

SASS

Cheat Sheet / Spickzettel



Variablen

```
$rot: #833;

body {
  color: $rot;
}
```

Nesting

```
.markdown-body {
  a {
    color: blue;
    &:hover {
      color: yellow;
    }
  }
}
```

...zu Eigenschaften (Properties)

```
text: {
  align: center;           // identisch zu text-align: center
  transform: uppercase;    // identisch zu text-transform: uppercase
}
```

Kommentare in SASS

```
/* Mehrzeiliger Kommentar */

// Einzeiliger Kommentar
```

Extend

```
.schaltflaeche {
  ...
}
.schaltflaeche-klicken {
  @extend .schaltflaeche;
}
```

Mixins

```
@mixin ueberschrift-font {
  font-family: sans-serif;
  font-weight: bold;
}
h1 {
  @include ueberschrift-font;
}
```

Mixins mit Parametern

```
@mixin font-size($n) {
  font-size: $n * 1.2em;
}
body {
  @include font-size(2);
}
```

Mixins mit Standardwerten

```
@mixin pad($n: 10px) {
  padding: $n;
}
body {
  @include pad(15px);
}
```

Mixins mit Standard-Variable

```
// Standardwert setzen

$standard-padding: 10px;

@mixin pad($n: $standard-padding) {
  padding: $n;
}
body {
  @include pad(15px);
}
```

Composing

@import './andere_sass_datei';

Die Dateierweiterung .scss oder .sass ist optional.

rgba

rgb(100, 120, 140)

rgba(100, 120, 140, .5)

rgba(\$farbe, .5)

Vermischen

mix(\$a, \$b, 10%) // 10% a, 90% b

HSLA modifizieren

darken(\$farbe, 5%)

lighten(\$farbe, 5%)

saturate(\$farbe, 5%)

desaturate(\$farbe, 5%)

grayscale(\$farbe)

adjust-hue(\$farbe, 15deg)

complement(\$farbe)

invert(\$farbe)

fade-in(\$farbe, .5) // wie opacity()

fade-out(\$farbe, .5) // wie transparentize() – halbe Transparenz

rgba(\$farbe, .5) // Alpha auf .5 setzen

HSLA – Werte abfragen

hue(\$farbe) // → 0 – 360 Grad

saturation(\$farbe) // → 0%..100%

lightness(\$farbe) // → 0%..100%

alpha(\$farbe) // → 0..1 Transparenz

RGB – Werte abfragen

red(\$farbe) // → 0..255

green(\$farbe)

blue(\$farbe)

Farben einstellen

// Änderung über festen Wert

adjust-color(\$farbe, \$blue: 5)

adjust-color(\$farbe, \$lightness: -30%)

adjust-color(\$farbe, \$alpha: -0.4)

adjust-color(\$farbe, \$hue: 30deg)

// Änderung über Prozentwert

scale-color(\$farbe, \$lightness: 50%)

// Ändert eine Eigenschaft komplett

change-color(\$farbe, \$hue: 180deg)

change-color(\$farbe, \$blue: 250)

Unterstützte Variablen:

\$red \$green \$blue \$hue \$saturation \$lightness \$alpha

Strings

```
unquote('hallo')
quote(hallo)
to-upper-case(hallo)
to-lower-case(hallo)
str-length(hallo Welt)
str-slice(hallo, 2, 5) // "allo" – startet
bei 1, nicht bei 0
str-insert("hallo", "X", 1) // "Xhallo"
```

Einheiten (Units)

```
unit(3em) // 'em'
unitless(100px) // false
```

Nummern

```
floor(3.5)
ceil(3.5)
round(3.5)
abs(3.5)
min(1, 2, 3)
max(1, 2, 3)
percentage(.5) // 50%
random(3) // 0...3
```

Sonstiges

```
variable-exists(rot) // prüft auf $rot
mixin-exists(roter-text) // prüft auf @mixin roter-text
function-exists(rot-machen)
global-variable-exists(rot)
selector-append('.menu', 'li', 'a') // .menu li a
selector-nest('.menu', '&:hover li') // .menu:hover li
selector-extend(...)
selector-parse(...)
selector-replace(...)
selector-unify(...)
```

Feature Check

```
feature-exists(global-variable-shadowing)
```

Features

- global-variable-shadowing
- extend-selector-pseudoclass
- units-level-3
- at-error

Schleifen (for)

```
@for $i from 1 through 4 {
  .item-#{$i} { left: 20px * $i; }
}
```

Schleifen (each, einfach)

```
$menue-bild: home ueber service kontakt;
```

```
@each $bild in $menue-bild {
  .foto-#{$bild} {
    background: url('bilder/#{$bild}.jpg');
  }
}
```

Schleifen (each, verschachtelt)

```
$backgrounds: (home, 'home.jpg'), (about,
'about.jpg');
```

```
@each $id, $bild in $hintergrund {
  .foto-#{$id} {
    background: url($bild);
  }
}
```

Schleifen (while)

```
$i: 6;
@while $i > 0 {
  .item-#{$i} { width: 2em * $i; }
  $i: $i - 2;
}
```

If-Abfragen

```
@if $position == 'left' {  
  position: absolute;  
  left: 0;  
}  
  
@else if $position == 'right' {  
  position: absolute;  
  right: 0;  
}  
  
@else {  
  position: static;  
}
```

Interpolation

```
#{ $klasse } { ... } // Klasse  
call( $funktionsname ) // Funktion  
  
@media #{ $tablet }  
font: #{ $size } / #{ $line-height }  
url( " #{ $hintergrund } .jpg " )
```



Sass Schnellstart

Designe moderne Webseiten in Rekordzeit mit dem beliebtesten CSS-Präprozessor.

- Vereinfache deinen CSS-Code für immer
- Sofort durchstarten
- Keine Installationen notwendig

Listen

```
$list: (a b c);  
  
nth($list, 1) // startet mit Wert 1  
length($list)  
  
@each $item in $list { ... }
```

Jetzt noch leichter **SASS lernen** mit dem [LerneProgrammieren SASS-Schnellstart](#). Ein Kurs für alle, die frustfrei mit Präprozessoren wie SASS arbeiten wollen.

- Ohne Vorkenntnisse
- Bequem im Browser lernen
- Baue echte Webseiten und Projekte

[Hier klicken](#)

