CUSTOM PROPERTIES

sind mehr als blob

CSS-VARIABLEN

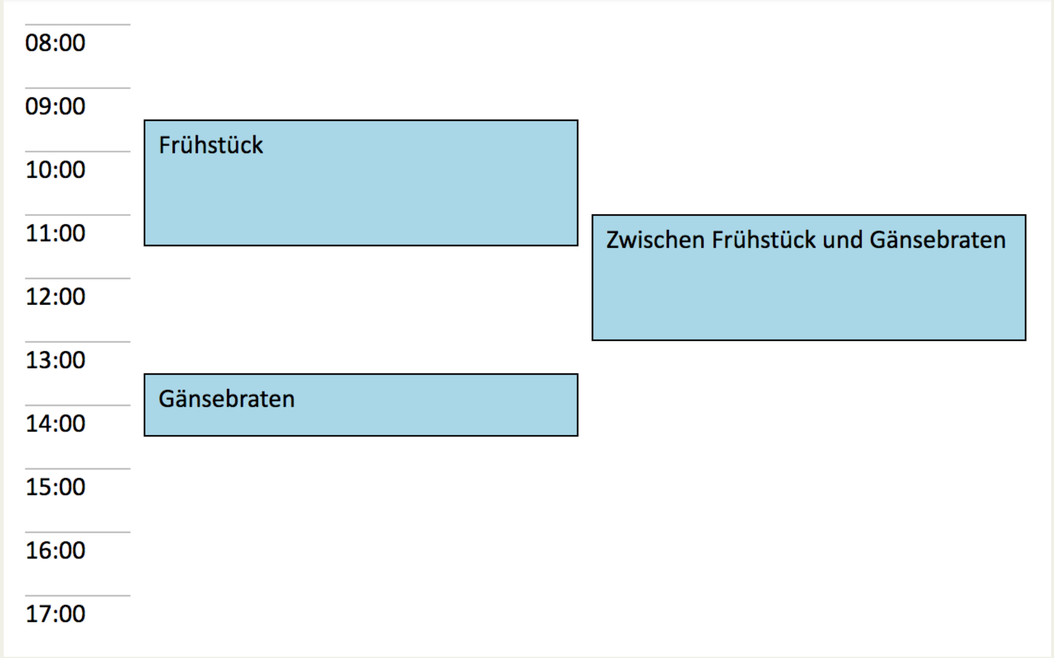
CUSTOM PROPERTIES

- beginnen immer mit `--`
- setzen: `--my-property: 42;`
- verwenden: `var(--my-property)`
- werden vererbt
- gehorchen der Spezifität

BEISPIEL: INFO-BOXEN

codepen.io/gunnarbittersmann/pen/QOxEoW

BEISPIEL: TERMINKALENDER



codepen.io/gunnarbittersmann/pen/mGVpgw

```
<li style="--start: h1100; --end: h1300">
  Zwischen Frühstück und Gänsebraten
</li>
```

Das sind keine Inline-Styles.

```
<div class="container">
  <div class="row">
    <div class="col-12 col-md-8 col-lg-6">
```

Aber das sind Inline-Styles. 💩

Schriftgröße: 16px bei 320px Breite, 18px bei 960px Breite

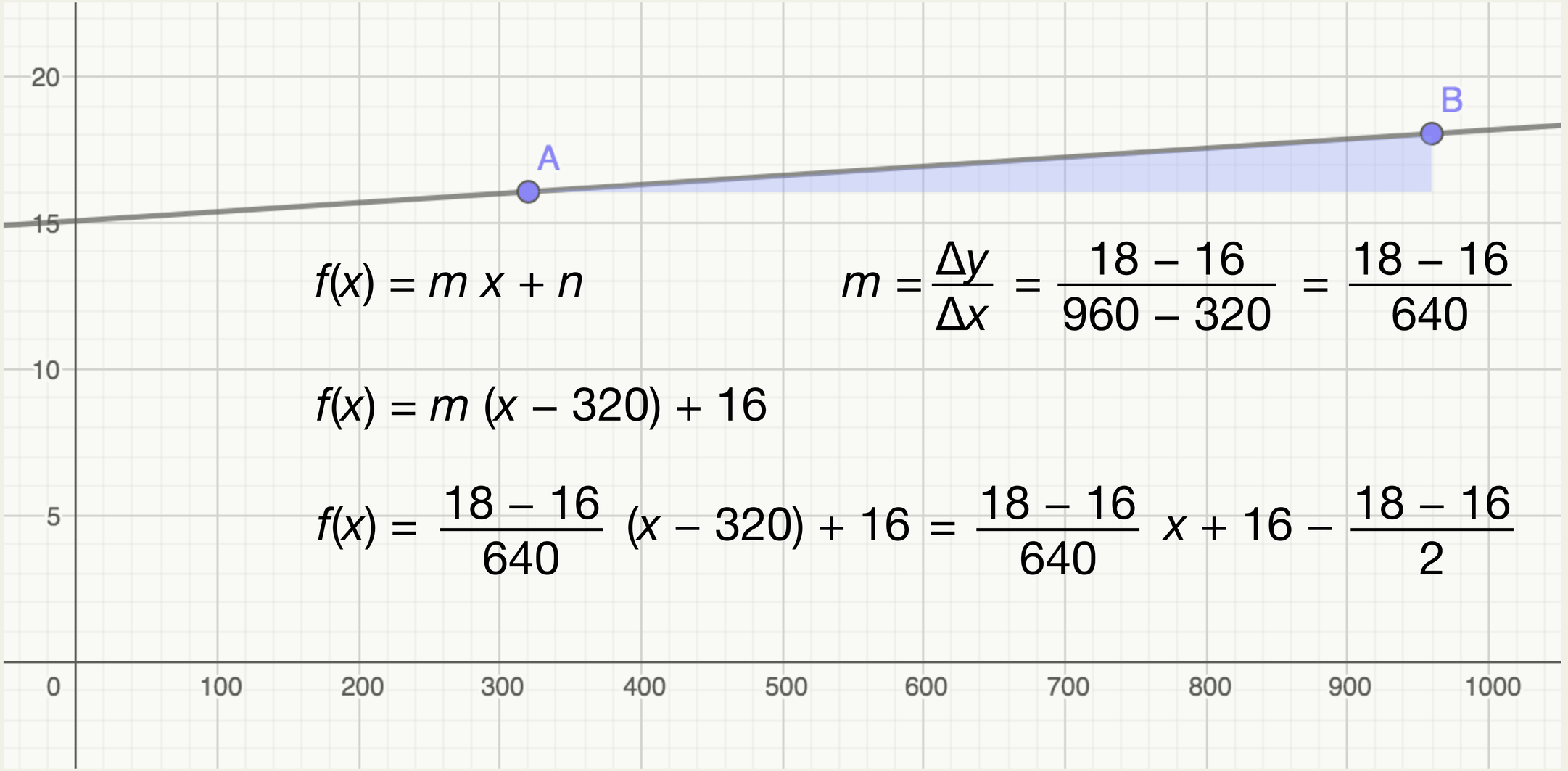
Schriftgröße: 16px bei 320px Breite, 18px bei 960px Breite

Schriftgröße: 16px bei 320px Breite, 18px bei 960px Breite

```
body { font-size: 16px }

@media (min-width: 640px)
{
  body { font-size: 17px }
}

@media (min-width: 960px)
{
  body { font-size: 18px }
}
```

```
body { font-size: 1em }

@media (min-width: 40em)
{
  body { font-size: 1.0625em }
}

@media (min-width: 60em)
{
  body { font-size: 1.125em }
}
```

Schriftgröße: 16px bei 320px Breite, 18px bei 960px Breite




```
calc((18 - 16)/640 * 100vw + (16 - (18 - 16)/2) * 1rem/16)
```

Schriftgröße: 16px bei 320px Breite, 18px bei 960px Breite

```
body
{
  font-size:
    calc((18 - 16)/640 * 100vw + (16 - (18 - 16)/2) * 1rem/16);
}
```

Überschrift: 26px bei 320px Breite, 36px bei 960px Breite

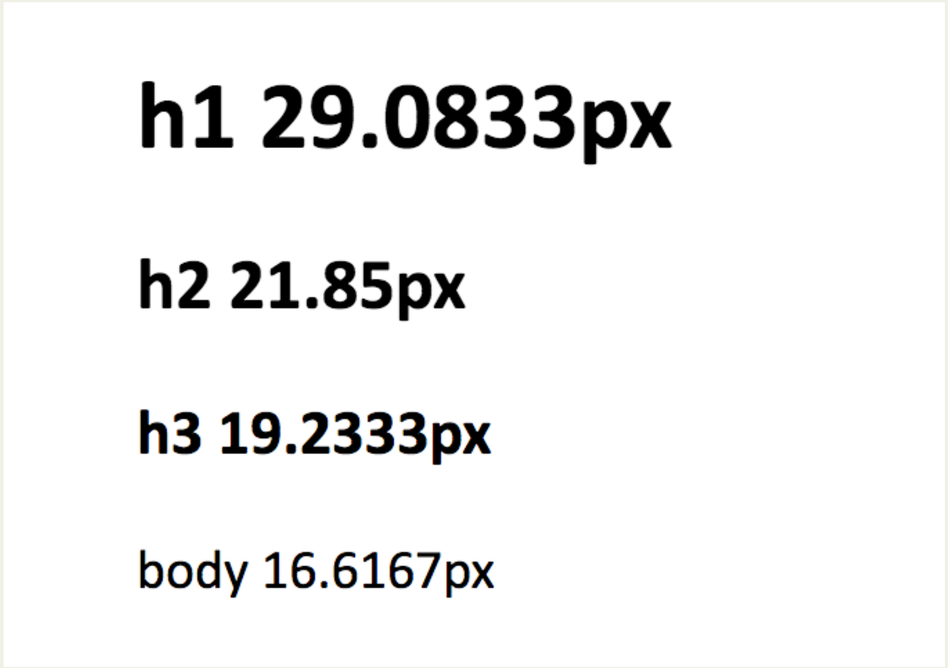
```
h1
{
  font-size:
    calc((36 - 26)/640 * 100vw + (26 - (36 - 26)/2) * 1rem/16);
}
```

```
body
{
  --font-size-320: 16;
  --font-size-960: 18;
  font-size:
    calc((var(--font-size-960) - var(--font-size-320))/640 * 100vw
      + (var(--font-size-320) - (var(--font-size-960)
      - var(--font-size-320))/2) * 1rem/16);
}
```

```
h1
{
  --font-size-320: 26;
  --font-size-960: 36;
}
```

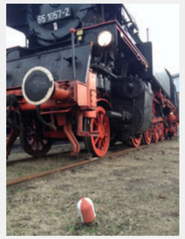
```
body
{
  --font-size-320: 16;
  --font-size-960: 18;
}

h1
{
  --font-size-320: 26;
  --font-size-960: 36;
}

body, h1
{
  font-size:
    calc((var(--font-size-960) - var(--font-size-320))/640 * 100vw
      + (var(--font-size-320) - (var(--font-size-960)
      - var(--font-size-320))/2) * 1rem/16);
}
```

Beispiel: Schriftgröße



codepen.io/gunnarbittersmann/pen/EebWbd

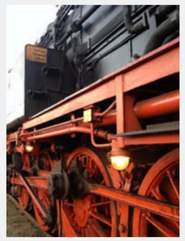
BEISPIEL: BILDERGALERIE

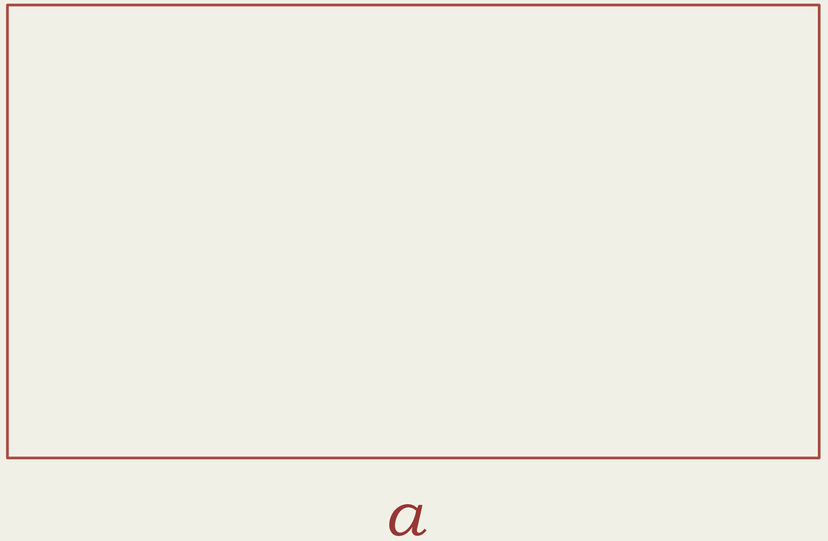









$$a \cdot b = A$$

$$a \,/\, b = r$$

$$a = r \cdot b$$

$$r \cdot b \cdot b = A$$

$$b^2 = A \,/\, r$$

$$b = \sqrt{A \,/\, r}$$

$$a = r \cdot \sqrt{A \,/\, r}$$

$$a = \sqrt{A \cdot r}$$

HERON-VERFAHREN (BABYLONISCHES WURZELZIEHEN)

$$x_0 \approx \sqrt{a} > 0$$

$$x_{n+1} = (x_n + a \,/\, x_n) \,/\, 2$$

$$\lim_{n \to \infty} x_n = \sqrt{a}$$

$a = 2$

$x_0 = 1$

$x_1 = (1 + 2 \,/\, 1) \,/\, 2 = 1.5$

$x_2 = (1.5 + 2 \,/\, 1.5) \,/\, 2 = 1.41666667$

$x_3 = (1.41666667 + 2 \,/\, 1.41666667) \,/\, 2 = 1.41421568...$

$\sqrt{2} = 1.41421356...$

Mathematische Funktionen mit Sass

Zurück zu den Wurzeln

Mit Sass könnt ihr mathematische Funktionen wie Wurzelziehen umsetzen. Wir zeigen euch anhand einer Bildergalerie, wozu das nützlich ist und wie ihr euch geschickt dem Wert für die Wurzel nähert.

Auf einer responsiven Webseite möchten wir eine responsive Bildergalerie einbauen. D.h. die Größen sollen nicht in em oder rem oder gar px angegeben sein, sondern die Bilder sollen ihre Größe dem Viewport anpassen. Die Bilder können unterschiedliche Seitenverhältnisse haben und im Quer- oder Hochformat vorliegen. Wir geben ihnen quadratische Rahmen, die an die Diasammlung unserer Eltern erinnen. Als Markup verwenden wir:

```
<ul class="gallery">
  <li>
    <div>
      <img src="…" alt="…"/>
    </div>
  </li>
  <li>
    <div>
      <img src="…" alt="…"/>
    </div>
  </li>
  ⋮
</ul>
```

Ihr werdet fragen: Wozu die div? Dazu gleich mehr.

Der Autor

Gunnar Bittersmann ist die lebende Vorlage für Waldorf oder Statler – er verrät aber nicht, für welchen der beiden. Er nickt aber auch schon mal bei einigen Showauftritten mit dem Kopf (sein Äquivalent zu »Applaus, Applaus, Applaus«), wenn bspw. Mr. John Allsopp sagt, dass *responsive design* und Webdesign doch dasselbe seien; oder wenn Mr. Jeremy Keith sagt, dass *progressive enhancement* ein wahrhaft befreiender Gedanke wäre. Gunnar lehnt sich auch gern mal beyond Logenrand … auf die Gefahr hin, abzustürzen und im Parkett aufzuschlagen.

@g16n |

HERON-VERFAHREN (BABYLONISCHES WURZELZIEHEN)

$$x_0 \approx \sqrt{a} > 0$$

$$x_{n+1} = (x_n + a \,/\, x_n) \,/\, 2$$

$$\lim_{n \to \infty} x_n = \sqrt{a}$$

```
@function sqrt($a)
{
  $x: 1;

  @for $i from 1 through 3
  {
    $x: ($x + $a / $x) / 2;
  }

  @return $x;
}
```

HERON-VERFAHREN (BABYLONISCHES WURZELZIEHEN)

$$x_0 \approx \sqrt{a} > 0$$

$$x_{n+1} = (x_n + a \,/\, x_n) \,/\, 2$$

$$\lim_{n \to \infty} x_n = \sqrt{a}$$

```
@function sqrt($a)
{
  $x: 1;


    $x: ($x + $a / $x) / 2;
    $x: ($x + $a / $x) / 2;
    $x: ($x + $a / $x) / 2;

  @return $x;
}
```

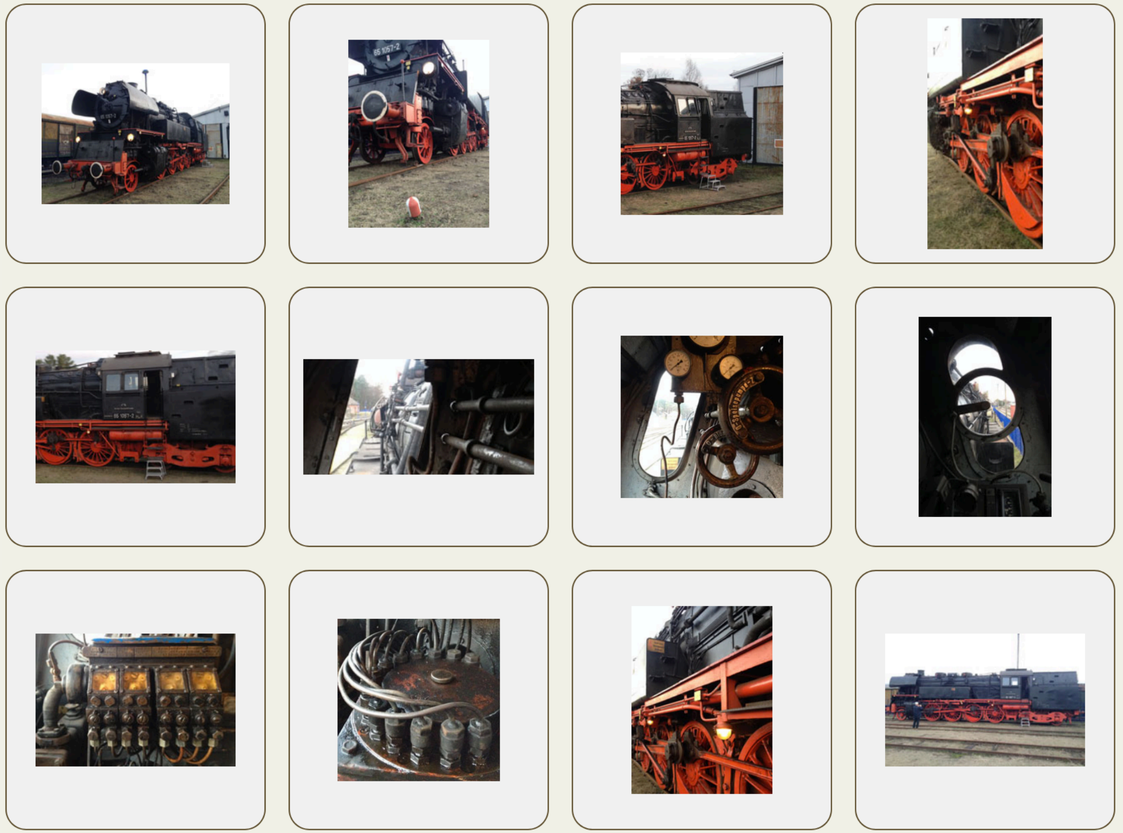
Heron-Verfahren (Babylonisches Wurzelziehen)

```
--area: 0.4;
--a: calc(var(--area) * var(--aspect-ratio));

--x0: 1;
--x1: calc((var(--x0) + var(--a) / var(--x0)) / 2);
--x2: calc((var(--x1) + var(--a) / var(--x1)) / 2);
--x3: calc((var(--x2) + var(--a) / var(--x2)) / 2);

width: calc(var(--x3) * 100%);
```

Heron-Verfahren (babylonisches Wurzelziehen)

```
--area: 0.4;
--a: calc(var(--area) * var(--aspect-ratio));

--x0: 1;
--x1: calc((var(--x0) + var(--a) / var(--x0)) / 2);
--x2: calc((var(--x1) + var(--a) / var(--x1)) / 2);


width: calc(var(--x2) * 100%);
```

BEISPIEL: BILDERGALERIE



codepen.io/gunnarbittersmann/pen/mXXYye

```
<img src="https://bittersmann.de/images/65.1057/580.jpg"
  style="--aspect-ratio: calc(4/3)" alt="" />
```

Das sind keine Inline-Styles.

```
<div class="container">
  <div class="row">
    <div class="col-12 col-md-8 col-lg-6">
```

Aber das sind Inline-Styles. 💩

CUSTOM PROPERTIES

- beginnen immer mit `--`
- setzen: `--my-property: 42;`
- verwenden: `var(--my-property)`
- werden vererbt
- gehorchen der Spezifität

„Der übelste Streich, den der Teufel je gespielt hat,
war Menschen dazu zu bringen,
CSS custom properties ‚Variablen' zu nennen."

— HUGO GIRAUDEL

CUSTOM PROPERTIES

sind mehr als blob

CSS-VARIABLEN

CUSTOM PROPERTIES

are more than just

CSS VARIABLES

CUSTOM PROPERTIES

amore

CUSTOM PROPERTIES

LiebeGerda & LIEBEDORIS

von Ulrike Rausch

LIEBEFONTS.COM