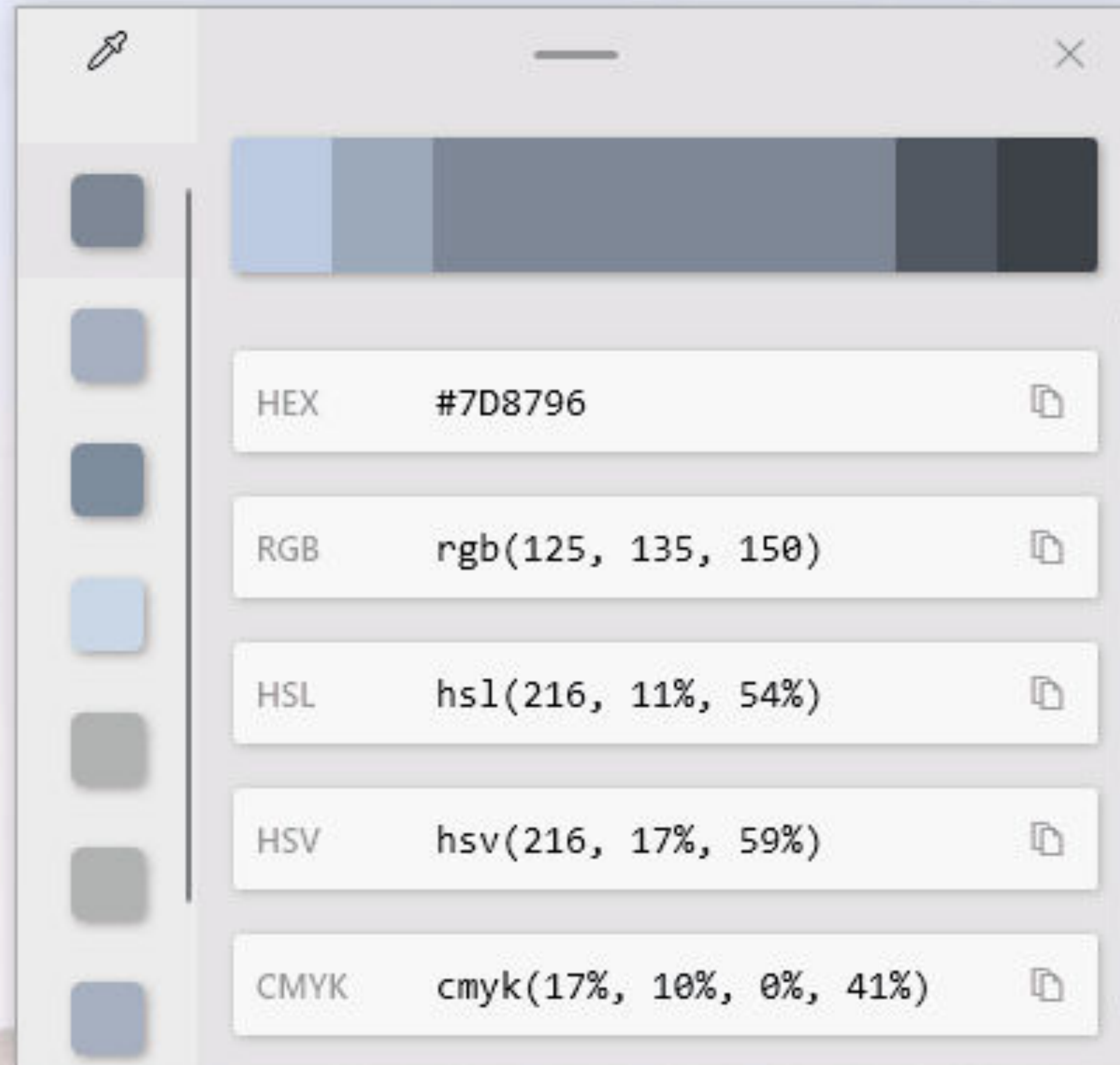
[Install PowerToys](#)

Processor support

- **x64:** Supported
- **Arm:** Supported

In this article

[Processor support](#)[Current PowerToy utilities](#)[Languages](#)[PowerToys video walk-through](#)[Show more](#) ▾



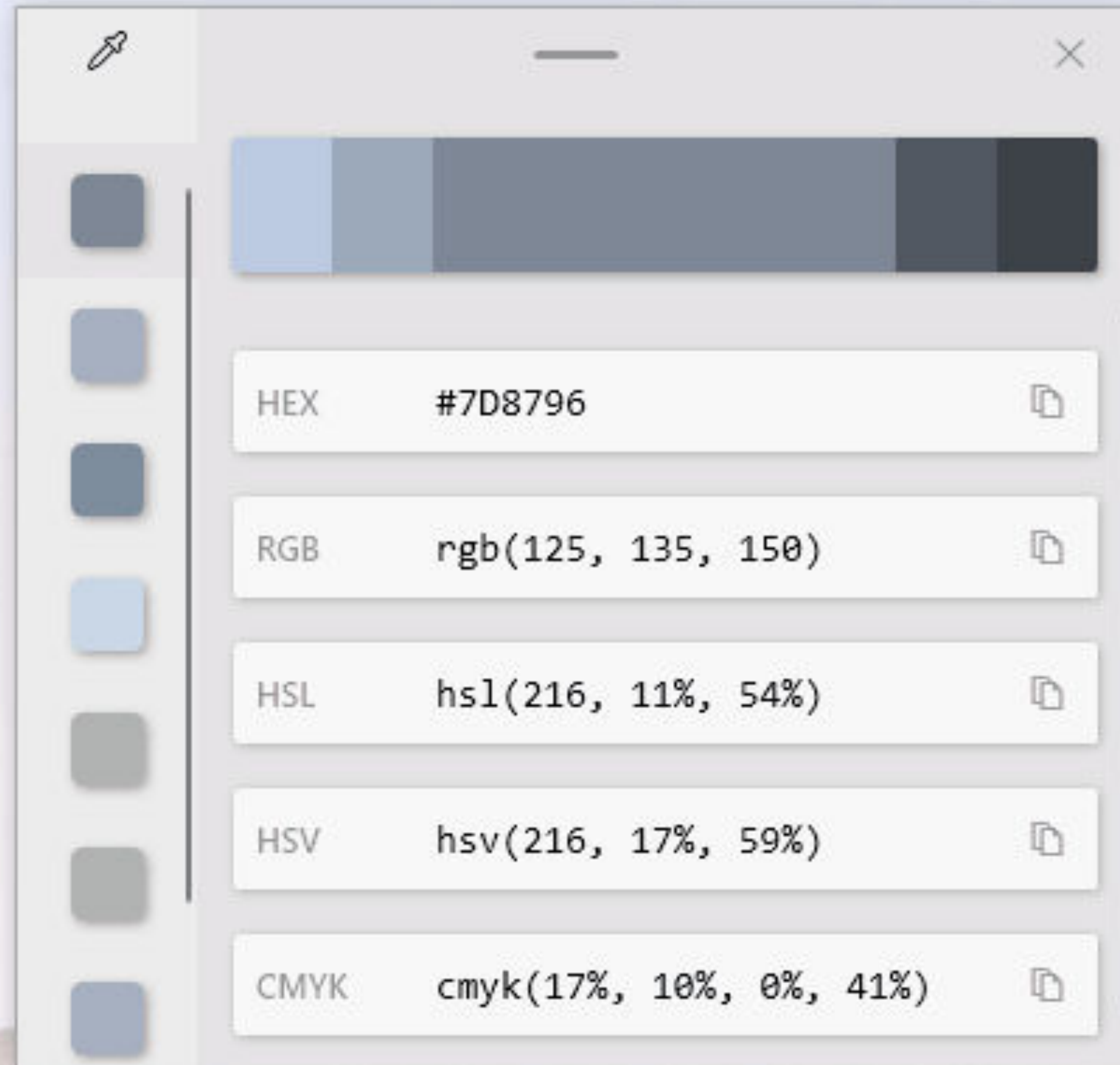
HEX #7D8796

RGB rgb(125, 135, 150)

HSL hsl(216, 11%, 54%)

HSV hsv(216, 17%, 59%)

CMYK cmyk(17%, 10%, 0%, 41%)



HEX #7D8796

RGB rgb(125, 135, 150)

HSL hsl(216, 11%, 54%)

HSV hsv(216, 17%, 59%)

CMYK cmyk(17%, 10%, 0%, 41%)



Slides

granneman.com/downloads/web-dev/CSS-Overview-Condensed.pdf

chnsa.ws/css-condensed-slides



Notes

granneman.com/downloads/web-dev/CSS-Overview-Condensed.txt

chnsa.ws/css-condensed-notes



All of my presentations

granneman.com/presentations/all-presentations

chnsa.ws/all-presentations

Thank you!

scott@granneman.com

granneman.com

@scottgranneman@mastodon.social

jans@websanity.com

websanity.com

CSS Overview Condensed

Selectors, Integration, Inheritance, Cascading

R. Scott Granneman & Jans Carton

© 2009 R. Scott Granneman • v3.4 2023-10-10 • Licensed CC BY-SA 4.0

Bonus

Changelog

2025-03-03 3.6: Updated theme to Granneman 1.14; minor formatting & wording fixes; labeled browser processing pipeline; updated number of Web APIs; hid Opera's built-in CSS; Color Picker supports OKLCH; added quotation re: developer understanding of CSS

2024-10-14 3.5: Changed Twitter link to Mastodon; added more quotes re: developer attitudes towards CSS; replaced Separation of Concerns slide with better one; added better example for descendant combinator; updated number of selectors; better examples for selector list

Changelog

2023-10-10 3.4: Added another slide on terminology to *Basic Selectors*; updated *Page Fragment Identifiers* to be a lot clearer; added slide showing HTML & DOM tree; converted theme to Granneman 1.11; minor formatting & wording fixes; increased %age of all websites using CSS; expanded explanations of complex selectors; added *Bonus* section & moved slides for * there, then simplified example for *; made *The Cascade* a little clearer; minor fixes in *Cascading Style Sheets*; clarified example using * with child combinator; slowed video for *Page Fragment Identifiers* by 50%; added hateful comment re: CSS

Changelog

2023-03-30 3.3: Updated Cascade diagram to fix spelling mistake

2023-03-14 3.2: Added slide showing block & inline boxes in `` & `<div>`; added example for browser's use of `!important`; minor changes; greatly expanded * example; added Chapter 3 slides for *Page Fragment Identifiers & JavaScript Hooks*; added short video showing page fragments in action

Changelog

2022-11-21 3.1: Added note that `` (separation of concerns) is obsolete

2022-10-04 3.0: (con't. from ↓) fixed links & screenshot for System Color Picker; re-did IE user CSS slide layout; re-did all Cascade screenshots; made my Cascade example clearer; linked to Pixel Winch; many other fixes, corrections, & improvements; so many changes I bumped it up to 3.0!

Changelog

2022-10-04 3.0: (con't. from ↓) better labeling & citations for *Formatting*; added full citations & better class names to `` & `<div>`; added citations & annotations to Inheritance; changed *Key Selectors* to *Selector Subjects*; added number of inherited properties; better images for *Formatting*; better explained when rendering engine matches, inherits, or cascades; ; removed *CSS Reset Keywords*; added more info to Jon Duckett's book; (con't. ↑)

Changelog

2022-10-04 3.0: (con't. from ↓) reworded `@import`; updated list of CSS 4 selectors & added citations; moved * last in *Simple Selectors*; you can override type selectors; details re: multiple classes; better on emojis for class names; added note re: `.center`; better annotations for `#ID`; moved *Universal* after *ID*; better labels for key selectors; removed complex example from *Child Combinator* & better labeled remaining examples; better labeling for *Selector List* examples; (con't. ↑)

Changelog

2022-10-04 3.0: In *Color Pickers*, replaced Sip with System Color Picker by Sindre Sorhus; added disco music in *Why CSS?*; changed *ruleset* to *style rule* per new W3C renaming; between annotations when introducing style rules; deleted details about CSS & UTF-8; added why you should keep CSS `<link>`s to a minimum; moved reasons for embedding to Notes; condensed why not to use inline styles; condensed `@import`; better annotations on *The Thing* & the DOM; added citations; (con't. ↑)

Changelog

2022-07-24 2.12: Added Color Picker from Microsoft PowerToys to *Color Pickers* & got rid of other Windows-based tools; added warning re descendant combinator

2022-04-15 2.11: Better specificity calculation screenshot; added UserStyles.world screenshots to Stylus; removed a few Stylus screenshots; added chart & *Raising Arizona* clip to Cascade to show conflicting directions; updated Cascade diagrams; moved example of using complex selector instead of class from descendent to child combinator

Changelog

2022-04-15 2.10: Added or updated all citations in *Specificity*; more examples of what get ignored when calculating specificity; updated theme to Granneman 1.7; minor formatting corrections; updated definition of the DOM; better universal selector example; moved more scoped universal selector example to child combinator; added shish kebab icon to Kebab case; new screenshot for `<div>`; added OpenDyslexic3 font to Origin & Importance; added screenshots of Stylus; new screenshots of Inspectors; add more complete, correct citations

Changelog

2021-11-17 2.9: Added Side Note re camelCase & other case names; added simple Browser Processing Pipeline before full diagram as reminder; added much better example of *** in *Universal*; clarified *Specificity*; added Spider-Man head icon to make my joke super-obvious

2021-08-16 2.8: Explained how `@import` is different from Sass `@import`; removed wrong advice to put `@charset` at top of style sheet with emoji; more detail about ~~`@charset`~~ & how parsers detect UTF-8

Changelog

2021-07-13 2.7: Added another explanation of the DOM, from Wikipedia; more improvements to *Specificity*; updated chart in *Reset Keywords*; added conceptual models to DOM

2021-04-12 2.6: Added in Notes that class & ID names are *identifiers*; added better explanation & chart for *Complex Selectors Using Combinators*; completely re-did *Specificity* section; added Side Note under *Linking* about using `@charset "utf-8";`; fixed W3C logo in MDN members

Changelog

2020-11-30 2.5: Re-ordered & changed wording slightly in *Specificity*; updated examples in *Compound Selectors*; added detail re: Shadow DOM & the Cascade; changed *Importance* to *Order & Importance* to match the spec more closely; updated Cascade diagram & improved wording throughout *Cascading Style Sheets*

2020-07-31 2.4: Created diagram for the *Browser Processing Pipeline*; made *Descendant Combinator* examples clearer by adding arrows

Changelog

2020-07-21 2.3: Added list of Web APIs after JavaScript; added slides about design patterns in *Formatting*; added example of looking up property's initial value at MDN; updated screenshot for `initial` value & added explanation to it; added additional, easier examples for descendant & child combinators; changed *CSS Resets* to *CSS Reset Keywords*; improved wording explaining `inherit` value; moved *Key Selectors* at the beginning of *Complex Selectors*

Changelog

2020-07-15 2.2: Minor fixes; re-did *Specificity* completely; removed the *Miller's Crossing* example; made clearer in *Order* what is in HTML & what is in CSS

2020-07-10 2.1: Added note about Safari bug with `all: unset`

Changelog

2020-07-09 2.0: (con't. from ↓) added better descriptions of `` & `<div>`; changed “Default inherited properties” to “Partial list of inherited properties” & added `text-edge`; added detail to MDN’s CSS Reference in *Tools*; updated screenshots of CSS-Tricks in *Tools*; added CSS Resets section for `initial`, `inherit`, `unset`, `revert`, & `all`; added explanation of CSS keywords; so many changes I bumped version number up to 2!

Changelog

2020-07-09 2.0: Added more details to Specificity; minor edits; added definition of simple selector; moved *Key Selectors* under *Complex Selectors*; for user styles, gave more detail re: Firefox's `userContent.css`, Chrome, & extensions, & updated screenshot of Safari's Advanced Preferences; better screenshot for embedded styles; updated screenshot for `.class` selector; updated table showing difference between HTML & CSS terms; added order Jans places things in his stylesheets; (con't. ↑)

Changelog

2018-12-06 1.20: Added screenshots showing embedding with `<style>` & how to link to `main.css`; added logos for MDN supporters; `*:first-child` is the same as `:first-child`; added Side Note re: using 2 `class` attributes; in Formatting, always put `;` at the end of each declaration; added CSS Tricks to Tools; in Class, told viewer to see Compound to learn how to match an element with 2 classes; screenshots for `@import` & reorganized those slides; minor wording changes; better example for Compound selectors; replaced ID screenshots

Changelog

2018-11-21 1.19: Called out items on browser processing pipeline illustration; fixed image for multiple classes (`<figurecaption>?!`); updated screenshots for child combinator; for selector lists, removed silly example & added headings slide; added Side Note on `<div>` that it should semantically be ``; fixed Cascade example so proper method is shown; added slides showing difference between descendant & child combinators; added table on Time & User Perception to `@import`; replaced `inherit` screenshot

Changelog

2018-10-01 1.18: Added how DOM changes attribute-value pairs; replaced DOM section with The Browser Processing Pipeline; fixed Default inherited properties slide; updated theme to Granneman 1.5; added Walt Whitman on contradicting yourself

2017-11-06 1.17: Improved key selectors; added reasons for user CSS

Changelog

2017-10-30 1.16: Added Just Color Picker; added screenshot of ColorPro website; changed color of some arrows & shapes to **Tulip Tree** (**#E8A433**); better solutions to specificity problem with **.callOut**; fixed wording to be correct &/or more specific; added emoji for class names; added Opera user-agent styles; better examples for Key Selectors; applied Granneman 1.4 theme; fixed formatting issues; added default order I use in stylesheets

Changelog

2017-10-25 1.15: Added better examples for DOM vs source code; organized Basic Selectors much better; moved Key Selectors, IDs as page fragment identifiers & JavaScript hooks, & Compound Selector example from CSS Selectors to here; gave full list of selectors & grayed out ones we're not covering here; fixed wording introducing the Cascade

Changelog

2017-10-18 1.14: Took out details about how to turn on Firefox 3D View, since it's not longer supported; made Cascading chapter slide italicized; corrected & added info on anonymous boxes; in History, hid modules & added Can I Use, minor fixes & corrections

2016-09-23 1.13: Moved slide comparing class & ID; under Importance, rearranged IE & Safari & added Firefox for Windows, Chrome, & Edge; re-ordered examples of Descendant Combinator; fixed formatting errors

Changelog

2016-09-16 1.12: Updated theme to Granneman 1.2; small changes in wording to make things clearer; cleaned up formatting in a few places; added slide re: using classes instead of IDs; fixed slides in Selector Grouping; changed Important example from WordPress to Bootstrap; fixed wrong information re: `class` & `id` values & clarified; added example for Descendant Combinator

Changelog

2016-01-20 1.11: Added slide re: CSS3 Taxonomy & Status; better explanation why we need `<div>`; added note re: specificity

2016-01-11 1.10: Minor improvements taken from CSS - Selectors; added Child Combinator to Selectors; added another example of Child Combinator; added screenshots of browser CSS; explained author vs author in Importance; explained how my Safari CSS works; added a long example showing how the Cascade works in practice

Changelog

2015-12-13 1.9: Clarified source of DOM quote; changed numbers of selectors; got rid of **E** & **F** in selectors & made them clearer; changed **.bigRed** to **.big-red**; clarified source of class & ID names; add tweet re: CSS to beginning

2015-05-10 1.8: Added info about CSS 4; clarified that **** & **<div>** draw boxes; added additional names of directories that are always created; changed “What Google prefers” to “... uses”; removed Hues & added Sip to Color Pickers; fixed URL & screenshots for CSS Vocabulary; moved Viewport Resizer to Bootstrap; for Separation of Concerns, added “& Meaning” to HTML

Changelog

2015-03-06 1.7: Added another example of selector grouping; added details about *resource inlining*

2015-01-12 1.6: Added my `safari.css` file; clarified & added info on specificity

2015-01-11 1.5: Clarified Inheritance

2014-09-27 1.4: Changed “browser” to “rendering engine” in a few places where it made sense

Changelog

2014-08-12 1.3: Improved Descendant Selector examples; improved wording & added slides in DOM section; improved Cascade diagram; fixed `<div>` screenshot; added URLs for `<div>` & `` screenshots; fixed Viewport Resizer screenshots

2014-08-10 1.2: Added DOM spec info & screenshots of DOM & Source Code; added details about Firefox Web Dev Tools

2014-08-04 1.1.1: Added definition of *anonymous object*

Licensing of this work

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.

To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/>.

You are free to:

- » *Share* — copy and redistribute the material in any medium or format
- » *Adapt* — remix, transform, and build upon the material for any purpose, even commercially

Under the following terms:

Attribution. You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. Give credit to:

Scott Granneman • www.granneman.com • scott@granneman.com

Share Alike. If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions. You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Questions? Email scott@granneman.com