THE CASE FOR MODERN CSS

By Chen Hui Jing / @hj_chen
“What we don't understand, we fear. What we fear, we judge as evil. What we judge as evil, we attempt to control. And what we cannot control...we **attack**.”

—Anonymous
How well do you know CSS? 🎨

Given these classes:

```
.red {
  color: red;
}

.blue {
  color: blue;
}
```

Which color would these divs be?

```
<div class="red blue">
  <div class="blue red">
```

0% First red, second blue.
44% First blue, second red.
42% Both blue 🎨
4% Both red

14,517 votes • Final results

13:36 pm - 7 Sep 2018

512 Retweets 938 Likes
Evolution of CSS Specifications

CSS1
Recommendation: 17 Dec 1996

CSS2
Recommendation: 12 May 1998

CSS2.1
Recommendation: 7 Jun 2011

CSS2.2
Working draft: 12 Apr 2016

CSS3
Decision to modularise: 14 Apr 2000
(26 modules)

CSS Snapshot 2017
(88 modules)
CSS1 does not offer:

- per pixel control: CSS1 values simplicity over level of control, and although the combination of background images and styled HTML is powerful, control to the pixel level is not possible.
- author control: the author cannot enforce the use of a certain sheet, only suggest
- a rich query language on the parse tree: CSS1 can only look for ancestor elements in the parse tree, while other style sheet languages (e.g. DSSSL [5]) offers a full query language.

We expect to see extensions of CSS in several directions:

- paper: better support for printing HTML documents
- support for non-visual media: work is in the process to add a list of properties and corresponding values to support speech and braille output
- color names: the currently supported list may be extended
- fonts: more precise font specification systems are expected to complement existing CSS1 font properties.
- values, properties: we expect vendors to propose extensions to the CSS1 set of values and properties. Extending in this direction is trivial for the specification, but interoperability between different UAs is a concern
- layout language: support for two-dimensional layout in the tradition of desktop publishing packages.
- other DTDs: CSS1 has some HTML-specific parts (e.g. the special status of the ‘CLASS’ and ‘ID’ attributes) but should easily be extended to apply to other DTDs as well.

We do not expect CSS to evolve into:

- a programming language
No layout  HTML Tables  CSS Floats  Frameworks  Grid and beyond
<table>
<thead>
<tr>
<th>Property</th>
<th>Module</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>grid-template-columns</td>
<td>CSS Grid Layout Module Level 1</td>
<td>CR</td>
</tr>
<tr>
<td>grid-template-rows</td>
<td>CSS Template Layout Module</td>
<td>NOTE</td>
</tr>
<tr>
<td>grid-template-rows</td>
<td>CSS Grid Layout Module Level 1</td>
<td>CR</td>
</tr>
<tr>
<td>hanging-punctuation</td>
<td>CSS Text Module Level 3</td>
<td>WD</td>
</tr>
<tr>
<td>height</td>
<td>CSS 2.1</td>
<td>REC</td>
</tr>
<tr>
<td>height</td>
<td>CSS Intrinsic &amp; Extrinsic Sizing Module Level 3</td>
<td>WD</td>
</tr>
</tbody>
</table>
#1: CSS Grid

Gives us an unprecedented level of control over where our elements are placed on a page.
CSS grid basics

Define your grid.

Place items in the grid.
From the simple...

```css
.container {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(10em, 1fr));
}
```
...to the elaborate

by Andy Barefoot

by Jon Kantner

by Yuan Chuan

by Andy Barefoot
#2: CSS Custom Properties

Provides us *dynamic* variables in native CSS, comes with scoping, inheritance and the cascade.
Modify variable values on the fly

```css
/* CSS that is applied to this presentation */
:root {
  --accent-colour: #ff4f5e
}

h2 {
  color: var(--accent-colour, #ff4f5e)
}
```

Credit to Mike Riethmuller for inspiring this live demo
:root {
  --spacer: 0.5em;
  --columns: 1;
}

.card {
  background-color: #fff;
  max-width: calc((100% / var(--columns, 1)) - var(--spacer, 0.5em) * 2);
  border: 1px solid #6f777e;
  margin: var(--spacer, 0.5em);
  padding: var(--spacer, 0.5em);
}

/* Modify the variable values at specific breakpoints */
@media screen and (min-width: 480px) {
  :root {
    --spacer: 0.75em;
    --columns: 2;
  }
}

@media screen and (min-width: 720px) {
  :root {
    --spacer: 1em;
    --columns: 3;
  }
}
Code by Dan Wilson from *Making Custom Properties (CSS Variables) More Dynamic*
/* Retrieves and sanitises the value of a custom property. */
const getVariable = (styles, propName) => String(styles.getPropertyValue(propName)).trim()

/* Sets the value of a custom property at the document level */
const setVariable = (propName, value) => {
  document.documentElement.style.setProperty(propName, value)
}

Code based off CSS Custom Properties (CSS Variables) Sample by Sérgio Gomes
#3: Feature Queries

Built-in feature detection with native CSS
Using `@supports` (AKA feature queries)

```
.selector {
    /* Styles that are supported in old browsers */
}

@supports (property:value) {
    .selector {
        /* Styles for browsers that support the specified property */
    }
}
```
Styles for every browser

Browsers that don't support feature queries  Browsers that do support feature queries
Self-updating designs

Pre v52 (before Grid)

Pre v62 (before Shapes)

Current
Useful resources

- CSS Grid Layout Module Level 1
- CSS Grid Layout Module Level 2
- Grid by Example by Rachel Andrew
- CSS Grid Layout Examples by Igalia
- Basic concepts of grid layout by Rachel Andrew
- Deep Dive into Grid Layout Placement by Manuel Rego Casasnovas
- Grid Auto-Placement Is Ready by Manuel Rego Casasnovas
- CSS Grid Layout and positioned items by Manuel Rego Casasnovas
- Codrops CSS reference: Grid by Chen Hui Jing
- Things I’ve Learned About CSS Grid Layout by Oliver Williams
- Learn CSS Grid by Jen Simmons
- Examine grid layouts by MDN
- Grid Level 2 and Subgrid by Rachel Andrew

- CSS Custom Properties for Cascading Variables Module Level 1
- CSS Variables: Why Should You Care? by Rob Dodson
- Codrops CSS reference: Custom Properties by Chen Hui Jing
- It’s Time To Start Using CSS Custom Properties by Serg Hospodares
- A Strategy Guide To CSS Custom Properties by Mike Riethmuller
- Developing Inspired Guides with CSS Custom Properties (variables) by Andy Clarke
- Making Custom Properties (CSS Variables) More Dynamic by Dan Wilson
- Using CSS custom properties (variables) by MDN

- CSS Conditional Rules Module Level 3: Feature queries: the ‘@supports’ rule
- Using Feature Queries in CSS by Jen Simmons
- Cascading Web Design with Feature Queries by Chen Hui Jing
- The Magic of Feature Queries, Part 1 – 5/7 Resilient CSS by Jen Simmons
- The Magic of Feature Queries, Part 2 – 6/7 Resilient CSS by Jen Simmons
- Feature Queries and Grid by Rachel Andrew
- @supports by MDN
Merci!

https://www.chenhuijing.com

@hj_chen

@hj_chen

@huijing