

JEN KRAMER • HARVARD UNIVERSITY EXTENSION SCHOOL  
@JEN4WEB

# INTRODUCING CALC() AND CUSTOM PROPERTIES

PART 1

CSS CALC()

# WHAT IS CSS CALC()?

- Ability to do math in CSS
- Compatible with length, frequency, angle, time, number, and integer
  - Addition: +
  - Subtraction: -
  - Multiplication: \*
  - Division: /
- `width: calc(100% - 80px);`

## # calc() as CSS unit value - CR

Method of allowing calculated values for length units, i.e. width:

`calc(100% - 3em)`

Global 94.28% + 2.43% = 96.7%  
unprefixed: 94.13% + 2.43% = 96.56%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?												
IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	Opera Mobile	Chrome for Android	Firefox for Android	IE Mobile	UC for			
6-8		2-3.6	4-18	3.1-5.1		3.2-5.1											
9		4-15	19-25	6	10-12.1	6.1		2.1-4.3									
10	12-17	16-67	26-75	6.1-12	15-60	7-12.1		4.4-4.4.4	7	12-12.1			10				
11	18	68	76	12.1	62	12.3	all	67	10	46	75	67	11	1			
	76	69-70	77-79	13-TP		13											

96.56% support as of Aug 9, 2019

# ADVANTAGES

- Can mix units when performing calculations (not possible in Sass)

```
.thing {  
    width: 90%; /* fallback if needed */  
    width: calc(100% - 3em);  
}
```

- Documentation of math (how do you derive a number?)
- <https://css-tricks.com/a-couple-of-use-cases-for-calc/>

# ADVANTAGES: MIXING UNITS, CALCULATING BOX SIZES

```
.area-one {  
    width: 40%;  
  
    float: left;  
  
    margin-right: 1em;  
  
}  
  
.area-two {  
  
    width: calc(60% - 1em);  
  
    float: right;  
  
}
```

# ADVANTAGES: MAKE MATH EASIER TO UNDERSTAND

```
.column-1-7 {  
    width: 14.2857%  
}
```

```
.column-2-7 {  
    width: 28.5714%  
}
```

```
.column-3-7 {  
    width: 42.8571%  
}
```

```
.column-1-7 {  
    width: calc(100% / 7);  
}
```

```
.column-2-7 {  
    width: calc(100% / 7 * 2);  
}
```

```
.column-3-7 {  
    width: calc(100% / 7 * 3);  
}
```

# SYNTAX QUIRKS

- The + and - operators must be surrounded by whitespace.
  - `calc(50% -8px)` will be parsed as a percentage followed by a negative length—an invalid expression—while `calc(50% - 8px)` is a percentage followed by a subtraction operator and a length.
  - Likewise, `calc(8px + -50%)` is treated as a length followed by an addition operator and a negative percentage.
- The \* and / operators do not require whitespace, but adding it for consistency is both allowed and recommended.
- In short: a `calc()` inside of a `calc()` is identical to just parentheses.
- <https://developer.mozilla.org/en-US/docs/Web/CSS/calc>



# CALC CAN BE USED WITH SASS

```
$a: 4em;
```

```
div {
```

```
    height: calc(#{ $a } + 7px);
```

```
    background: #e53b2c;
```

```
}
```

<https://codepen.io/thebabydino/pen/hgFno>

# PEMDAS

```
.col-3 {
```

```
    flex: 0 0 calc( 20% * 3 + 4% * 2);
```

```
}
```

- Please Excuse My Dear Aunt Sally (PEMDAS)
  - Parenthesis
  - Exponents
  - Multiplication/Division, left to right
  - Addition/Subtraction, left to right

# SUMMARY

- Use `calc()` anywhere in CSS to perform numeric calculations.
  - Don't forget about whitespace around the operator.
  - You can mix and match your units.
  - You can use parentheses or nest `calc()` statements.
  - PEMDAS applies.
  - Well-supported by most browsers in use.

PART 2

# CSS CUSTOM PROPERTIES (VARIABLES)

# WHAT IS CSS CUSTOM PROPERTIES?

- Kind of like variables, but with "weird quirks" (due to inheritance)
  - Must declare custom field within a CSS property – just like other CSS declarations
  - Values inherit – just like elsewhere in CSS
  - Can be used in JavaScript

#

## CSS Variables (Custom Properties) - CR

Permits the declaration and usage of cascading variables in stylesheets.

Usage

% of all users

?

Global

92.48% + 0.03% = 92.51%

Current aligned

Usage relative

Date relative

Apply filters

Show all

?

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	UC for Android
	12-14		4-47		10-34									
	2 15	2-30	1 48	3.1-9	1 35	3.2-9.2								
6-10	16-17	31-67	49-75	9.1-12	36-60	9.3-12.1		2.1-4.4.4	7	12-12.1			10	
11	18	68	76	12.1	62	12.3	all	67	10	46	75	67	11	1
	76	69-70	77-79	13-TP		13								

92.51% support as of Aug 9, 2019

# DECLARING A VARIABLE

```
:root {  
  --primary-color: blue;  
  --gap: 1.75rem;  
}  
  
.wrapper {  
  background-color: var(--primary-color);  
}  
  
h1 {  
  color: var(--primary-color);  
  margin-bottom: calc(var(--gap) * 5px);  
}
```

# DECLARING A VARIABLE

```
:root {  
  --primary-color: blue;  
  --default-size: 1rem;  
  --bkgd-img: url(..img/myimage.jpg);  
  --center-margin: 0 auto;  
  --bezier: cubic-bezier(.17,.67,.83,.67);  
}
```



# CSS VARIABLES VS SASS VARIABLES

```
$primary-color: #123456;
```

```
h1 {
```

```
    color: $primary-color;
```

```
}
```

```
article {
```

```
    border: 1px solid $primary-color;
```

```
}
```

After compiling:

```
h1 {
```

```
    color: #123456;
```

```
}
```

```
article {
```

```
    border: 1px solid #123456;
```

```
}
```

<https://css-tricks.com/difference-between-types-of-css-variables/>

# CSS VARIABLES VS SASS VARIABLES

```
:root {  
    --primary-color: #123456;  
}  
  
h1 {  
    color: var(--primary-color);  
}  
  
article {  
    border: 1px solid var(--primary-color);  
}
```

<https://css-tricks.com/difference-between-types-of-css-variables/>

# CSS VARIABLES HAVE INHERITANCE BUT IT'S WEIRDER THAN YOU THINK

- Sass: <https://codepen.io/jen4web/pen/rQMVKr?editors=1100>
- Variables: <https://codepen.io/jen4web/pen/RqGPxw?editors=1100>

# WHEN SHOULD I USE SASS VS. CUSTOM PROPERTIES?

- Use Sass for global values that don't typically change: color, font family, etc.
- Use custom properties for values that will change in the media queries: font size, margin, padding, widths, flex basis, etc.
- Outside the media query, in the mobile CSS layout, establish the logic for layout, then change the values of the custom properties inside the media query.
- Result is no duplicated code with different values.

# CSS VARIABLES AND JAVASCRIPT

<https://developers.google.com/web/updates/2016/02/css-variables-why-should-you-care> (scroll down halfway)

# RESOURCES

<https://codepen.io/jen4web/post/rwd20readinglist>

Includes resources for Flexbox, Flexbox grid systems, Grid, calc(), and custom properties

# QUESTIONS?

Jen Kramer

Watertown, MA, USA

Phone: 802-257-2657

[jen@jenkramer.org](mailto:jen@jenkramer.org)

[www.jenkramer.org](http://www.jenkramer.org)

Twitter: @jen4web

Facebook: [facebook.com/webdesignjen](https://facebook.com/webdesignjen)

Slides available at  
[www.slideshare.net/jen4web](http://www.slideshare.net/jen4web)

CSS  
IS  
AWESOME