

KHARKIV CSS

CSS VARIABLES

FIND-REPLACE STRATEGY

A screenshot of a Mac OS X desktop environment. At the top, the Dock shows 'QuickTime Player' and other icons. The menu bar includes 'File', 'Edit', 'View', 'Window', and 'Help'. In the top right corner, there are system status icons for battery, signal, and date ('Thu 11:42 AM'). The main window is a terminal or code editor showing a file named 'sample.css' located at '~/Documents/KharkivCSS'. The code contains several CSS rules:

```
sample.css — ~/Documents/KharkivCSS

1 .one{
2     color: #333;
3     padding: 10px;
4 }
5
6 .two{
7     color: #333;
8     font-weight: bold;
9 }
10
11 .three{
12     float: left;
13 }
14
```

The status bar at the bottom displays 'sample.css 12:15' on the left and 'LF UTF-8 CSS' on the right, along with small icons for file, search, and refresh.

SASS/LESS STRATEGY

A screenshot of a Mac OS X desktop environment. At the top, the Dock shows icons for QuickTime Player, Finder, and other applications. The menu bar includes Apple, QuickTime Player, File, Edit, View, Window, Help, and system status items like battery level, signal strength, and date.

The main window is a terminal or code editor showing the following content:

```
variables.scss — ~/Documents/KharkivCSS
sample.css    variables.scss    main.scss
1 $color: #333;
2
```

The terminal window has tabs for sample.css, variables.scss (which is active), and main.scss. The file path ~/Documents/KharkivCSS is shown above the tabs. The code editor shows two lines of SCSS: line 1 defines a variable \$color: #333; and line 2 is blank.

At the bottom of the screen, the Dock shows the current application is variables.scss, it's 1:14, and the file is saved in UTF-8 SCSS format.

FRAMEWORKS AND LIBRARIES (CURRENT VERSIONS)

A screenshot of a Mac OS X desktop showing a web browser window. The browser has three tabs open: 'bootstrap-sass/_variables.scss' (active), 'foundation-sites/_settings.scss', and 'react-toolbox/_colors.scss'. The address bar shows the URL for the active tab: https://github.com/twbs/bootstrap-sass/blob/master/assets/stylesheets/bootstrap/_variables.scss. The GitHub page displays the SCSS code for Bootstrap's color variables. The code includes comments explaining the color ranges and their corresponding hex values. The browser interface includes a menu bar (QuickTime Player, File, Edit, View, Window, Help) and a toolbar with various icons.

875 lines (668 sloc) | 30.6 KB

```
1 $bootstrap-sass-asset-helper: false !default;
2 //
3 // Variables
4 // -----
5
6
7 //== Colors
8 //
9 /** Gray and brand colors for use across Bootstrap.
10
11 $gray-base: #000 !default;
12 $gray-darker: lighten($gray-base, 13.5%) !default; // #222
13 $gray-dark: lighten($gray-base, 20%) !default; // #333
14 $gray: lighten($gray-base, 33.5%) !default; // #555
15 $gray-light: lighten($gray-base, 46.7%) !default; // #777
16 $gray-lighter: lighten($gray-base, 93.5%) !default; // #eee
17
18 $brand-primary: darken(#428bca, 6.5%) !default; // #337ab7
19 $brand-success: #5cb85c !default;
20 $brand-info: #5bc0de !default;
21 $brand-warning: #f0ad4e !default;
22 $brand-danger: #d9534f !default;
23
24 //== Scaffolding
25 //
26 /** Settings for some of the most global styles.
27
28 /** Background color for `<body>`.
29 $body-bg: #fff !default;
30 /** Global text color on `<body>`.
31 $text-color: $gray-dark !default;
32
33 /** Global textual link color.
34 $link-color: $brand-primary !default;
35 /** Link hover color set via `darken()` function.
36 $link-hover-color: darken($link-color, 10%) !default;
```

DAY/NIGHT THEME

QuickTime Player File Edit View Window Help Fri 9:31 AM webcamp Helen

qa.live4.io/cloud9/org1

Day Night A

ksi V VG

Missions In Progress

Android Mission
Anna (anna.pogribnyak@gmail.com)

Meth Lab Fly By
Anna (anna.pogribnyak@gmail.com)

Scheduled Missions

Weed Crops Audit
Anna (anna.pogribnyak@gmail.com)

Users

A Anna (anna.pogribnyak@gmail.com) ...

A Alex Zhukov (alex@videogorillas.com) ...

A Anton Linevich (anton@videogorillas.com) ...

MANAGE

Hardware

box1

box2

box3

box4

drone1

See other 3 Hardware

MANAGE

CSS VARIABLES



CSS4 SYNTAX



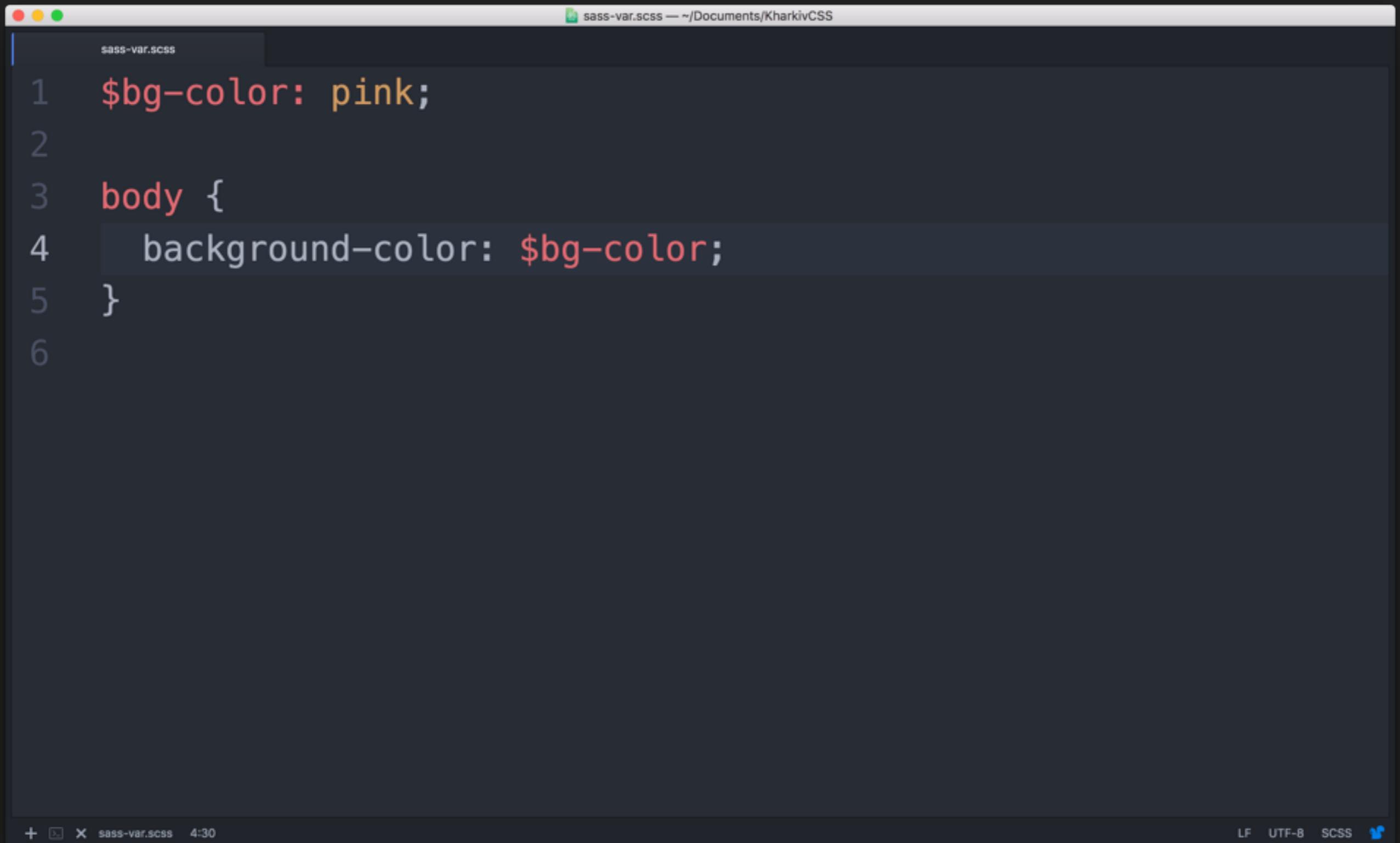
The screenshot shows a code editor window with a dark theme. The file is named 'var.css' and is located in the 'KharkivCSS' folder. The code demonstrates the syntax for defining and using custom properties.

```
var.css — ~/Documents/KharkivCSS
var.css

1 /*SYNTAX*/
2
3 var( <custom-property-name> [, <declaration-value> ]? )
4
5 /*EXAMPLES*/
6 :root {
7   --main-bg-color: pink;
8 }
9
10 body {
11   background-color: var(--main-bg-color);
12 }
13
```

The code includes a comment block for syntax rules, a section for examples, and a root selector defining a custom property named '--main-bg-color' with a value of 'pink'. This value is then used in a 'body' selector to set the background color. Line numbers are present on the left side of the code.

SASS SYNTAX



A screenshot of a code editor window titled "sass-var.scss" in a dark-themed interface. The file contains the following SASS code:

```
 sass-var.scss — ~/Documents/KharkivCSS
1 $bg-color: pink;
2
3 body {
4     background-color: $bg-color;
5 }
6
```

The code defines a variable \$bg-color with the value pink, and then uses it within a body selector to set its background color. The editor shows line numbers 1 through 6. The status bar at the bottom indicates the file is saved ("sass-var.scss 4:30") and shows file type options: LF, UTF-8, SCSS, and a small icon.



WHY ARE CSS
VARIABLES BETTER

A black and white photograph showing two wind turbines against a backdrop of a cloudy sky. The turbines are positioned vertically, with one in the foreground and another slightly behind it. The blades of the turbines are visible, and the overall scene conveys a sense of industrial activity and natural environment.

DYNAMIC

SASS MEDIA QUERY



A screenshot of a code editor window titled "media.scss — ~/Documents/KharkivCSS". The editor shows the following SASS code:

```
1 $width: 100%;  
2 .col-sm-6 {  
3   width: $width;  
4 }  
5 @media (min-width: 30em) {  
6   $width: 50%;  
7   .col-sm-6 {  
8     width: $width;  
9   }  
10 }  
11
```

The code defines a variable \$width and applies it to a .col-sm-6 class. It then uses an @media query to change the \$width variable to 50% and reapply it to the same class.

SASS MEDIA QUERY

The screenshot shows a web browser window titled "Media Query with SASS". The page content displays two orange rectangular boxes side-by-side, each labeled "col-sm-6". The browser's developer tools are open, specifically the "Elements" tab, which shows the HTML structure:

```
<body>
  <h1>Media Query with SASS</h1>
  <div class="row">
    <div class="col-sm-6">...</div> == $0
    <div class="col-sm-6">...</div>
  </div>
</body>
</html>
```

The "div.col-sm-6" tab is selected in the tools. The "Styles" tab is active, showing the CSS rules defined in the SASS file:

```
@media (min-width: 30em)
  .col-sm-6 {
    width: 50%;
  }

.col-sm-6 {
  width: 100%;
```

The right panel of the developer tools shows the element inspector for the first "col-sm-6" div, displaying its bounding box (337 x 300), position (10, 10), and padding (10, 10, 10, 10).

CSS VARIABLES MEDIA QUERY

A screenshot of a code editor window titled "media-next.css — ~/Documents/KharkivCSS". The editor shows two tabs: "media-next.css" and "media-next.html". The "media-next.css" tab contains the following CSS code:

```
1  :root{  
2      --w: 100%;  
3  }  
4  .col-sm-6 {  
5      width: var(--w);  
6  }  
7  @media (min-width: 30em) {  
8      :root{  
9          --w: 50%;  
10     }  
11  }
```

The code uses CSS variables to define a width for the root element and then applies it to a class ".col-sm-6". A media query then changes the width of the root element again. The status bar at the bottom of the editor shows the file name "media-next.css", the last save time "9:14", and file encoding information "LF UTF-8 CSS".

CSS VARIABLES MEDIA QUERY

QuickTime Player File Edit View Window Help

Thu 3:20 PM webcamp

Media Query with css variables

file:///Users/webcamp/Documents/KharkivCSS/media-next.html

Olena

Elements Console Sources Network :: X

Media Query with css variables

col-sm-6 col-sm-6

!DOCTYPE html>

<html lang="en">

><head>...</head>

><body>

><h1>Media Query with css variables</h1>

><div class="row">

>><div class="col-sm-6">...</div> == \$0

>><div class="col-sm-6">...</div>

>>...

html body div.row div.col-sm-6

Styles Event Listeners DOM Breakpoints Properties

Filter :hov .cls +

.col-sm-6 { media-next.html:22 padding: 10px; }

.col-sm-6 { media-next.css:4 width: var(--w); }

* { media-next.html:14 box-sizing: border-box; }

div { user agent stylesheet display: block; }

Inherited from body

body { media-next.html:11

margin -

border -

padding 10

10 323 x 300 10 -

10 -

-

Show all

box-sizing border-..

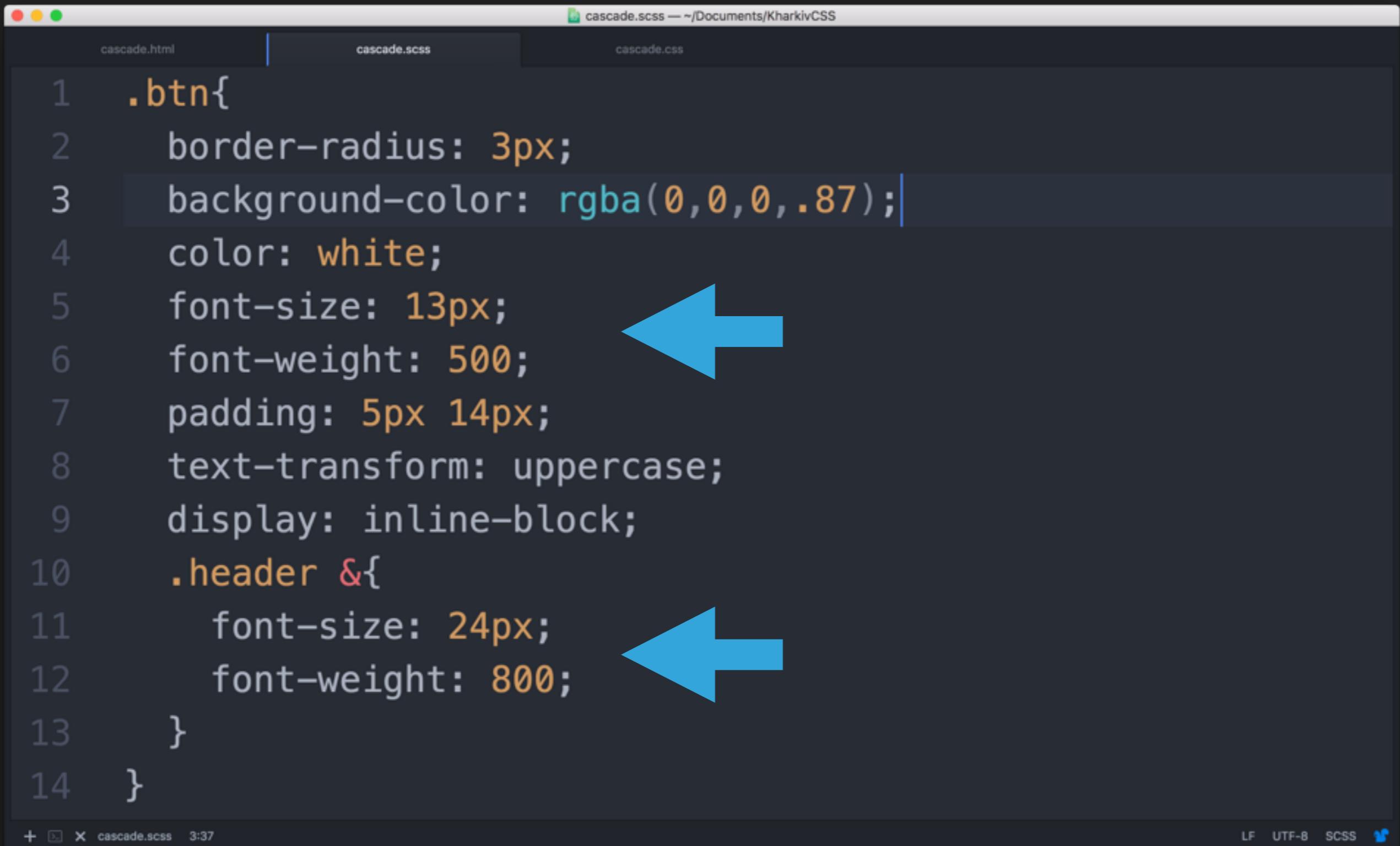
display block

border border

A black and white photograph showing two wind turbines against a dramatic, cloudy sky. The turbines are positioned vertically, with one in the foreground and another slightly behind it. The sky is filled with various cloud formations, from wispy to heavy, creating a textured background.

CASCADE

SASS CASCADE



A screenshot of a code editor window titled "cascade.scss" showing SASS code. The code defines a ".btn" class with various styles and then uses the "&" operator to inherit those styles into a ".header" class. Two large blue arrows point from the text "background-color:" in line 3 and "font-weight:" in line 12 towards the "&" operator in line 10, illustrating how styles are cascaded from the first block to the second.

```
cascade.html | cascade.scss cascade.css  
1 .btn{  
2   border-radius: 3px;  
3   background-color: rgba(0,0,0,.87);  
4   color: white;  
5   font-size: 13px;  
6   font-weight: 500;  
7   padding: 5px 14px;  
8   text-transform: uppercase;  
9   display: inline-block;  
10  .header &{  
11    font-size: 24px;  
12    font-weight: 800;  
13  }  
14 }
```

+ cascade.scss 3:37 LF UTF-8 SCSS

SASS CASCADE

The screenshot illustrates the SASS cascade principle. On the left, a browser window titled "Cascade with SASS" displays two buttons. The top button is labeled "Button in header" and the bottom one "Button without context". Both buttons have the text "BUTTON" inside them. A large blue arrow points from the "Button without context" section towards the developer tools on the right.

The developer tools are open to the "Elements" tab, showing the DOM structure:

```
<h3 class="header">
  "Button in header"
<div class="row">...</div>
```

The "div.btn" node is selected in the tree view. Below the tree, the "Styles" tab is active, showing the computed styles for the selected element. The styles are listed as follows:

```
element.style {
}
.header .btn { cascade.scss:10
  font-size: 24px;
  font-weight: 800;
}
.btn { cascade.scss:1
  border-radius: 3px;
  background-color: rgba(0, 0, 0, 0.87);
  color: white;
  font-size: 13px;
  font-weight: 500;
  padding: 5px 14px;
  text-transform: uppercase;
  width: 100px;
  height: 30px;
}
```

On the right side of the developer tools, there is a detailed view of the element's bounding box with dimensions and padding values:

margin	border	padding	width	height
14	5	5	92.625 × 28	14

CSS VARIABLES CASCADE

The screenshot shows a code editor window with two tabs: "cascade-next.html" and "cascade-next.css". The "cascade-next.css" tab is active, displaying the following CSS code:

```
1  :root{  
2      --btn-font-size: 13px;  
3      --btn-font-weight: 500;  
4  }  
5  .header {  
6      --btn-font-size: 24px;  
7      --btn-font-weight: 800;  
8  }  
9  .btn {  
10     border-radius: 3px;  
11     background-color: rgba(0, 0, 0, 0.87);  
12     color: white;  
13     font-size: var(--btn-font-size);  
14     font-weight: var(--btn-font-weight);  
15     padding: 5px 14px;  
16     text-transform: uppercase;
```

Two blue arrows point from the declarations in the ".header" and ".btn" rules back to the variable definitions in the ":root" rule, illustrating how CSS variables are cascaded down the document tree.

CSS VARIABLES CASCADE

The screenshot illustrates the use of CSS variables for styling a button. On the left, the browser window displays two buttons: one labeled "Button in header" and another labeled "Button without context". A large blue arrow points from the "Button without context" area towards the developer tools on the right.

The developer tools show the DOM structure:

```
<h1>Cascade with CSS variables</h1>
  <div class="row">
    <div class="box">
      ... <h3 class="header"> == $0
        ...
        "Button in header"
        ...
      <div class="btn">Button</div>
    </h3>
```

The "Styles" tab is selected in the developer tools. It shows the following CSS rules:

```
element.style {
}
.header { cascade-next.css:5
  --btn-font-size: 24px;
  --btn-font-weight: 800;
}
* { cascade-next.html:14
  box-sizing: border-box;
}

h3 { user agent stylesheet
  display: block;
  font-size: 1.17em;
  -webkit-margin-before: 1em;
  -webkit-margin-after: 1em;
  -webkit-margin-start: 0px;
```

The "Computed" panel on the right shows the final dimensions of the button element:

margin	18.720
border	-
padding	-
width	261.578
height	38
total width	280.278
total height	56

At the bottom, a sidebar lists properties: **box-sizing**, **display**, **font-size**, **font-weight**, and **font-family**.

CSS VARIABLES CASCADE

The screenshot shows a code editor with two files open: `cascade-next.html` and `cascade-next.css`.

`cascade-next.html` contains the following code:

```
27      }
28  
```

`</style>`

```
29 </head>
30 <body>
31   <h1>Cascade with CSS variable
32   <div class="row">
33     <div class="color">
34       text in .color
35     </div>
36   </div>
37 </body>
38 </html>
39
```

A large blue arrow points from the text "text in .color" in the `cascade-next.html` file towards the `.color` selector in the `cascade-next.css` file.

`cascade-next.css` contains the following code:

```
1  :root{
2    --color: #000;
3  }
4  /*Cascade*/
5  body{
6    color: var(--color);
7  }
8
9 .color {
10   --color: blue;
11 }
12
```

The status bar at the bottom of the editor shows the following information:

+ × cascade-next.css 12:1 LF UTF-8 CSS 1 update

CSS VARIABLES CASCADE

The screenshot shows a code editor with two files open: `cascade-next.html` and `cascade-next.css`.

`cascade-next.html` contains the following HTML:

```
27      }
28  
```

```
</style>
```

```
29 </head>
```

```
30 <body>
31   <h1>Cascade with CSS variable
32   <div class="row">
33     <div class="color">
34       text in .color
35     </div>
36   </div>
37 </body>
38 </html>
39
```

A large blue arrow points from the text "text in .color" in the `cascade-next.html` file towards the `.color` selector in the `cascade-next.css` file.

`cascade-next.css` contains the following CSS:

```
1  :root{
2    --color: #000;
3  }
4  /*Cascade*/
5  .color {
6    --color: blue;
7  }
8
9  body{
10   color: var(--color);
11 }
```

The status bar at the bottom of the editor shows the following information:

```
+ x cascade-next.css 11:2 LF UTF-8 CSS 1 update
```

CSS VARIABLES CASCADE

The screenshot shows a browser window with the title "Cascade with CSS variables". The page content displays the text "text in .color" inside a red box. The browser's developer tools are open, specifically the Elements and Styles panels.

Elements Panel: Shows the DOM structure:

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <h1>Cascade with CSS variables</h1>
    <div class="row">
      ... <div class="color">
        text in .color
      </div> == $0
    </div>
  </body>
</html>
```

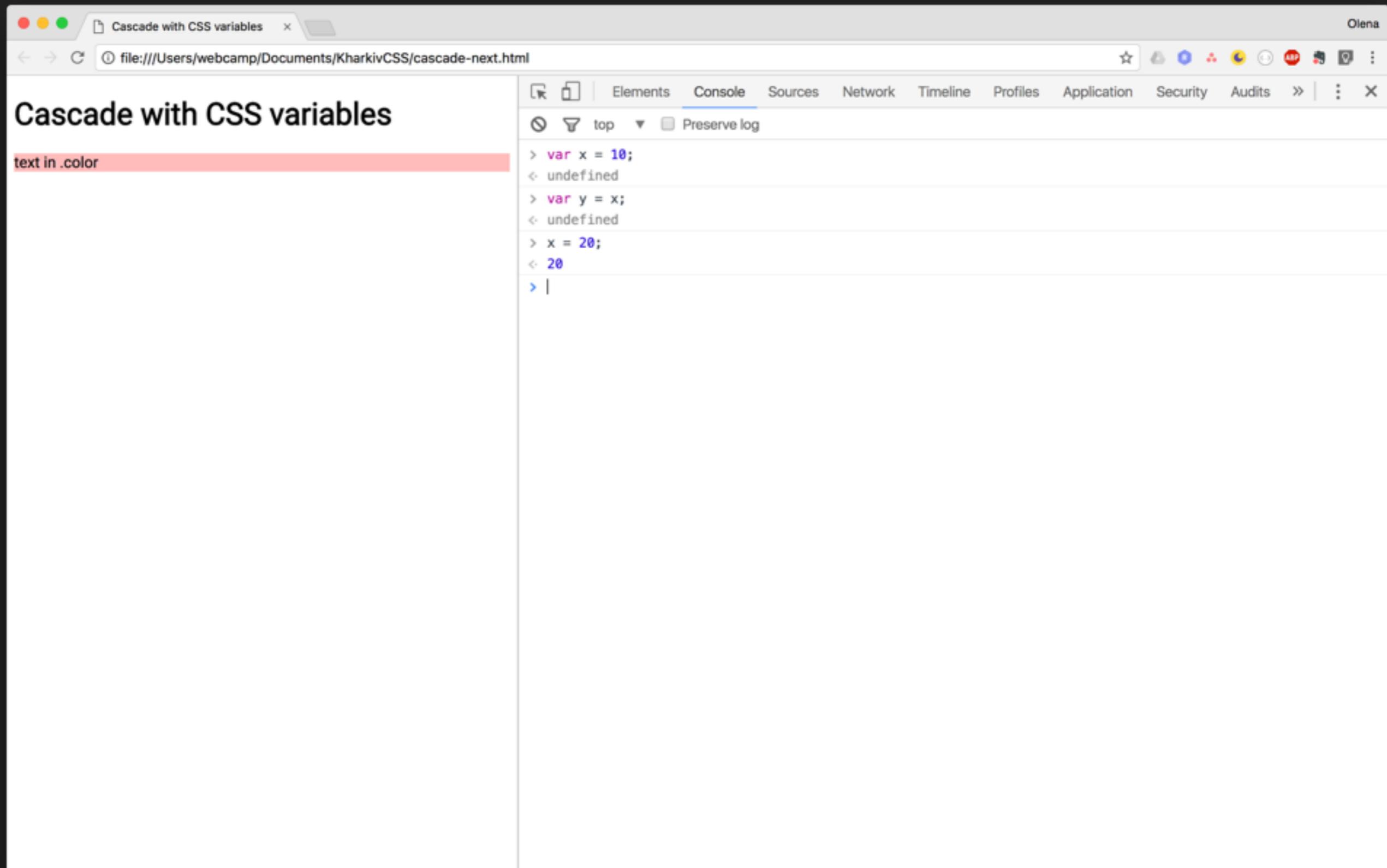
A blue arrow points from the ".color" div in the Elements panel to the corresponding CSS rule in the Styles panel.

Styles Panel: Shows the CSS rules:

```
element.style { }
.color {
  --color: blue;
}
* {
  box-sizing: border-box;
}
div {
  display: block;
}
Inherited from body
body {
  font-family: 'Roboto', sans-serif;
}
body {
  color: var(--color);
}
Inherited from html
:root {
  --color: #000;
```

At the bottom right, there is a visual representation of the element's box model, showing margin, border, padding, and content areas.

CSS VARIABLES CASCADE



CSS VARIABLES CASCADE

The screenshot shows a browser window with the title "Cascade with CSS variables". The address bar indicates the file is located at "file:///Users/webcamp/Documents/KharkivCSS/cascade-next.html". The page content displays the text "text in .color" in a pink box. On the right, the developer tools' "Console" tab is active, showing the following JavaScript interactions:

```
> var x = 10;
< undefined
> var y = x;
< undefined
> x = 20;
< 20
> y
< 10
>
```

The output shows that while `x` is explicitly set to 20, `y` retains its initial value of 10, demonstrating the cascading behavior of CSS variables.

CSS VARIABLES CASCADE

The screenshot shows a code editor with two files: `cascade-next.html` and `cascade-next.css`.

`cascade-next.html` contains:

```
27      }
28  
```

`</style>`

```
29 </head>
30 <body>
31   <h1>Cascade with CSS variable
32   <div class="row">
33     <div class="color">
34       text in .color
35     </div>
36   </div>
37 </body>
38 </html>
39
```

A blue arrow points from the text "text in .color" in the HTML file to the `.color` selector in the `cascade-next.css` file.

`cascade-next.css` contains:

```
1  :root{
2    --color: #000;
3  }
4  /*Cascade*/
5  .color {
6    --color: blue;
7  }
8
9  *{
10   color: var(--color);
11 }
```

The status bar at the bottom of the editor shows: `+ X cascade-next.css* 9:2`, `LF UTF-8 CSS`, `1 update`, and a GitHub icon.

CSS VARIABLES CASCADE

The screenshot shows a browser window with the title "Cascade with CSS variables". The page content displays the text "text in .color" in blue. The browser's developer tools are open, specifically the Elements and Styles panels.

Elements Panel: Shows the HTML structure:

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <h1>Cascade with CSS variables</h1>
    <div class="row">
      <div class="color">
        text in .color
      </div> == $0
    </div>
  </body>
</html>
```

Styles Panel: Shows the computed styles for the selected element (the inner div with class="color"). A large blue arrow points from the "text in .color" text on the page to the corresponding style entry in the developer tools.

Style	Value	Source
.color {	--color: blue;	cascade-next.css:5
* {	color: var(--color);	cascade-next.css:9
div {	display: block;	user agent stylesheet
Inherited from div.row		
* {	color: var(--color);	cascade-next.css:9
Inherited from body		
body {	font-family: 'Roboto', sans-serif;	cascade-next.html:11
* {	color: var(--color);	cascade-next.css:9
Inherited from html		
:root {	--color: #000;	cascade-next.css:1
* {	color: var(--color);	cascade-next.css:9

At the bottom of the developer tools, there are color swatches for "margin" and "border".



MATH

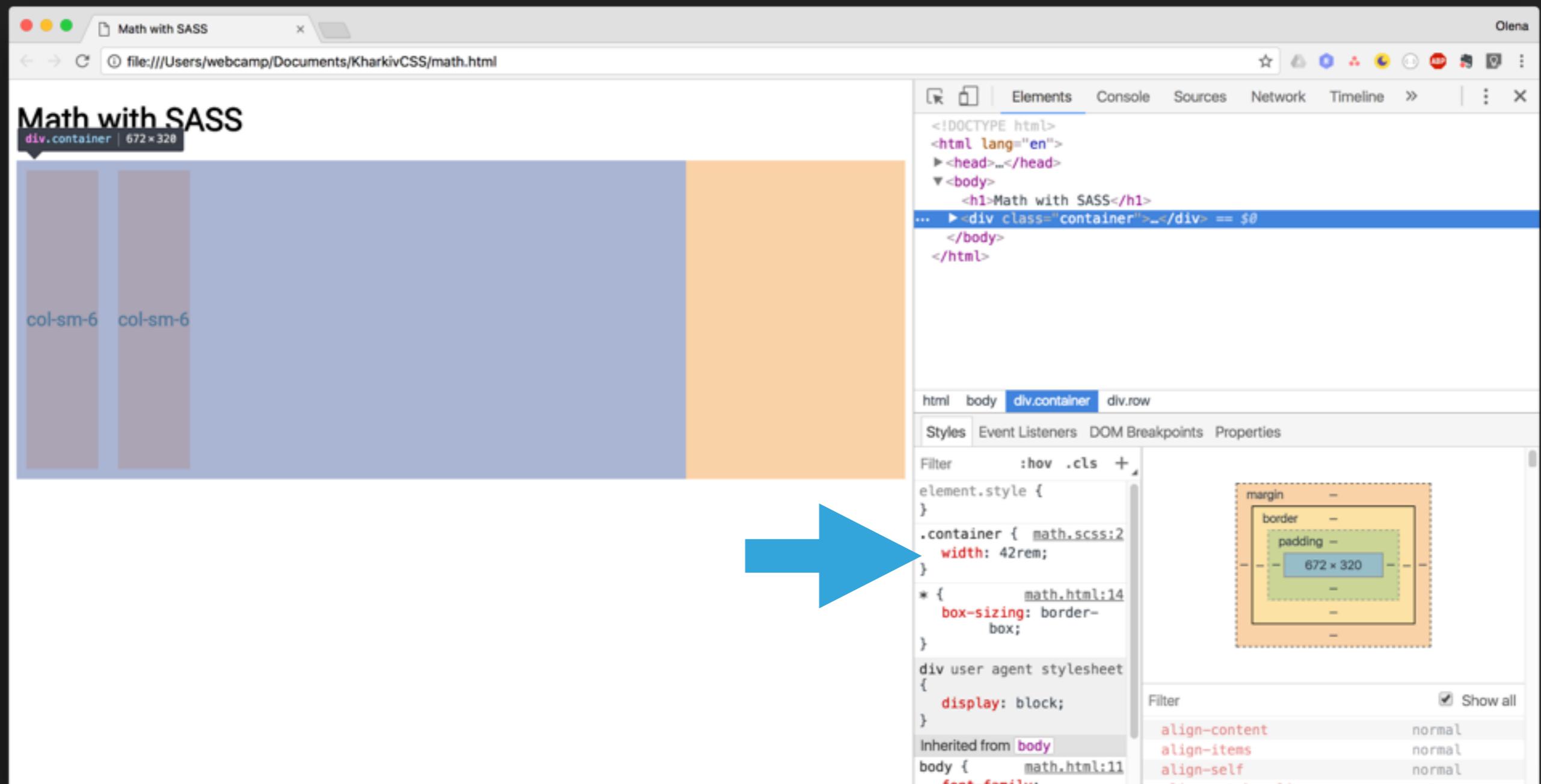
SASS MATH

A screenshot of a code editor window titled "math.scss" in a dark-themed interface. The editor shows the following SASS code:

```
1 $unit: 1rem;
2 .container{
3   width: 42 * $unit;
4 }
5
```

The code defines a variable \$unit set to 1rem, and a container class that has a width of 42 times the value of \$unit. The editor's status bar at the bottom indicates the file is 3-21 lines long, uses LF line endings, is in SCSS mode, and has 1 update pending.

SASS MATH



CSS VARIABLES MATH

A screenshot of a code editor window titled "math-css4.css" showing the following CSS code:

```
1 :root{  
2   --unit:1rem;  
3 }  
4 .container {  
5   width: 42 * var(--unit);  
6 }  
7
```

The line containing the CSS variable `var(--unit)` is crossed out with a large red X.

At the bottom of the editor, there are status icons and text: "+ × math-css4.css 5:28", "LF UTF-8 CSS", "1 update", and a GitHub icon.

CSS VARIABLES MATH

The screenshot shows a browser window with the title "Math with CSS variables". The page content displays two columns with the classes "col-sm-6" and "col-sm-6". The developer tools are open, specifically the Elements and Styles panels.

Elements Panel: Shows the DOM structure:

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <h1>Math with CSS variables</h1>
    ... <div class="container">...</div> == $0
  </body>
</html>
```

Styles Panel: Shows the CSS styles for the ".container" class:

```
element.style {
}
.container {
  width: 42 * var(--unit);
}
* {
  box-sizing: border-box;
}

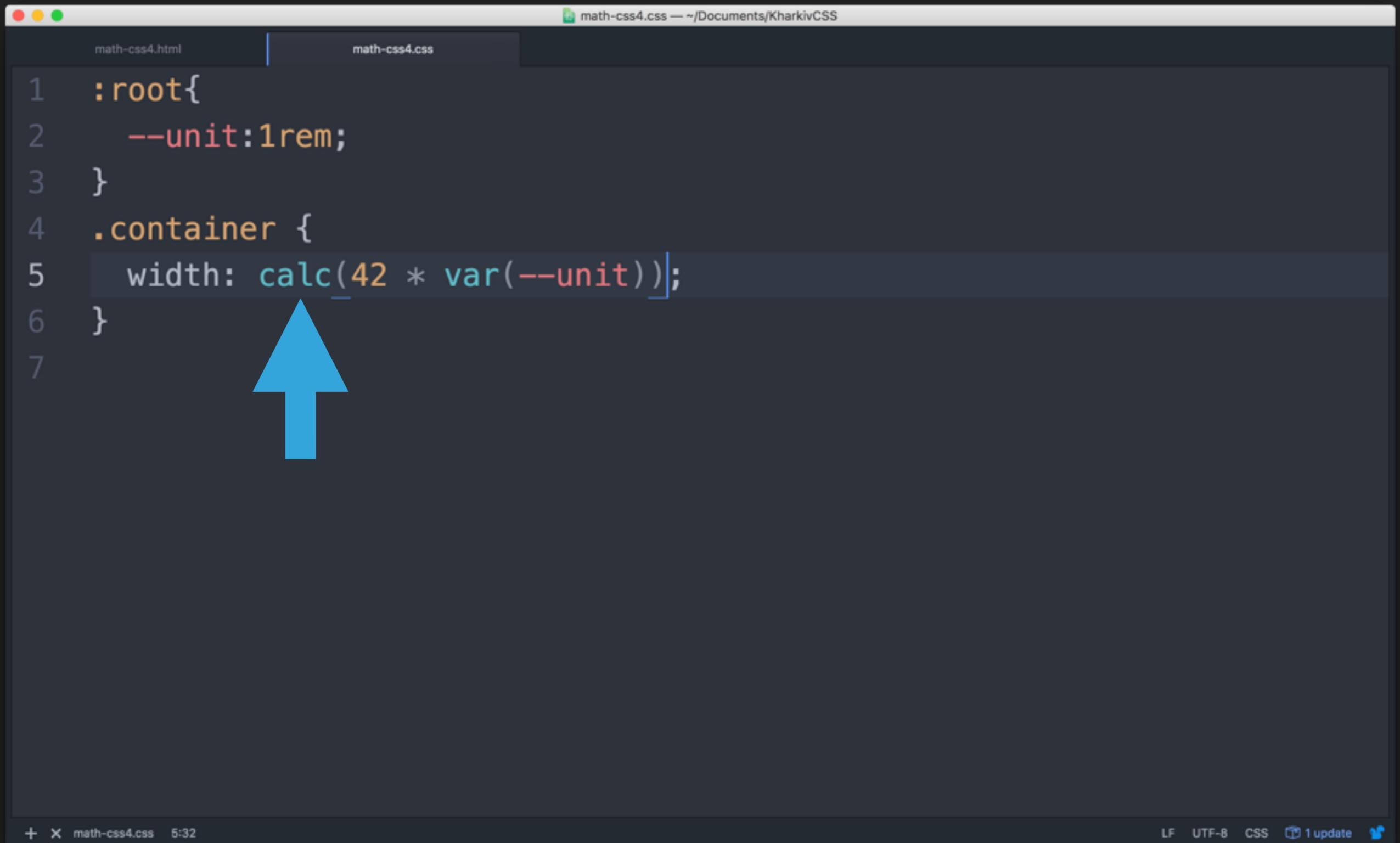
div user agent stylesheet
{
  display: block;
}

Inherited from body
```

A large blue arrow points from the browser's main content area towards the developer tools panel.

Computed Box Model Diagram: A diagram illustrating the box model for the ".container" element. It shows a blue central box labeled "892 x 320" with a green padding box around it, and an orange border box surrounding the entire structure. Labels indicate "margin", "border", and "padding" for their respective parts.

CSS VARIABLES MATH



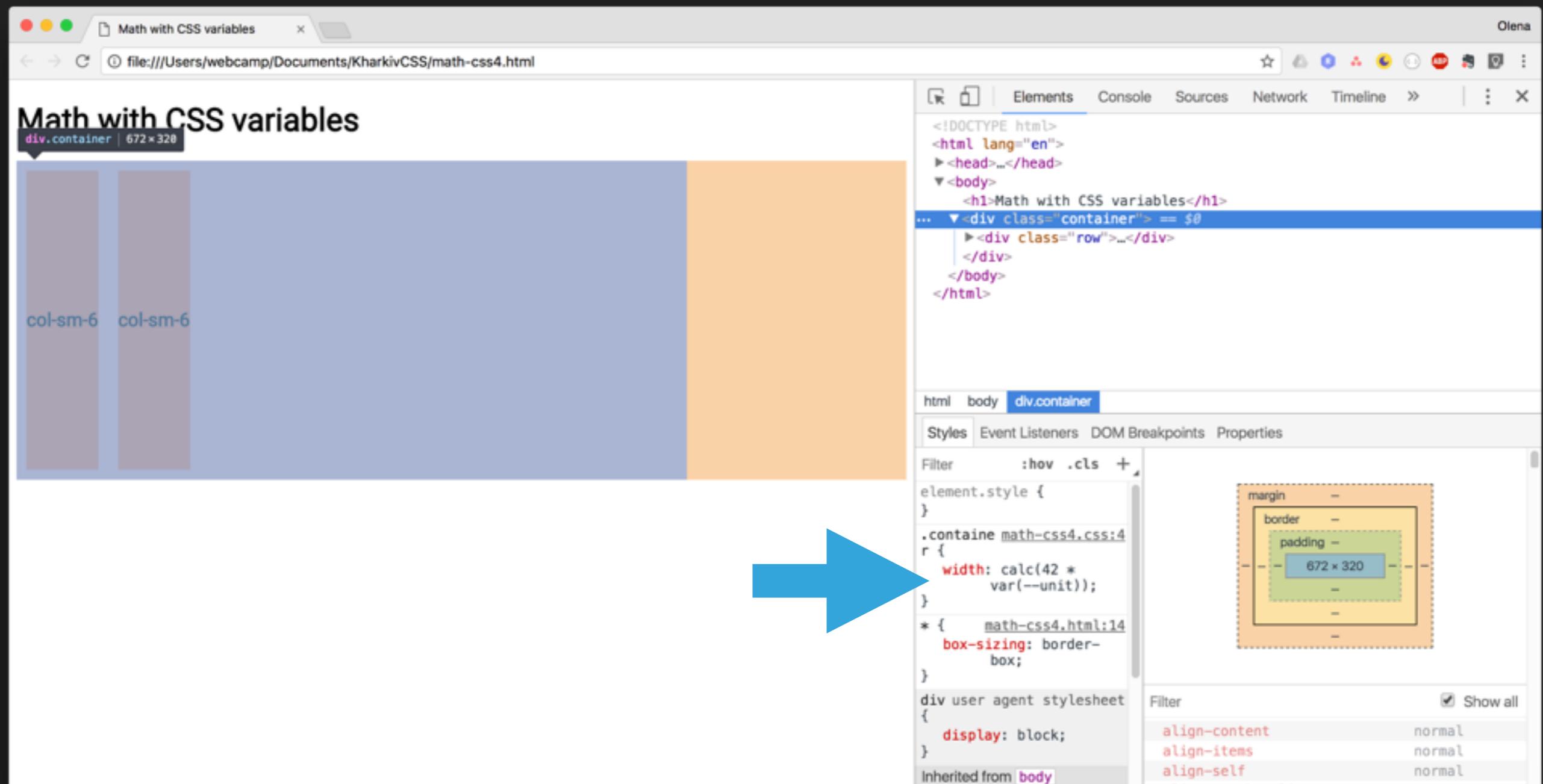
A screenshot of a code editor window titled "math-css4.css — ~/Documents/KharkivCSS". The editor shows two tabs: "math-css4.html" and "math-css4.css". The "math-css4.css" tab is active and displays the following CSS code:

```
1 :root{  
2   --unit:1rem;  
3 }  
4 .container {  
5   width: calc(42 * var(--unit));  
6 }  
7
```

A large blue arrow points upwards from the bottom of the screen towards the "width" declaration in line 5, highlighting the use of CSS variables and mathematical functions.

At the bottom of the editor, there are status icons and text: "+ X math-css4.css 5:32" on the left and "LF UTF-8 CSS 1 update" on the right.

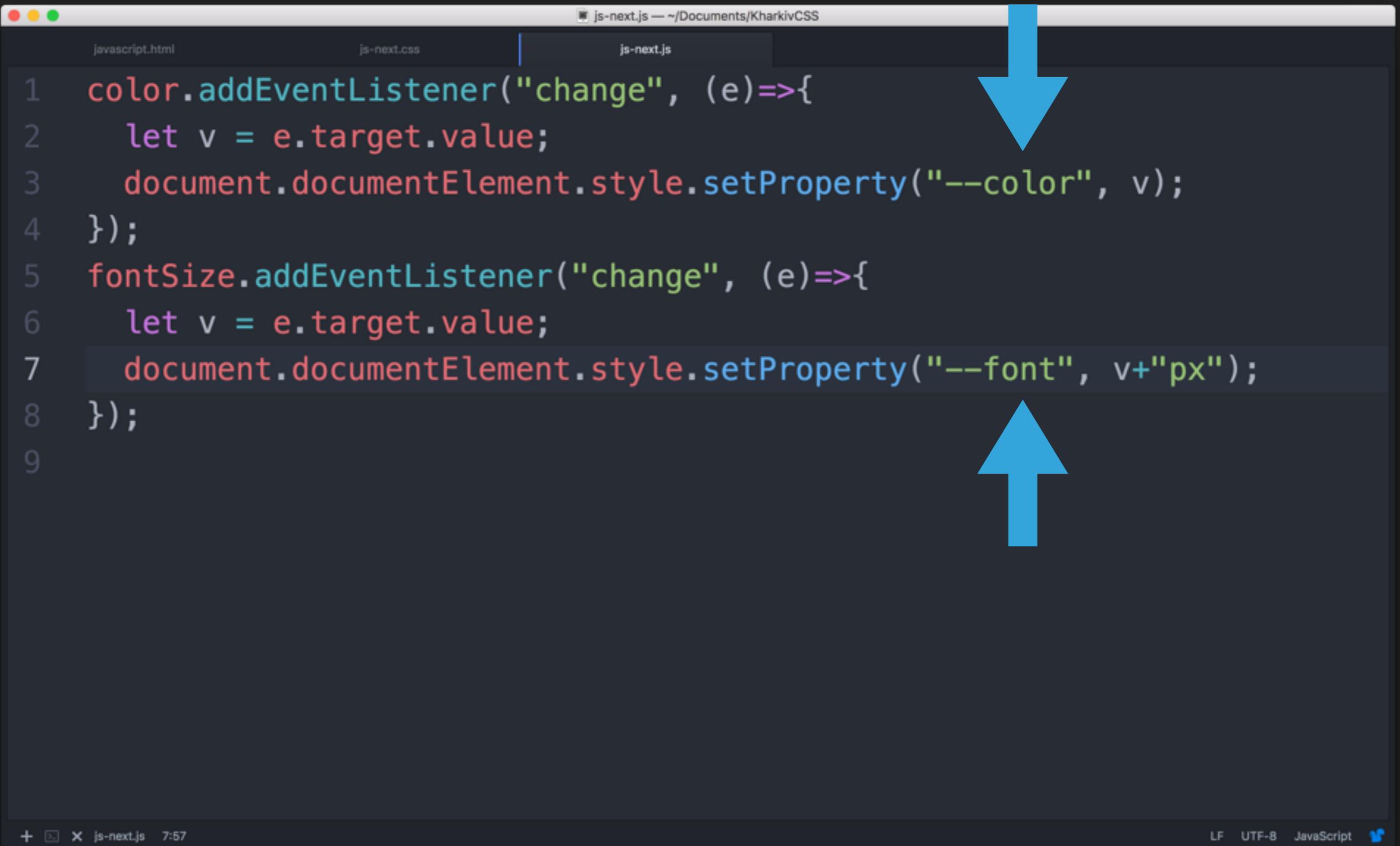
CSS VARIABLES MATH





JAVASCRIPT

CSS VARIABLES JAVASCRIPT



A screenshot of a code editor window titled "js-next.js — ~/Documents/KharkivCSS". The window contains two tabs: "javascript.html" and "js-next.css", with "js-next.js" currently active. The code in "js-next.js" is as follows:

```
1 color.addEventListener("change", (e)=>{
2     let v = e.target.value;
3     document.documentElement.style.setProperty("--color", v);
4 );
5 fontSize.addEventListener("change", (e)=>{
6     let v = e.target.value;
7     document.documentElement.style.setProperty("--font", v+"px");
8 );
9
```

Two large blue arrows are overlaid on the image: a downward-pointing arrow on the right side of the code editor window, and an upward-pointing arrow below it.

CSS VARIABLES JAVASCRIPT

Chrome File Edit View History Bookmarks People Window Help

Thu 5:39 PM webcamp Olena

Cascade with CSS variables file:///Users/webcamp/Documents/KharkivCSS/javascript.html

Cascade with CSS variables

BUTTON

Theme paragraph

Theme text

Pick a color

Change font size

Elements Sources Network Timeline Profiles

```
<!DOCTYPE html>
...<html lang="en"> == $0
  > <head>...</head>
  > <body>
    > <h1>Cascade with CSS variables</h1>
    > <div class="row">...</div>
    > <div class="row">...</div>
    > <script src="js-next.js"></script>
  </body>
</html>
```

html body

Styles Event Listeners DOM Breakpoints Properties

Filter :hov .cls +

```
element.style {
}
:root { js-next.css:1
  --color: ■#000;
  --inverse: □#fff;
  --font: 10px;
}
* { javascript.html:14
  box-sizing: border-box;
}
html[Attributes Style] {
  -webkit-locale: "en";
}
```

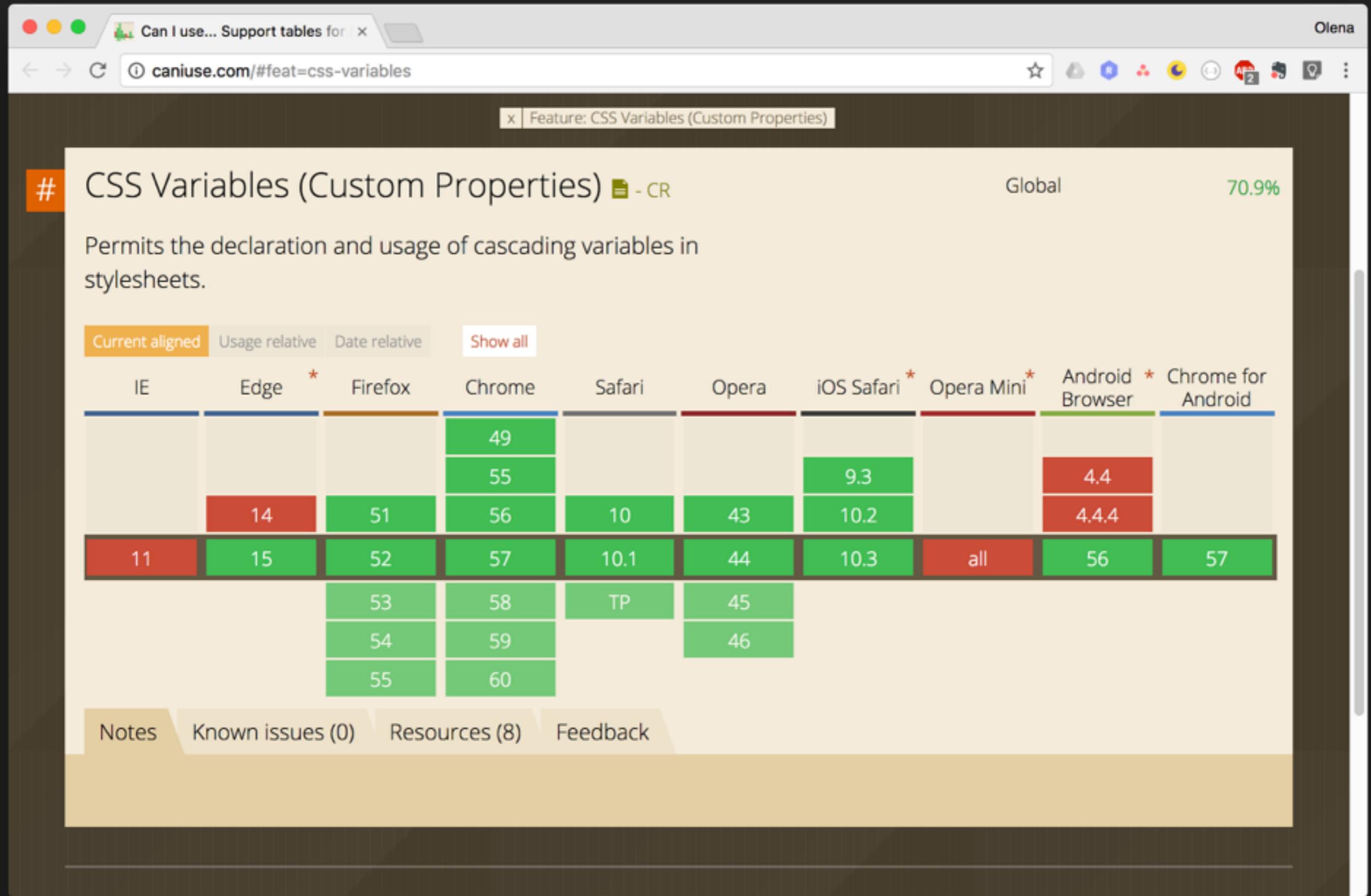
margin border padding 738 x 568.875

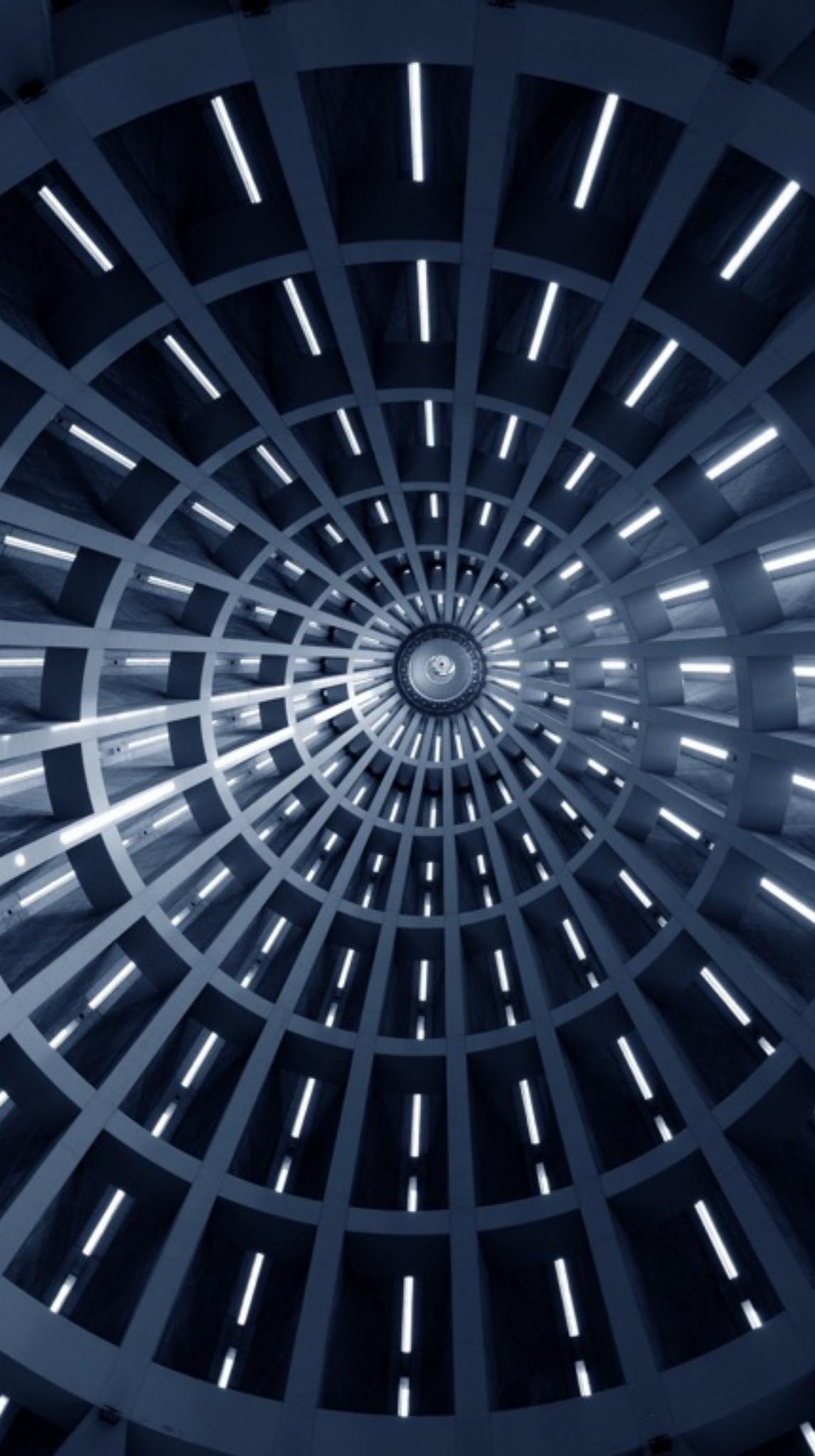
Show all

A black and white photograph showing two wind turbines against a backdrop of dramatic, billowing clouds. The turbines are positioned vertically, with one in the foreground and another slightly behind it. The blades of the turbines are visible, and the overall scene conveys a sense of renewable energy and industrial activity.

BROWSER SUPPORT

CSS VARIABLES BROWSER SUPPORT





REAL PROJECT

CSS VARIABLES IN LIBRARIES

The screenshot shows a browser window with the title "Flexbox Grid" and the URL "flexboxgrid.com". The main content features a large "Flexbox Grid" heading, a subtitle "A grid system based on the `flex` display property.", and two buttons: "Download" and "Github". Below this, there's a section titled "Responsive" with a description and three blue horizontal bars representing responsive grid columns.

The right side of the screen displays the Chrome DevTools developer tools. The "Elements" tab is selected, showing the DOM structure:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body class="layout"> == 58
    <div class="page js-page"></div>
    <script></script>
  </body>
</html>
```

The "body.layout" node is expanded, showing its styles. A specific style for ".layout" is highlighted:

```
.layout {
  display: flex;
  min-height: 100vh;
  flex-direction: column;
}
```

The "Properties" panel on the right shows various CSS properties and their values. A search bar at the top of the properties panel contains the text "margin".

Property	Value
margin	-
border	-
padding	-
width	1322 × 10241700
height	-
font-size	18px

The "Styles" panel lists other properties and their values:

- align-content: normal
- align-items: normal
- align-self: normal
- align-items: baseline: auto
- all: initial
- animation-delay: 0s
- animation-direction: normal
- animation-duration: 0s
- animation-fill-mode: none
- animation-iteration-count: 1
- animation-name: none
- animation-play-state: running
- animation-timing-function: ease
- backface-visibility: visible
- background-attachment: scroll
- background-blend-mode: normal
- background-clip: border-box
- background-color: #rgb(2)
- background-image: none
- background-origin: padding-box
- background-position-x: 0%

CSS VARIABLES IN LIBRARIES

The screenshot shows the official PostCSS website (postcss.org) running in a Chrome browser. The page features a dark background with a decorative circular emblem and a bird perched on a branch. The PostCSS logo and tagline "A tool for transforming CSS with JavaScript" are prominently displayed.

-A- Increase code readability

Add vendor prefixes to CSS rules using values from Can I Use. Autoprefixer will use the data based on current browser popularity and property support to apply prefixes for you.

:fullscreen CSS input

```
:fullscreen {  
}  
:-webkit-:full-screen {  
}  
:-moz-:full-screen {  
}  
:full-screen {  
}
```

:fullscreen CSS output

```
:full-screen {  
}
```

{ Use tomorrow's CSS, today!

Write future-proof CSS and forget old preprocessor specific syntax. Use the latest CSS syntax today with [cssnext](#). It transforms CSS specs into more compatible CSS so you don't need to wait for browser support.

OPEN CHAT

CSS Input

```
:root {  
    --red: #d33;  
}  
a {  
    &:hover {  
        color: color(var(--red) a(54%));  
    }  
}  
a:hover {  
    color: #dd3333;  
}
```

CSS Output

```
:root {  
    color: #d33;  
}  
a {  
    &:hover {  
        color: #d33;  
    }  
}  
a:hover {  
    color: #dd3333;  
}
```



[HTTPS://GITHUB.COM/WEBCAMP-UA/KHARKIV-CSS](https://github.com/webcamp-ua/kharkiv-css)