CSS Custom Properties

CSS Frankfurt Meetup – 13.09.2018

Overview

Current solutions

- CSS preprocessors like Sass, Less and Stylus
- Taskrunner like Grunt & Gulp
- „Postprocessor“ PostCSS
- Enhance functionality of CSS with assets like variables, operators, functions and mixins

Goal: Make the developer‘s life easier!

But about these variables...

- You cannot change them dynamically
- They are not aware of the DOM‘s structure
- They cannot be read or changed from JS

Introduction

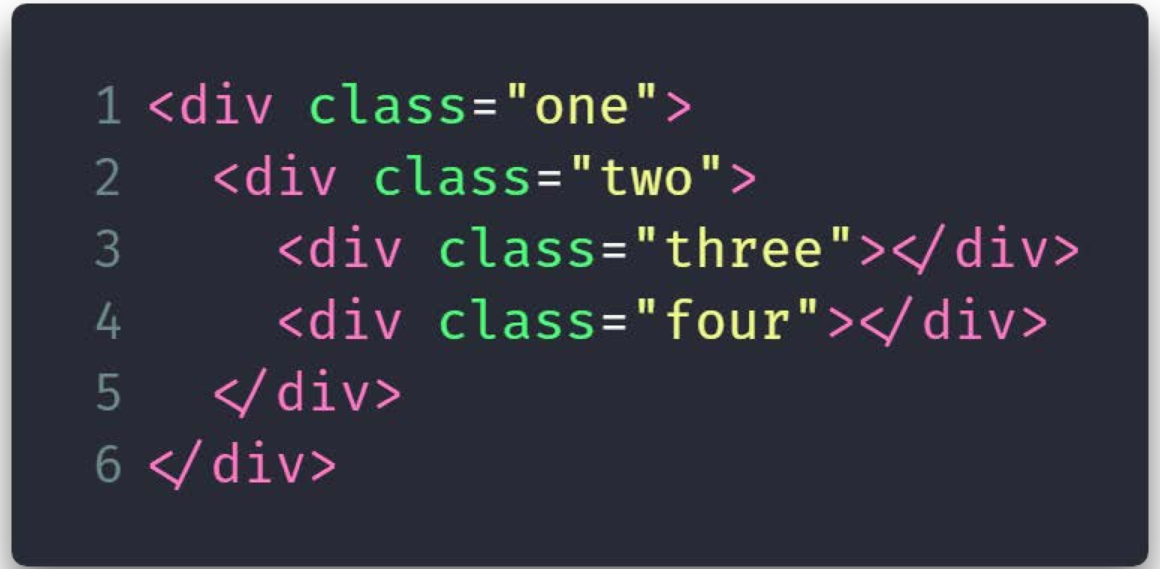
Basics: Usage

```css
:root {
  --primary: #0677cc;
  --highlight: #25b4d8;
}

a {
  color: var(--primary);
  text-decoration-color: var(--highlight, #0677cc);
  border-bottom: 1px solid var(--highlight, var(--primary));
}
```

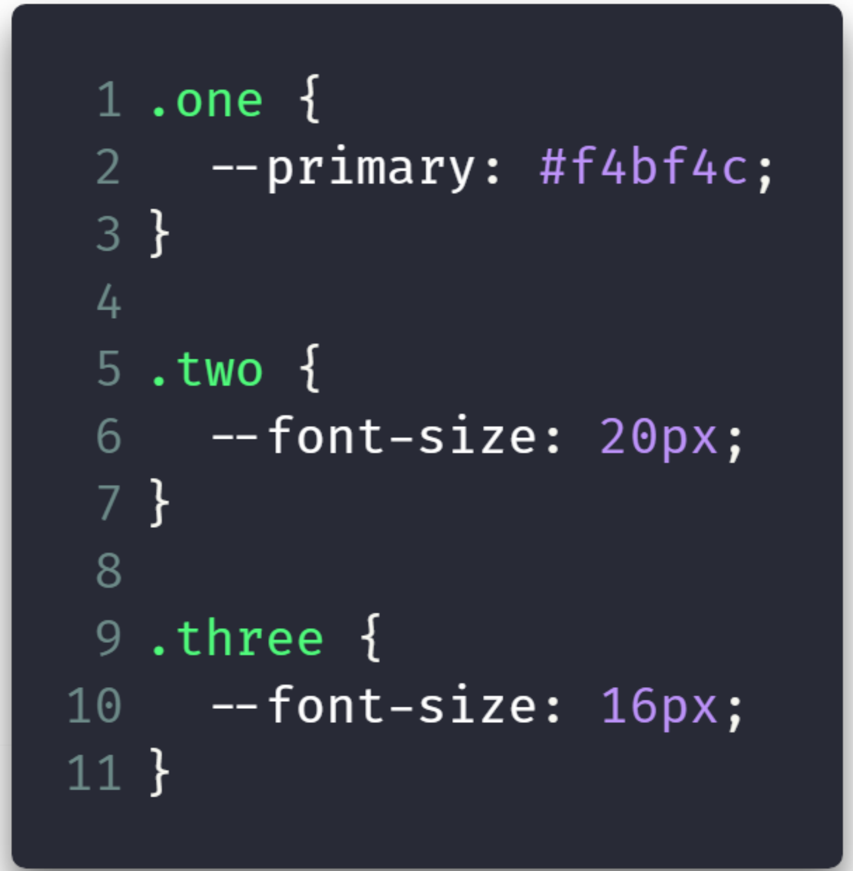
Basics: Usage

```css
:root {
  --logo: rebeccapurple;
  --main-color: #4d4e53;
  --main_bg: rgb(255, 255, 255);
  --mainLink: 201, 189, 161;
  --header-height: 68px;
  --content-padding: 10px 20px;
  --base-line-height: 1.48923849;
  --duration: .35s;
  --filler: "click me";
  --spacer: calc(2vh + 20px);
}
```

Basics: Scope and Inheritance



```html
<div class="one">
  <div class="two">
    <div class="three"></div>
    <div class="four"></div>
  </div>
</div>
```



```css
.one {
  --primary: #f4bf4c;
}

.two {
  --font-size: 20px;
}

.three {
  --font-size: 16px;
}
```

```
var(--font-size)

var(--primary)
```

Tips & Tricks

Your own CSS rules

```css
.box {
  --box-shadow-color: orange;
  box-shadow: 0 10px 30px var(--box-shadow-color);
}

.box:hover {
  --box-shadow-color: red;
}
```

calc()

```css
:root {
  --spacer: 1rem;
  --spacer-s: calc(0.5 * var(--spacer));
  --spacer-b: calc(var(--spacer) + 16px);
}

.box {
  --size: 30;
  width: calc(1vw * var(--size));
  height: calc(1vh * var(--size));
}
```

Fallbacks

```css
.info {
  background: red;
  background: var(--primary, red);
}
```

```css
.box {
  --col: var(--color, red);
  background: var(--col);
  color: var(--primary, var(--col));
}
```

```css
@supports (--custom: var) {
  /* code to run when custom
  properties are supported */
}
```

Media Queries

```css
:root {
  --gutter: 5px;
}

section {
  margin: var(--gutter);
}

@media (max-width: 600px) {
  :root {
    --gutter: 20px;
  }
}
```

Things you can‘t do

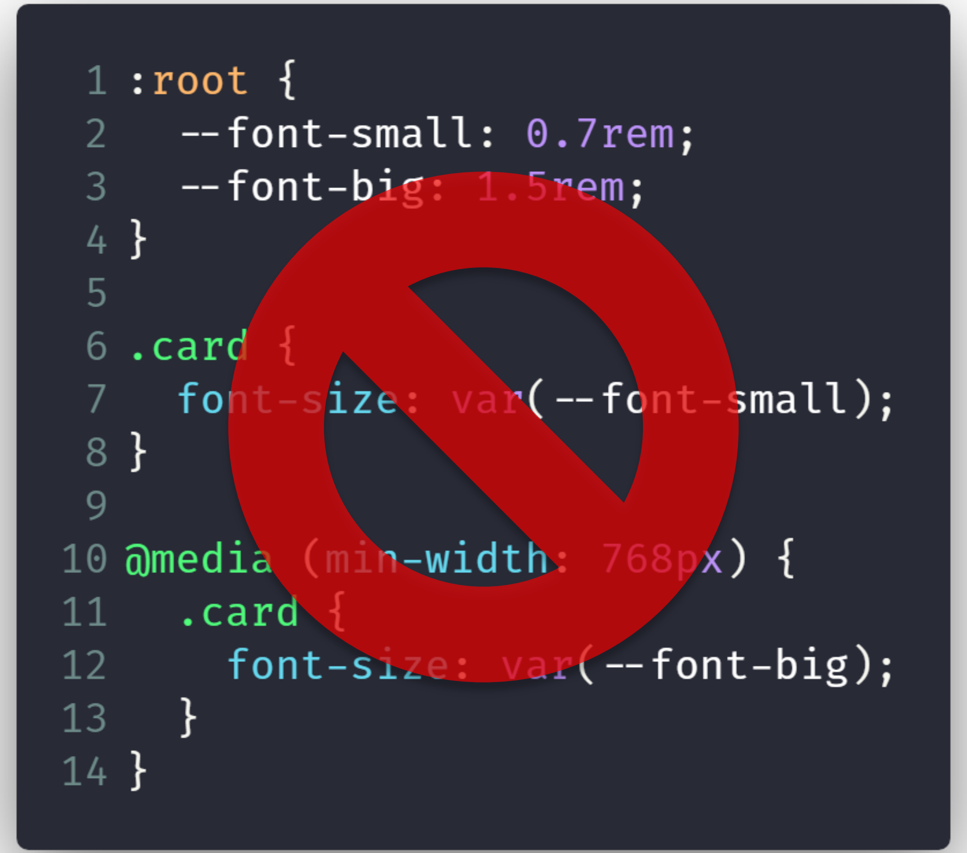
```css
.image {
  --img: "nooo";
  --img2: url("img/nooo.jpg");
  background: url("img/"var(--img)".jpg");
  background: var(--img2);
}
```

```css
.invalid {
  --foo:;
  --value: 20;
  margin: var(--value)px;
  --prop: font-size;
  var(--prop): 24px;
}
```

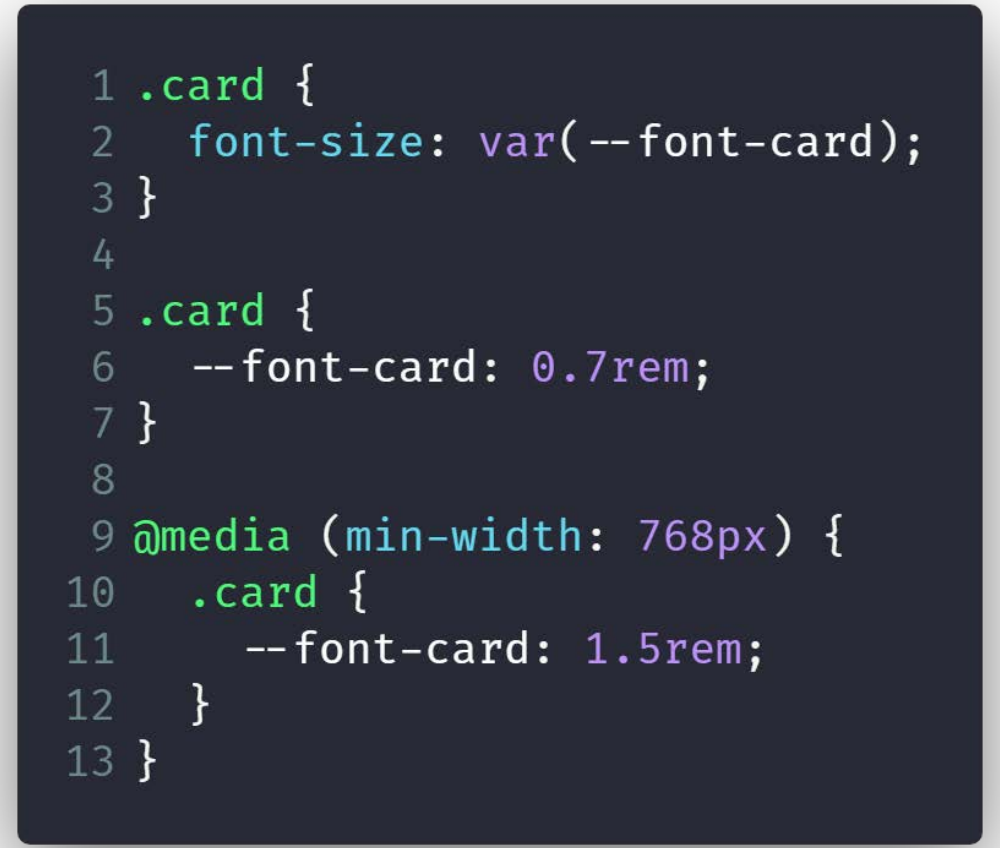
```css
:root {
  --breakpoint-s: 640px;
}

@media (max-width: var(--breakpoint-s)) {
  /* Rules */
}
```

```css
:root {
  --name: "box";
}

.var(--name) {
  /* Won't work */
}
```

Things you shouldn‘t do



```css
:root {
  --font-small: 0.7rem;
  --font-big: 1.5rem;
}

.card {
  font-size: var(--font-small);
}

@media (min-width: 768px) {
  .card {
    font-size: var(--font-big);
  }
}
```

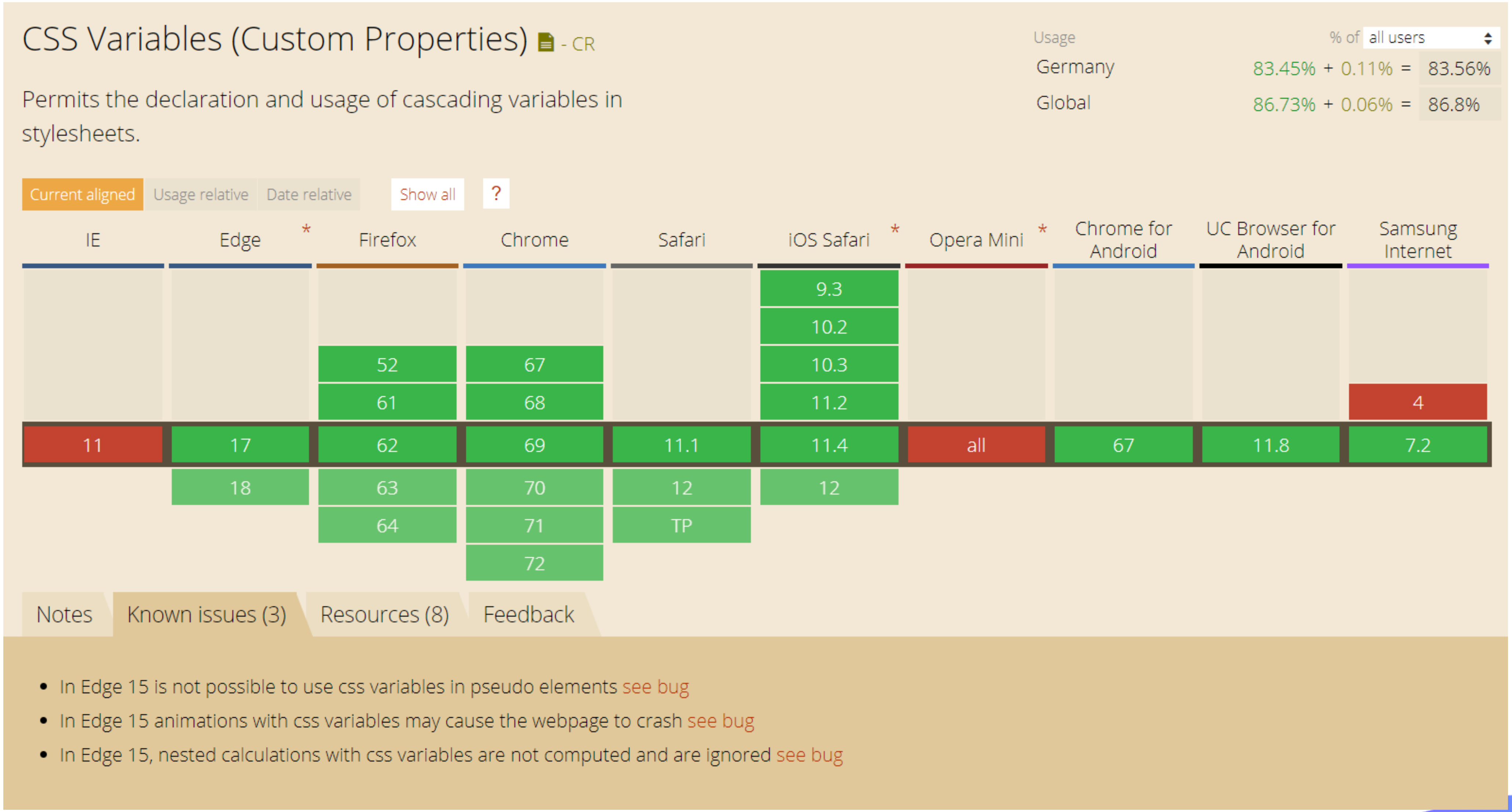


```css
.card {
  font-size: var(--font-card);
}

.card {
  --font-card: 0.7rem;
}

@media (min-width: 768px) {
  .card {
    --font-card: 1.5rem;
  }
}
```

Use cases

- Create mixins / Create your own rules
- Color themes
- Animations / Interactivity
- CSS-Grid
- Manipulate colors

Browser support



CSS Variables (Custom Properties) - CR

Permits the declaration and usage of cascading variables in stylesheets.

Usage	% of all users
Germany	83.45% + 0.11% = 83.56%
Global	86.73% + 0.06% = 86.8%

Current aligned | Usage relative | Date relative | Show all | ?

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
					9.3				
					10.2				
		52	67		10.3				
		61	68		11.2				4
11	17	62	69	11.1	11.4	all	67	11.8	7.2
	18	63	70	12	12				
		64	71	TP					
			72						

Notes | Known issues (3) | Resources (8) | Feedback

- In Edge 15 is not possible to use css variables in pseudo elements see bug
- In Edge 15 animations with css variables may cause the webpage to crash see bug
- In Edge 15, nested calculations with css variables are not computed and are ignored see bug

Conclusion

What we learnt about CSS Custom Properties:

- You can use them today without any tooling
- Dynamic, Inherit & Cascade, Can be changed with JS
- Easy syntax and clear rules
- Good browser support
- Separate style from behaviour
- Change the value, not which custom property is used

Are preprocessors dead?

No!

- Differentiate between static and dynamic values
  → Use preprocessors for global (static) variables
- Heavy calculations with Custom Properties cost performance
- Preprocessors can do the „Things you can‘t do“

Thank you!

We‘ll continue with the live examples

css-variables.netlify.com

lekoarts.de hello@lekoarts.de

You can find the slides & examples on Spectrum:

spectrum.chat/cssfrankfurt